Information Visualization (2007) 6, 75–88
© 2007 Palgrave Macmillan Ltd. All rights reserved 1473-8716 $30.00

www.palgrave-journals.com/ivs

An automated approach for the optimization of pixel-based visualizations

Jörn Schneidewind[1]
Mike Sips[2]
Daniel A. Keim[3]

[1]University of Konstanz, Konstanz, Germany;
[2]Stanford University, Palo Alto, CA, U.S.A.;
[3]University of Konstanz, Konstanz, Germany

Correspondence:
Jörn Schneidewind, Computer Science Institute, Universität Konstanz Box D78, Universitätsstr. 10, 78457 Konstanz, Germany.
Tel: +49 7531 88 3077;
Fax: +49 7531 88 3062;
E-mail: schneide@inf.uni-konstanz.de

Abstract

During the last two decades, a wide variety of advanced methods for the visual exploration of large data sets have been proposed. For most of these techniques user interaction has become a crucial element, since there are many situations in which users or analysts have to select the right parameter settings from among many in order to construct insightful visualizations. The right choice of input parameters is essential, since suboptimal parameter settings or the investigation of irrelevant data dimensions make the exploration process more time consuming and may result in wrong conclusions. But finding the right parameters is often a tedious process and it becomes almost impossible for an analyst to find an optimal parameter setting manually because of the volume and complexity of today's data sets. Therefore, we propose a novel approach for automatically determining meaningful parameter- and attribute settings based on the combined analysis of the data space and the resulting visualizations with respect to a given task. Our technique automatically analyzes pixel images resulting from visualizations created from diverse parameter mappings and ranks them according to the potential value for the user. This allows a more effective and more efficient visual data analysis process, since the attribute/parameter space is reduced to meaningful selections and thus the analyst obtains faster insight into the data. Real-world applications are provided to show the benefit of the proposed approach.

Information Visualization (2007) **6,** 75–88. doi:10.1057/palgrave.ivs.9500150

Keywords: Visual data exploration; visualization technique; visual analytics

Introduction

A wide variety of advanced visual exploration and visualization methods have been proposed in the past. These techniques have proven to be of high value in supporting researchers and analysts to obtain insight into large data sets and to turn raw data into useful and valuable knowledge by integrating the human in the exploration process. However, with the increasing volume and complexity of today's data sets, new challenges for visualization techniques arise. To keep step with the growing flood of information, visualization techniques are getting more sophisticated, for example by integrating automated analysis methods or providing new visualization metaphors as proposed in the context of visual analytics.[1]

But this also means, that visualization techniques are getting more complex, forcing the user to set up many different parameters to adjust the mapping of attributes to visual variables on the display space. In classical data exploration, playing with parameters to find a promising parameter setting is an important part of the exploration process, but with the increasing number and diversity of the parameters it becomes more and more difficult to determine a good parameter setup, which is vital for insightful visualizations.

Received: 23 June 2006
Revised: 31 July 2006
Accepted: 9 September 2006

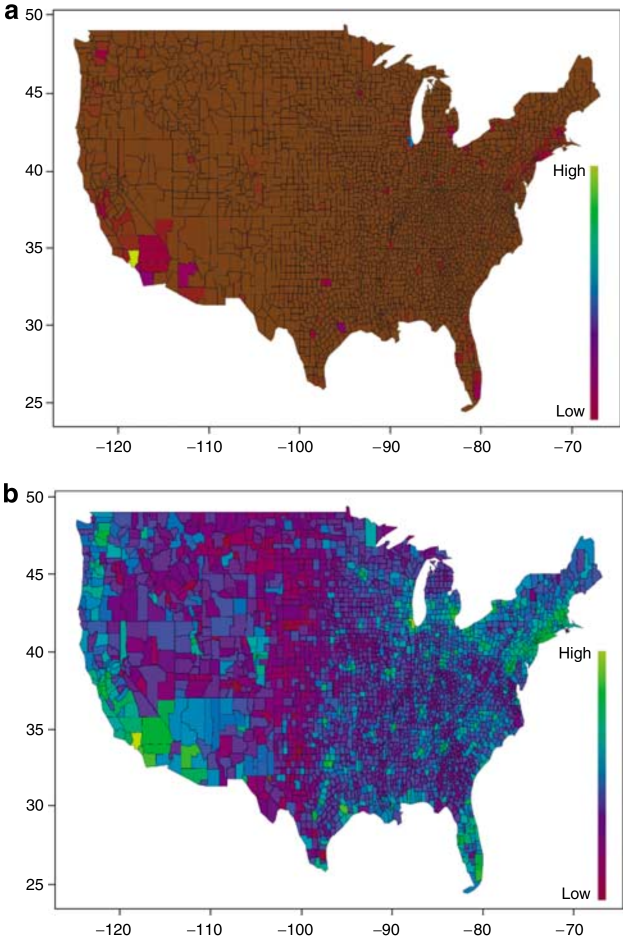

Figure 1 A typical application scenario: the visual analysis of a census data set involves different normalizations to a color scale; Although both visualizations are based on exactly the same input data, figure 1b provides more insight since a logarithmic color scale is more suitable for the underlying data distribution.

For example, if we have 50 attributes (or attribute dimensions) and four parameters for the visual mapping, as for example, in pixel bar charts[2] employed in the section Pixel bar charts, then we have over 5 million possible, mappings and it is very unlikely to find useful ones interactively.

Suboptimal parameter settings or the investigation of irrelevant data dimensions make the exploration process tedious and an interactive search impossible. In general, finding a good parameter setup is a challenging task for the analyst, since it is often not clear what is the best parameter setting for a given task, due to the huge parameter and attribute space.[3]

A simple application scenario is shown in Figure 1. The figure shows two choropleth maps visualizing U.S.A. population density data at county level. The two maps are based on the same input data, but created with two different parameter settings. More precisely, in the left figure a linear color mapping was chosen, in the right figure a logarithmic color mapping was chosen. It is easy to see that the linear mapping provides much less insight into the data than the logarithmic mapping, because the data are highly non-uniformly distributed. For instance, very high populated areas around Los Angeles, Chicago or Manhattan cause uniform dark colors for the remaining U.S.A. and it is almost impossible to see fine structures or differences in population density among them. In practice, the analyst does not know *a priori* which normalization function is best suited for a given data set and he may test some preferred ones. Of course, there are typically much more parameters that have to be selected.

But the growing data complexity and data volumes do not allow such playing with data by hand anymore. Therefore, the paper aims at supporting the user in finding promising parameter setups from the available parameter space to speed up the exploration process. We present a framework that employs automated analysis methods to detect potentially useful parameter settings for a given pixel-based visualization technique and an associated input data set, with respect to a given user task like Clustering or Outlier detection. This approach is an extension of our work introduced in.[4]

Background

In many application scenarios, analysts have to deal with large parameter spaces when using visualization techniques to explore large data sets. These parameters control the visual encoding of the data, including the selection of attributes from the input data, the selection of the color scale, algorithm parameters, the selection of visual variables and so on. The problem is that the optimal parameter setting for a given task is often not clear in advance, which means that the analyst has to try multiple parameter settings in order to generate valuable visualizations. Since such selections can hardly be done manually, the integration of automated methods to support the analyst has been recognized as an important research problem in the context of visual analytics.[1]

First approaches have been proposed in the context of visualization. In House *et al*[5], a semi-automated technique to search the visualization parameter space with applications in surface texturing is presented that focuses especially on perceptual and aesthetic concerns. The basic idea is to employ a genetic algorithm to guide a human-in-the-loop search through the parameter space. The approach produces some initial visualizations constructed from different parameter settings (parameter vectors). The resulting visualizations are rated by the user, and this rating is then used to guide the progress of the genetic algorithm. This technique follows approaches proposed in Sims[6] and Greenfield[7], which coupled image generations with user feedback in the context of genetic algorithms. As a result, the approach builds a database of rated visualization solutions. Data mining technique may then be used to extract information from the database. The drawback of these approaches is that the user is still involved in the evaluation stage, that means that the number of visualizations that can be evaluated is rather limited.

