DICON : Interactive Visual Analysis of Multidimensional Clusters

Nan Cao, David Gotz, Jimeng Sun and Huamin Qu, Member, IEEE