In the field of InfoVis, some techniques were proposed which avoid this problem by applying exclusively automated methods. In Wilkinson *et al*,[8] Tukey[9] and Tukey and Tukey[10] graph theoretic approaches to analyze Scatterplots were proposed. This work called Scagnostics highly influenced our work. Since Scatterplot matrices contain as many scatterplots as there are pairs of parameters (attributes), they do not scale well to high numbers of dimensions. Therefore, it would be useful to reduce the number of scatterplots by pruning irrelevant ones with respect to a given task. The goal of the mentioned approaches was to find interesting attribute relationships by creating scatterplot matrices from the data and then analyze each scatterplot, which reveals a relationship between two attributes, for certain properties using graph theoretic methods. The basic idea is to construct geometric graphs based on the data points of each scatterplot and then to compute relevance measurements from these graphs. For example, properties of the convex hull and the minimal spanning trees of the scattered points are used for outlier or cluster analysis. These techniques have shown that automated analysis works well to filter relevant from irrelevant scatterplots.

With our approach, we extend this idea to a broader set of visualization techniques. We provide analysis functions to analyze both patterns in the data using data analysis techniques as well as the patterns contained in the images by using image analysis techniques. Hence, we suggest a general process model for automated parameter space analysis and show how we applied this model to pixel based visualization techniques namely pixel bar charts and space filling curves. Although data mining methods are commonly used for data analysis, our approach is novel since only little research has been done on analyzing visualizations with respect to their information content using image analysis methods.

The pixnostics approach

The next sections focus on the problem of automatically searching through visualization parameter space to support the user in finding promising parameter settings to speed up the exploration process. Since we deal with pixel images resulting from visualizations, instead of scatterplots, we call our approach pixnostics instead of scagnostics.

The basic idea is to integrate image analysis and retrieval methods in the visualization process to compute measurements based on the properties of the image pixels like color and pixel neighborhoods. Therefore we consider visualization of data as a process of creating images I, whereas we focus in our experiments on pixel-based visualization techniques in which every data point corresponds to a pixel.

In the following, we describe this problem and how we addressed it in more detail.

Visualization parameter space

The challenging task in generating expressive visualizations is to find an adequate visual encoding of the input data set in the data display space. The visual encoding depends on the input data, the employed visualization technique and the visual variables given by a particular parameter setting. In classical data exploration, the mapping to visual variables described by Bertin,[11] such as position (x,y), size or color is manually controlled by the user.

More precisely, the central process of visualization V can be described as

$$I(t) = V(D, S(P), t), \tag{1}$$

where data D is transformed using a specification S into a time varying image $I(t)$, according to the work proposed by Wijk.[12] To simplify matters, we aim at determining initial parameter settings for non-animated visualizations, therefore the time t can be excluded from our considerations. (Note that the complexity of the problem would be boosted exponentially, if we take time into consideration.) Figure 2 illustrates this classical visualization pipeline.

Based on the formula above, the data set D is given as input data within a database environment with $D = (d_1, \ldots, d_n)$. In general the employed visualization technique V is defined by the application scenario and is given by the user. In[13] an approach is proposed where the visualization technique is determined automatically, but his approach is limited to relational data.

In the following we use novel pixel-based techniques namely pixel bar charts[2] and Jigsaw maps[14] as examples to explain our new idea and focus mainly on demographic data provided by the U.S. Census Bureau to evaluate our approach.

Pixel bar charts (described in the section pixel bar charts) have four parameters and more to adjust the visual encoding. It uses one parameter to separate the data into bars, two parameters for ordering of pixels in x and y directions and one parameter for color coding. However, by using pixel bar charts for analyzing typical data sets with at least 10–20 attribute dimensions, it is very hard for the user to find interesting patterns hidden in the data via manual parameter selection, since there are thousands of visual mappings possible, but the number of parameter settings that the user may try manually is limited.

In our setting, the input data D and the visualization technique V is given by the user and $P = \{P_i = (p_i^1, \ldots, p_i^m)\}$ as instance of a parameter setting generating image $I(S(P_i))$ is determined by the system.

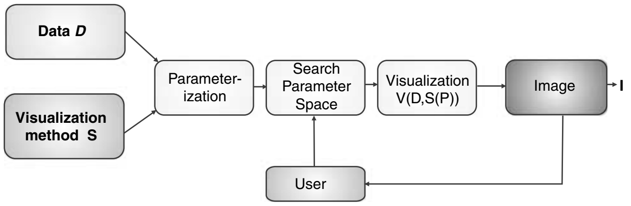

Figure 2 Classical visualization process: the user has to find an optimal parameter setting manually. Ideally such a setting should produce an insightful visualization *I*.

Limits and problem complexity

Most visualization techniques can handle less attributes than provided by the dimensionality of the input data set, so in the visualization step potentially useful attribute combinations must be selected from the data. As an example, we consider a data analyst who wants to use the pixel bar chart technique to analyze real-world customer purchase data. Such data sets typically contain at least 20 dimensions (attributes), including name of item, price of item, name/id of the customer, status and so on. The number of possible parameter settings $P_i = (p_i^1, \ldots, p_i^m)$ that control the visual encoding and therefore the number of possible images I_{P_i} is defined by the number of different attribute combinations of size m from the available number of dimensions d, given as

$$|\{I(S(P_i)) = V(D, S(P_i), t)\}| = \frac{d!}{(d-m)!}. \qquad (2)$$

For the pixel bar chart example, we may specify $m = 4$ attributes at once from the input data with 20 dimensions, which would result in 116,280 mappings. This number may be increased by additional parameters like different colormaps or different scalings. The equation above also shows that increasing dimensionality boosts the parameter space exponentially. If we have 50 attributes the number of possible mappings is 5.5 million. Then the analyst faces the problem, how to determine interesting subsets from the available data dimensions for visual analysis, that could reveal interesting relationships.

The pixnostics process model

Our pixnostics approach follows a three-step process based on the current task-at-hand:

- *Analytical filtering and pruning* of the set of possible images $\{\{I(S(P_i))\}$ by analyzing the parameter space P_i. The aim is to extract useful attribute selections and useful parameter settings automatically (candidate set CS),
- *Image analysis* of the remaining candidate set CS-generating visualizations using the determined candidate attributes
- *Ranking and output* of the candidate set CS-providing a ranking of candidate images I_{CS}

Figure 3 illustrates the pixnostics process model. In classical visual exploration, the user visually analyzes a collection of data items to find answers to various questions (analysis tasks). In our framework, an analysis task T describes conditions that the data items needs to fulfil in the resulting visualization, which is the input of the pixnostics pipeline together with the data D and the specification of the visualization method S (left side in Figure 3). Then data analysis and image analysis methods are applied to select potential interesting visualizations and present them to the user (center/right side in Figure 3). An issue for future work is to adapt these functions by integrating the user feedback in form of a relevance feedback loop.

Since the applied methods highly depend on the selected task, it is one of the major challenges to identify the most common tasks, identify their impact to unique visual properties in the resulting image I and finally to find adequate analysis functions for each of the tasks, that is to find good predictors for these properties in images. Unique properties in images are homogenous areas, color outliers, edges and segments, etc. Previous studies on visualization design proposed a range of different analysis goals and tasks.[15–19] They propose individual taxonomies of information visualizations using different backgrounds and models, so that users and analysts can quickly identify various techniques that can be applied to their domain of interest. Based on the proposed approaches, we identified the following generic tasks: identify, locate, cluster, associate, compare, correlate, match and sort whereas we focus in our initial experiments in the context of this paper mainly on cluster and outlier analysis.

In the following, we describe the individual steps in more detail.

Step 1: analytical filtering and pruning

In practice, the number of attributes is greater than the capabilities of most visualization techniques. The first step of our pixnostics approach is therefore to determine relevant relationships among the different attributes analytically. In our experiments we use classical data mining techniques and statistic measures, more precisely, correlation analysis, partial matching techniques, classification techniques and cluster analysis to accomplish this step, but of course, other task-specific analysis techniques may be used as well.

Correlation analysis Attributes that are correlated may be interesting for detailed analysis, because they may reveal relevant impact relationships. Therefore, we employ correlation analysis to find groups of correlated attributes. We determine the pair-wise global correlations among all measurements as given by Pearson's correlation matrix. Pearson's correlation coefficient r between bivariate data A_{1i} and A_{2i} with $(i = 1, \ldots, n)$ is defined as

$$r = \frac{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})(A_{2i} - \bar{A_2})}{\sqrt{\sum_{i=1}^{n}(A_{1i} - \bar{A_1})^2 \sum_{i=1}^{n}(A_{2i} - \bar{A_2})^2}}, \qquad (3)$$

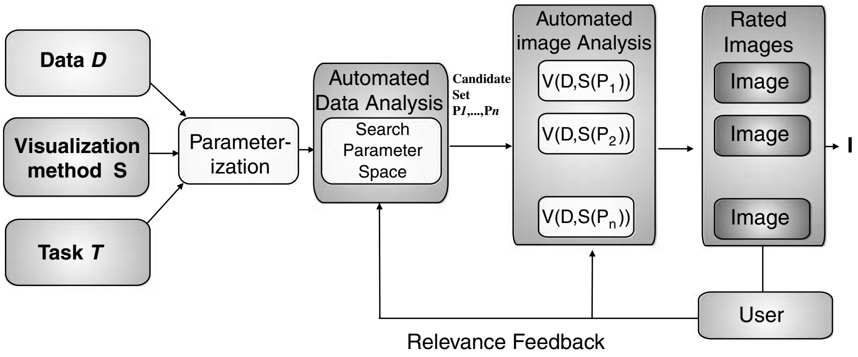

Figure 3 Pixnostics process model: combining task-dependent automated data and image analysis techniques.

where $\bar{A}_1$ and $\bar{A}_2$ are the means of the A_{1i} and A_{2i} values, respectively.

If two dimensions are perfectly correlated, the correlation coefficient is 1, in case of an inverse correlation it is −1. In case of a perfect correlation, we can omit one of the attributes since it contains redundant information.

In Figure 4 an example from the census housing data[20] on U.S. state level is shown, correlation coefficients for pairs of attributes are shown in the upper right half of the matrix, histograms in the diagonal show the data distribution. The data set contains, for example, information about U.S. education levels, crime rates, housing or household incomes on different levels of detail (country, state, county, block level). Typical exploration tasks focus on the extraction of information about housing neighbourhoods for particular areas within the U.S. including the identification of correlations between statistical parameters like household income, house prices, education levels and crime rates. The figure clearly shows that states with high total population have high gross rents (0.96) or that Median Household incomes are correlated with Median House prices per state (0.68). The analyst may now investigate such relations in more detail. In most cases, however, the correlations are not perfect and we are interested in high correlation coefficients and select sets of highly correlated attributes to be visualized.

An available alternative for adjacently depicting similar dimensions is to use the normalized Euclidean distance as a measure for global similarity Sim_{Global} defined as

$$Sim_{Global}(A_i, A_j) = \sqrt{\sum_{i=0}^{N-1} (b_i^1 - b_i^2)^2}, \tag{4}$$

where $b_i^j = (d_i^j - min(A_j))/(max(A_j) - min(A_j))$.

The global similarity measure compares two whole dimension such that any change in one of the dimensions has an influence on the resulting similarity. The defined similarity measure allows it to determine groups of similar attributes for the following visualization. Since in general, computing similarity measures is a non-trivial task, because similarity can be defined in various ways and for specific domains, special measures may be included for specific tasks.

Cluster analysis In order to perform a visual analysis, it is important to have the possibility to partition the data appropriately and then to focus on certain parts of the data. Cluster analysis can help to do this based on the characteristics of the data instances. The cluster analysis may, for example, find out that the data instance of a data set may be partitioned into different groups, which may be then independently analyzed using visualization techniques. Since attribute parameter values may be continuous (sales amount) or categorical values (item name), the clustering approach has to take these properties into account. There are a large number of clustering methods that have been proposed in the literature Hinneburg and Keim[21] presents a nice overview). In the pixnostics prototype we employed *k-means* clustering,[22] one of the most popular approaches.

Classification analysis In some applications, for example in visual root-cause analysis, the goal of the data exploration is to understand the relationship between data attributes and some specific target attribute, for example which attributes have an influence on the target attribute. The task is to find the attributes that are best predicting the outcome of the target attribute. A well-known heuristic for this task is the GINI index,[23] which is commonly used in decision tree construction.

Given a target attribute (e.g a business metric) A_T which is partitioned into a disjoint set of k classes (e.g accept, reject) or value ranges (e.g large, medium, small) denoted by $C_1, \ldots, C_k$ ($B = \bigcup_{i=1}^{k} C_i$), then the GINI index of an attribute A, which induces a partitioning of A into $A_1, \ldots, A_m$, is defined as

$$InfoGain_{GINI}(A_T, A) = \sum_{i=1}^{m} \frac{|A_i|}{|A_T|} GINI(A_i), \tag{5}$$

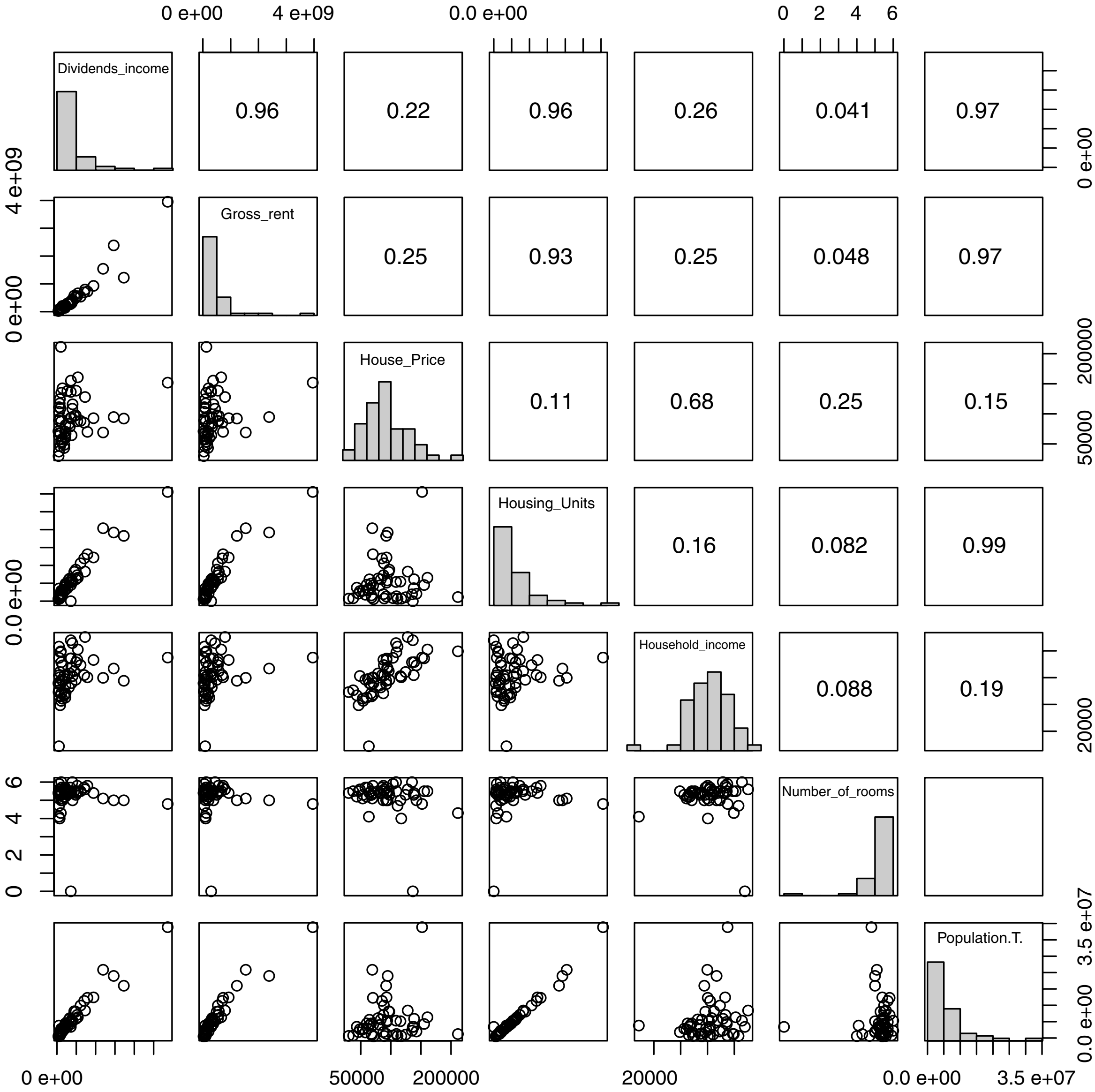

Figure 4 Identifying correlations in census housing data on U.S. state level: besides trivial correlations (e.g population and number of house units), some interesting correlations are revealed, for example, between population and gross rent (because of demand and supply effects). Highly correlated attributes may be analyzed in more detail.

where

$$GINI(A_i) = 1 - \sum_{j=1}^{k} \left[\frac{|C_j|}{|A_i|}\right]^2.$$

The *InfoGain* is determined for all attributes and attribute combinations and the attributes with the highest *InfoGain* with respect to the target attribute A_T are chosen for visualization. These attributes are best predicting the outcome of the target attribute and therefore they may be relevant for detailed analysis.

Step 2: image analysis

Once we have selected candidate parameter settings P_i, $i = 1, \ldots, max$ based on promising attribute selections where max is the number of parameter settings, we generate visualizations by computing all possible mappings of the candidate parameter set to visual variables.

We then apply image analysis methods to determine the relevance, that is, the potential value of each image (visualization), with respect to the given task. In comparison to existing semi-automated methods where the user is forced to rate the generated visualizations, the image analysis is completely automated. The goal is to process the images in order to generate some measurements of its relevance with respect to a given task T. The challenge is to find adequate image analysis functions to reach this goal. Numerous image processing operations $\sigma()$ for many different tasks exist in the literature, for example for Image Segmentation, Image Retrieval, Edge Detection, Image Denoising or Image Inpainting.[24] Many of these techniques may be very useful in visualization analysis.

In our initial experiments, we focused mainly on the information content of each resulting visualization $V(D, S(P_i))$ and employ this measure to compute the relevance of each visualization for certain tasks. A very promising way to extract such information from an image, besides well-known color histograms, is Shannon's entropy measure.[25] It is frequently used in image processing and analysis. In Koskela *et al.*[26] an entropy-based approach for image cluster analysis is proposed, a more general approach for image retrieval using entropy is introduced in Zachary *et al.*[27]

In the first step, we generate and store $I_{Pi} = I(P_i) = V(D, S(P_i))$ as a matrix U of scalars representing gray-scale values, the pixel-matrix representation with $U = (u_{i,j})$, $i \in [0, \ldots, I_{width}]$, $j \in [0, \ldots, I_{height}]$. (color images are converted to gray values). The base for our analysis is the distribution of gray values within the image. Thus, we are interested to know the pixel distribution H in certain areas of the Image as a function of gray levels g. Image histograms are an efficient way to reach this goal. The histogram of the 2D image $g(U)$ can be seen as a 1D function $H[g]$ where the independent variable is the gray value g and the dependent variable is the number of pixels H with that level. We can then use the histogram properties to make assumptions about the information contained in the image. For example, if most pixels in an image are contained in a small range of gray levels, the image can be seen as redundant since it provides little new information and thus the underlying parameter setting would not lead to insightful visualizations. If there are too many different gray levels, the image represents noise and it is not likely that it contains relevant information. An image with a bimodal histogram (i.e a histogram with two peaks) may contain clusters and may be relevant for visual exploration. Since all pixels in the image must have some gray value in the allowed range, the sum of populations of the histogram bins must be equal the total number of image pixels N:

$$N = \sum_{g=0}^{g_{max}} H(g), \tag{6}$$

where g_{max} is the maximum gray value ($g_{max} = 255$ for an 8-bit quantizer). The histogram function is equal to the scaled probability distribution function $p(g)$ of gray levels in that image:

$$p(g) = \frac{1}{N} H(g), \quad \text{with} \sum_{g=0}^{g=max} p(g) = 1. \tag{7}$$

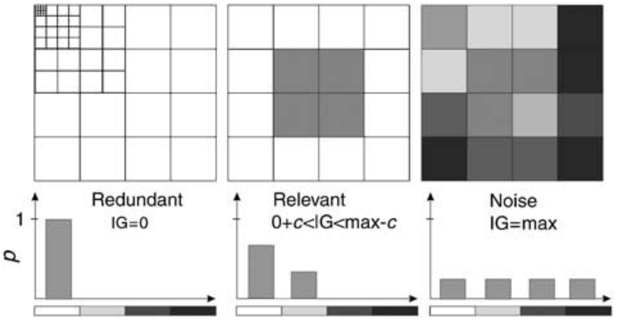

Figure 5 Information content (IG) of different gray-level images. From an analyst's point of view, interesting images should have an Information content in a certain range c between 0 and IG_{max}.

Based on the probability distribution, we can now compute Shannons Entropy, which is equal to the minimum number of bits that are required to store the image. If the probability of gray level g in the image $u(i, j)$ is represented as $p(g)$, the definition of the quantity of information in the image is

$$E(g) = -\sum_{g=0}^{g_{max}} p(g) \log_2(p(g)). \tag{8}$$

From this definition, it is easy to show that the maximum information content E is obtained if each gray-level has the same probability; in other words, a flat histogram corresponds to maximum information content. The minimum information content $E = 0$ is obtained if the image contains only one gray level. Since minimal information content means redundancy and maximum information content means information overload or noise, the interesting images should have an information content in between, for example, in a task-dependent range c shown in Figure 5.

Alternatively, we use the standard deviation *stdev* as a measure of spread of gray levels g in a given image I with N as the number of different gray levels in the image:

$$stdev(I, g) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (g_i - \overline{g})^2}. \tag{9}$$

Since we want to analyze the images not only in whole, but also find interesting local patterns in the image, we use a regular grid to separate the image in regular grid cells and than apply the methods mentioned above to compute values for each grid cell. The computation of the information content of each cell is identical to the methods described above. The only difference is that from the individual grid values we then compute a single relevance value for each image, described in the next section. To adapt our method to given application scenarios, we do not only use a fixed grid-resolution, but a hierarchy of grid

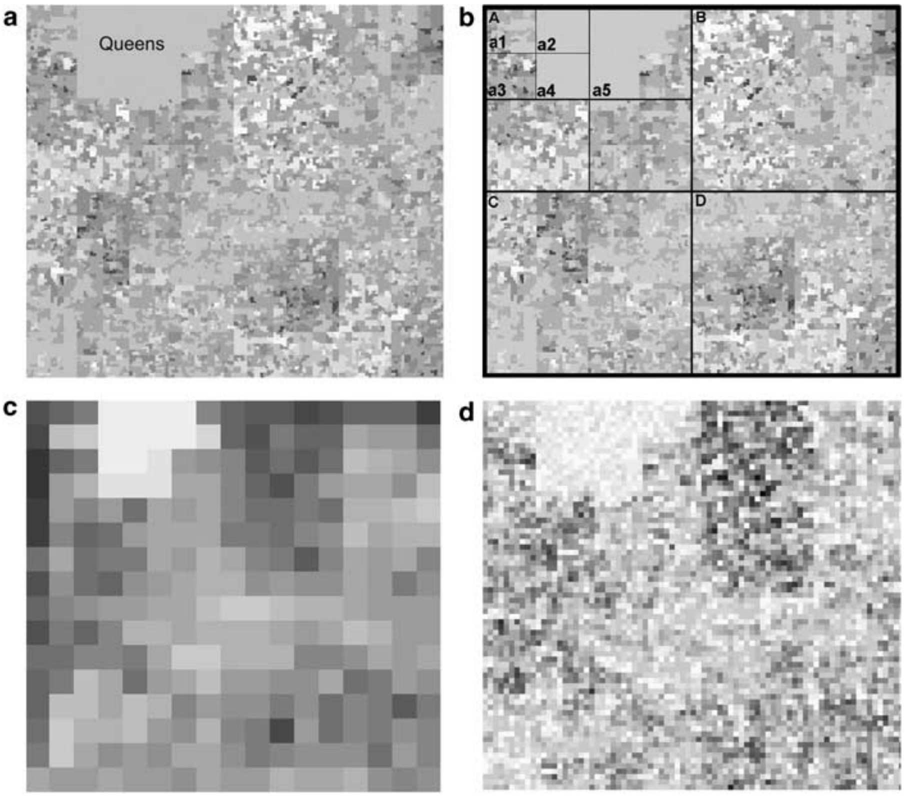

Figure 6 Basic idea of grid-based information content. Based on the entropy values for certain grid resolutions, measurements for the relevance of the image are generated. Darker gray levels correspond to higher entropy values.

cells as shown in Figure 6(b), efficiently implemented by a quadtree data structure.[28]

```
Input:  Candidate Set
        $C = \{\{A_D^1, \dots A_D^h\}, \{P_1, \dots, P_l\}\}$
with Candidate Data Attributes $A_D^1, \dots A_D^h$ $(h < n)$,
Candidate Parameter Settings $P_1, \dots, P_l$: $(l < k)$,
Visualization $V$ with Specification $S$
Performed Task $T$
Output: Ranking Scores $R(\{I(S(P_i)) =$
        $V(\{A_D^1, \dots A_D^h\}, S(P_1, \dots, P_l)\})$
Procedure Visualization Analysis
Generate $\{I_i = V(\{A_D^1, \dots A_D^h\}, S(P_1, \dots, P_l)\}$
for $i \leftarrow 1$ to $|C|$ do
    ComputeRegularGridonImage($I_i$)
    for each GridCell $GC(I_i)$ do
        ComputeEntropyValues $f(GC(I))$
    end
    for all Entropy values $f(GC(I_i))$ do
        $R =$ComputeRankingScore($f(GC(I_i)), T$)
    end
end
return Ranking Scores $R(I_i)$
```

Algorithm 1 Entropy-based image analysis

Step 3: ranking and output to the user

Having a function *f*, such as $E, stdev$, which measures the information content of an image, we can now compute rankings from the candidate parameter sets with respect to a given user task *T*. To show the basic concepts of our approach, we focused two major tasks that are common in most analyzes processes, namely outlier analysis, including the search for local outliers or values of interest (e.g find all counties or cities that have similar household income or unexpected household income) and cluster analysis (e.g find areas with similar statistical parameters). In our prototype framework, we provide ranking functions for both tasks and show how we applied it to real-world data sets. Of course, the user may also use other ranking functions for specific tasks, which can be easily integrated into the pixnostics framework.

Computing the global ranking score To determine the relevance of each visualization, we have to compute a global ranking score from the grid cell values of each image. Besides simple aggregation functions like average or stdev, which have shown to be very useful to prune a large number of irrelevant visualizations, we employed several more sophisticated functions.

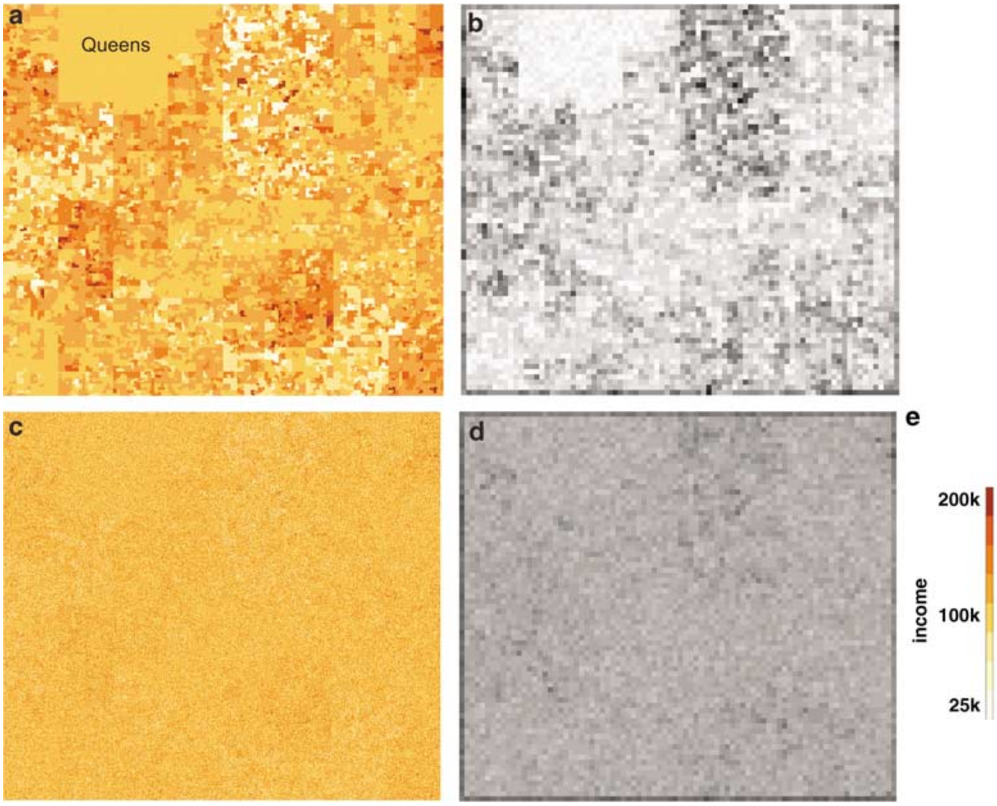

Figure 7 Visualization of information content: jigsaw maps are generated from NY median household income, darker colors correspond to higher income. Gray levels show the information content of image sections, darker gray levels correspond to higher information content. The permutated image has significant higher information content, which indicates bad clustering properties.

For cluster analysis for example, we initialize each regular grid cell $GC(I_i)$ with its information content score $f(GC(I_i))$. Then we start to merge regular grid cells $GC_k(I_i)$ and $GC_l(I_i)$ that have similar content scores, to larger cells. The new content score $f'(GC(I_i))$ is determined using local term weighting $f(GC_{common}(I_i)) = l(f(GC_k(I_i)) + f(GC_l(I_i))$. The term weight function l is defined over the set of all regular grid cells $\{GC(I_i)\}$. This allows us to investigate clustering properties, where we are looking for images with higher information content f on coarser grid resolutions and with lower information content f at finer grid resolutions. The technique is similar to single linkage clustering. We order on each resolution level the grid cells according to their value f, starting with the finest resolution. Then we expand grid cells with similar low content scores and sum up their weight while enlarging the grid cells. Since the spread of gray levels is typically higher if the size of the grid cell increases, we use cell size as an weighting factor. Note that the local term weight l directly depends on the given task T. That means, the user just needs to provide useful weight functions to get a relevance measure of the image for the given task. Well-known and widely used weight functions are binary operators, logarithmic or augmented normalized term frequency.

A second approach we employed for the analysis of cluster properties is the *BB* Score (or BlackBlack count),[29] which is commonly used in Geo-Analysis. This approach takes the position (i, j) and (Entropy) value of each grid cell as input and returns a measure that indicates if adjacent cells have similar values. Images where adjacent cell have similar values have a higher *BB* score than images where they have different values. Therefore, we rank images with higher scores higher, since they may provide better cluster properties.

In outlier analysis, the goal is to compute a ranking of a collection of images in such a manner that images showing outliers should have higher global ranking scores. The inverse normalized term frequency is a common choice for outlier analysis. The normalized term frequency l is defined as the logarithm of the sum of two information content scores $f(GC_k(I_i))$ and $f(GC_l(I_i)$ normalized by the total number of regular grid cells $|\{GC(I_i)\}|$.

$$l(f(GC_k(I_i)) + f(GC_l(I_i)) = \log\left(\frac{|\{GC(I_i)\}|}{f(GC_k(I_i)) + f(GC_l(I_i)}\right).$$

Evaluation and application

To show the usefulness of our approach, we applied the pixnostics technique to generate jigsaw maps and pixel

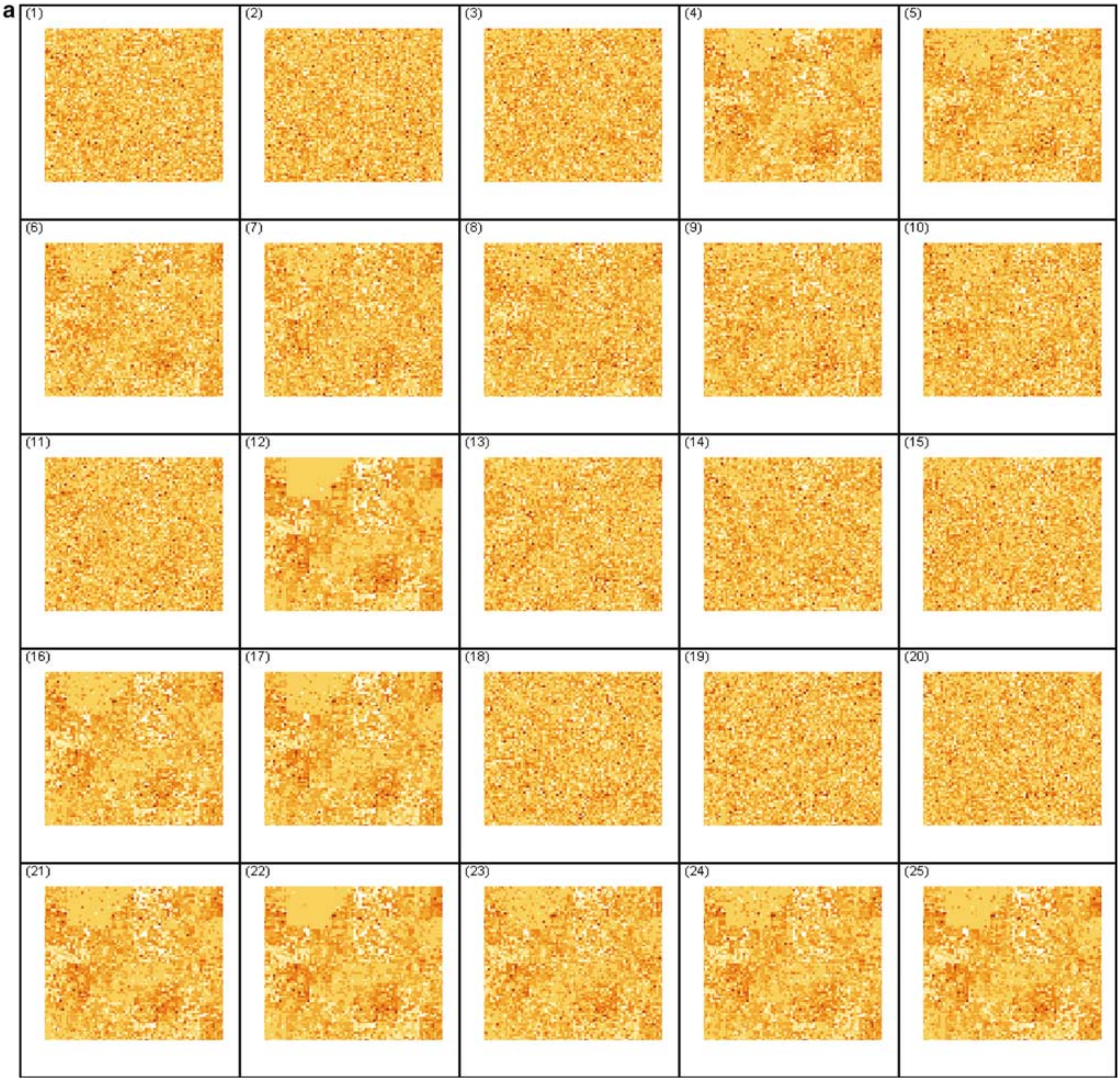

Figure 8a Census data jigsaw unsorted (8a) and sorted (line by line starting at top left corner with most relevant) by ranking function based on entropy and clustering task (8b).

bar charts. The proposed experiments show how Pixnostics can steer the visual exploration process in an unsupervised manner, to increase the efficiency of the exploration process and to actively support the analyst to reduce the effort of getting insight from the data.

Census data jigsaw maps

Our first application example analyzes U.S. census data, in particular median household income for the state of New York on block level. We generated visualizations using the jigsaw maps,[14] a pixel-based technique based on space filling curves. The basic idea is to map the census data into the 2D plane in such a way that properties like locality and clusters in the data are preserved by using a space filling curve. To verify our proposed techniques, we generated a jigsaw map from the New York state census median household income data on block level which should preserve the clusters in the data (clusters of areas with high/low income) and their spatial location shown in Figure 7(a). Then we permutate the data points at different permutation rates. This should of course destroy, or

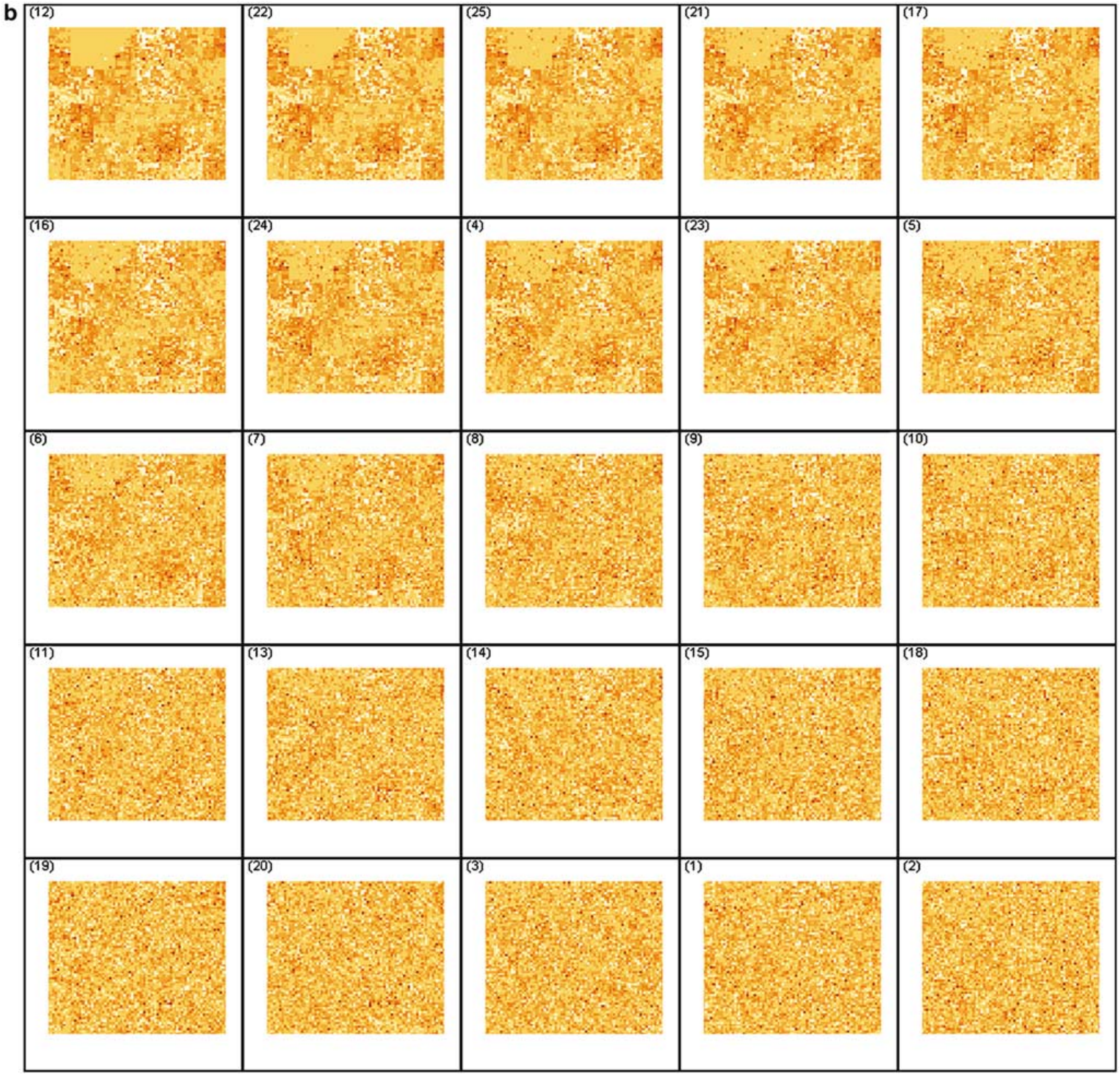

Figure 8b (*continued*).

at least reduce the clustering/locality properties. Now we apply our automated analysis function based on the clustering task, that ranks the underlying figures according to their clustering properties. The original jigsaw should of course be the image with the highest rank, since it provides the best clustering properties. The more permutation in the image, the lower should the relevance of the image, that is its rank, be.

Thus, we consider the permutation rate as our input parameter and want to find input parameters which produce visualizations with good clustering properties. Figure 8 shows the experimental results. The upper figure shows the unordered input data set, a set of jigsaw images. It is easy to visually identify images with good clustering properties, that is, images having a cluster with low income in the upper left corner surrounded by high income areas. In the lower figure, the result after the analysis step is shown. It is easy to see that figures with good clustering properties are ranked first, while images containing more noise have lower relevance. To determine the ranking, either the entropy or the Standard deviation of the pixel grey levels in combination with the regular grid cell hierarchies are employed.

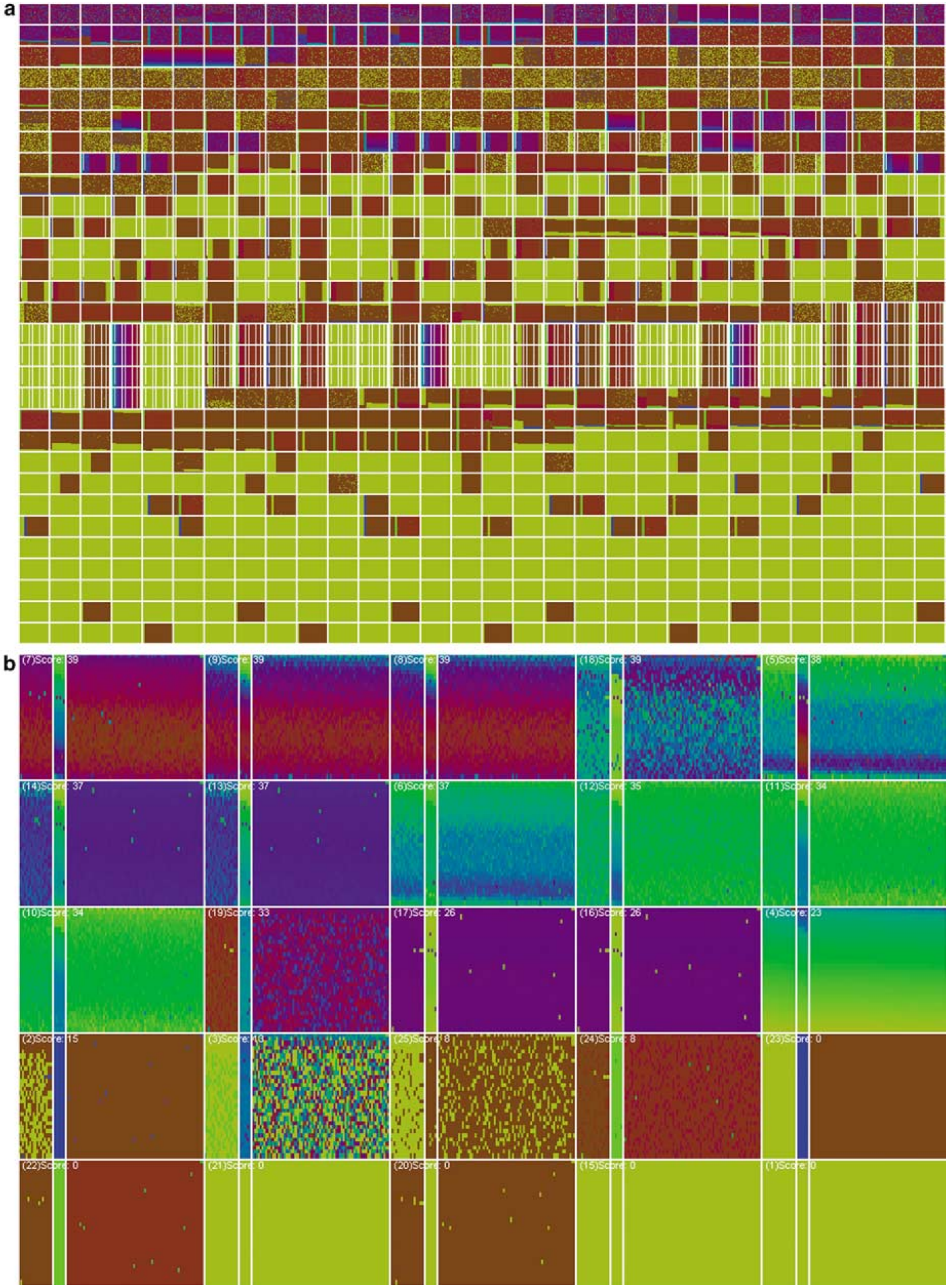

Figure 9 Pixel bar chart showing top 25 results after image analysis using entropy measure. It is easy to see that the bar in the middle ('reject' parts) show significant differences in comparison to the two other bars. (a) Ranking of images generated from promising attribute selections. Each box shows a thumbnail of a PixelBarChart based on a certain attribute selection, ordered by information content (desc). (b) Ranking after image analysis: the top 25 images are shown, with a fixed target attribute as splitting attribute for the bars.

Figure 7 shows the rationale for the ranking. An image that provides a good clustering has areas with very low Entropy or low Stdev of gray levels while the complete figure does not necessarily have low Entropy. Therefore, we start with a fine grid and determine the information content of each cell like shown in Figure 7(b). Then we hierarchically compare neighboring grid cells similar to single linkage clustering and try to extend clusters. Finally, we aggregate the information content of the clusters and order the images according to their information content values.

Pixel bar charts

Pixel bar charts[2] are derived from regular bar charts. The basic idea of a pixel bar chart is to present the data values directly instead of aggregating them into a few data values by representing each data item by a single pixel in the bar chart.

The detailed information of one attribute of each data item is encoded into the pixel color and can be accessed and displayed as needed. To arrange the pixels within the bars one attribute is used to separate the data into bars and then two additional attributes are used to impose an ordering within the bars along the *x* and *y* axes. The pixel bar chart can be seen as a combination of traditional bar charts and *x*–*y* diagrams.

Although pixel bar charts have been successfully applied to explore large data sets (see Keim *et al.*[30]), the analyst has to choose selections of attributes for separation, ordering and color coding of data points from the underlying data manually, according to his analysis tasks.

On the one hand, this is time consuming since he has to try multiple parameter settings even those that do not reveal interesting patterns, on the other hand he may overlook interesting patterns since only a few attribute combinations can be analyzed manually. To face this problem, we applied pixnostics to pixel bar charts, to guide the analyst through the exploration process and indicate potentially interesting parameter settings.

We applied our approach to a production data example. The data set contains data from an assembly line, in particular measurements from different stages of the assembly line like cast temperatures, part measurements and the quality of the output. All in all the data set contains 22 attributes. The output parts are classified into three groups: accept, reject, rework. Parts that are grouped 'accept' pass the quality check, 'rework' parts need to be reworked to pass the quality check and 'reject' parts must be rejected because of defects. The analysis of such data is an important task in order to reduce rejected parts and thus to reduce production cost. Using pixel bar charts, the analyst faces to problem of how to find groups of attributes that may influence the quality of the output. There are 175,560 combinations possible to choose 4 attributes as visual variable from 22, even if the target attribute 'Quality' is fixed for separation of the bars, there are still over 9000 combinations for selection of three attributes out of 22, which cannot be checked manually.

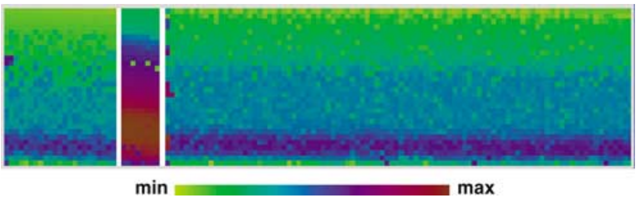

Figure 10 Single bar chart constructed from the output result of Pixnostics (from the Chart in the upper right corner in Figure 9(b).

Therefore, we first apply our automated analysis tools to determine attributes that most influence the 'Quality' variable, using correlation and classification analysis. Of course, we can additionally prune all parameter settings where 'quality' is not involved, since these will not bring us any new insight.

From the remaining combinations we either generate images and order them by information content directly, as shown in Figure 9(a) or we can filter pixel bar charts where the target attribute is fixed as splitting attribute and select the most valuable ones from them, as shown in Figure 9(b).

The figure shows the 25 most relevant pixel bar charts having 'quality' as splitting attribute. Note that the left bar shows parts that are 'rework', the middle bar shows 'reject' parts and the right bar show 'accept' parts. It is easy to see that the 'reject' bars look significantly different than the rest. The analyst may now select a single image from the provided images, and a pixel bar charts is created from this selection as shown in Figure 10.

The analyst can now easily discover relevant patterns by visual-based root cause analysis. In the image, the color shows the temperature of a particular casting mold and the ordering in *y* direction shows the duration of the part at this stage. It is easy to see that the casting mold had a significantly higher temperature for 'reject' parts, which is a potentially reason for a damaged part.

In this manner, the analyst may investigate further high ranked images, which provides a more efficient way of visual analysis than manual feature selection.

Conclusion and future work

Integrating automated analysis methods into the visual exploration process is an important challenge in the age of massive data sets and has been recognized as a major research area in the context of visual analytics. Therefore, the aim of this paper is to show how unsupervised analysis functions can help to speedup the visual exploration process by supporting the user with task-driven relevance functions for a more effective data analysis. The basic idea of the proposed method is to measure the relevance of the resulting visualization with respect to input parameters and user tasks and to provide a ranking of potentially useful initial visualizations and initial parameter settings. This helps the analyst to focus on relevant parts of the data and relevant parameter settings and leads to an improved exploration process. We provided a formal definition of our work and showed how the technique can be used with jigsaw maps and pixel bar charts.

Future work will focus the improvement of the proposed technique and its application to a variety of visualization techniques, not only pixel based but also geometric and iconic techniques. Furthermore, we will include the user in the analysis process in form of relevance feedback to dynamically adapt our relevance functions. An important issue for future work is also the evaluation of the proposed methods, since although our experiments show that the concept has a great potential, real user studies that evaluate the quality of our relevance functions are still pending and are subject of future research.

Acknowledgements

This article is based on *Pixnostics*: *Towards measuring the value of visualization* by J. Schneidewind, M. Sips and D.A. Keim[4] which appeared in *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology 2006 (VAST 2006)*.©2006 IEEE. This work was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC).

References

1 Thomas JJ, Cook KA. Illuminating the path: research and development agenda for visual analyics. *IEEE Computer Society*: Los Alamitos, 2005.

2 Keim DA, Hao M-C, Dayal U, Hsu M. Pixel bar charts: a visualization technique for very large multi-attribute data sets. *Information Visualization* 2002; **1**: 20–34.

3 Spence R. *Information Visualization*. ACM Press Books, Pearson Education Ltd.: UK, 2001.

4 Schneidewind J, Sips M, Keim DA. Pixnostics: towards measuring the value of visualization. In: *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST 2006)* (Baltimore, MA), October 2006.

5 House DH, Bair A, Ware C. On the optimization of visualizations of complex phenomena. In: *IEEE Visualization* 2005 (Minneapolis, MN), 2005; 12.

6 Sims K. Artificial evolution for computer graphics. In: *SIGGRAPH '91: Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques* 1991 (New York, NY, USA), ACM Press: New York, 1991; 319–328.

7 Greenfield GR. Computational aesthetics as a tool for creativity. In: *C&C '05: Proceedings of the 5th Conference on Creativity & Cognition* 2005 (New York, NY, USA), ACM Press: New York, 2005; 232–235.

8 Wilkinson L, Anand A, Grossman R. Graph-theoretic scagnostics. In: *INFOVIS '05: Proceedings of the 2005 IEEE Symposium on Information Visualization* 2005 (Washington, DC, USA), IEEE Computer Society: Silver Spring, MD, 2005; 21.

9 Tukey JW. *Exploratory Data Analysis*. Addison Wesley Publishing: Reading, MA, 1977.

10 Tukey JW, Tukey PA. Computing graphics and exploratory data analysis: an introduction. *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics 85*, (Fairfax, VA) National Computer Graphics Associations, 1985.

11 Bertin J. *Semiology of Graphics*. University of Wisconsin Press: Madison, WI, 1983.

12 van Wijk JJ. The value of visualization. In: *Proceedings of IEEE Visualization*, 2005 (Minneapolis, MN), October 2005; 79–86.

13 MacKinlay J. Automating the design of graphical presentations of relational information. In: *Readings in Information Visualization*, 1999 (San Francisco, CA, USA), Morgan Kaufmann: Los Altos, 1999; 66–81.

14 Wattenberg M. A note on space-filling visualizations and space-filling curves. *In*: Proceedings of the 2005 IEEE Symposium on Information Visualization Infovis 2005 (Minneapolis, MN), 2005; p 24.

15 Shneiderman B. The eye have it: a task by data type taxonomy for information visualizations. *Proceedings of the IEEE Conference on Visual Languages*, 1996, (Boulder: CO), 1996; 336–343.

16 Keller PR, Keller MM. *Visual Cues—Practical Data Visualization*. 1st edn. IEEE Press: New York, 1993.

17 Casner SM. Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics* 1991; **10**: 111–151.

18 Chi Hh. (Ed). A taxonomy of visualization techniques using the data state reference model. In: *INFOVIS* 2000; 69–76.

19 Keim D. Designing pixel-oriented visualization techniques: theory and applications. *Transactions on Visualization and Computer Graphics* 2000; **6**: 59–78.

20 United States Department of Commerce. US Census Bureau website. http://www.census.gov, September 2003.

21 Hinneburg A, Keim DA. Clustering techniques for large data sets from the past to the future. In: *KDD '99: Tutorial Notes of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1999 (New York, NY, USA), ACM Press: New York, 1999; 141–181.

22 Kanungo T, Mount D, Netanyahu C, Piatko R, Silverman A, Wu D. An efficient *k*-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002; **24**: 887–892.

23 Zytkow JM. Types and forms of knowledge (patterns): decision trees. Klösgui W and Zytkow JM (Eds). *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, Inc: New York, NY, U.S., 2002.

24 Mann S. *Intelligent Image Processing*. Wiley and Sons: New York, 2001.

25 Esteban MD, Morales D. A summary of entropy statistics. *Kybernetika* 1995; **31**: 337–346.

26 Koskela M, Laaksonen J, Oja E. Entropy-based measures for clustering and some topology preservation applied to content-based image indexing and retrieval. *17th International Conference on Pattern Recognition (ICPR'04)* 2004 (Cambridge, U.K.) Vol. 2, 2004; 1005–1009.

27 Zachary J, Iyengar SS, Barhen J. Content based image retrieval and information theory: a general approach. *Journal of American Society Information Science Technology* 2001; **52**: 840–852.

28 Finkel RA, Bentley JL. Quad trees: a data structure for retrieval on composite key. *Acta Informatica* 1974; **4**: 1–9.

29 Davis JC. *Statistics and Data Analysis in Geology*. John Wiley & Sons Inc.: New York, NY, USA, 1973.

30 Keim DA, Schneidewind J, Sips M, Panse C, Dayal U, Hao MC. Pushing the limit in visual data exploration: techniques and applications. In: Günther A, Kruse R, Neumann B (Eds.) *Advances in Artificial Intelligence, 26th Annual German Conference on AI, KI 2003* September 15-18 (Hamburg Germany), Lecture notes in Artificial intelligence, vol. 2821, Berlin, 2003.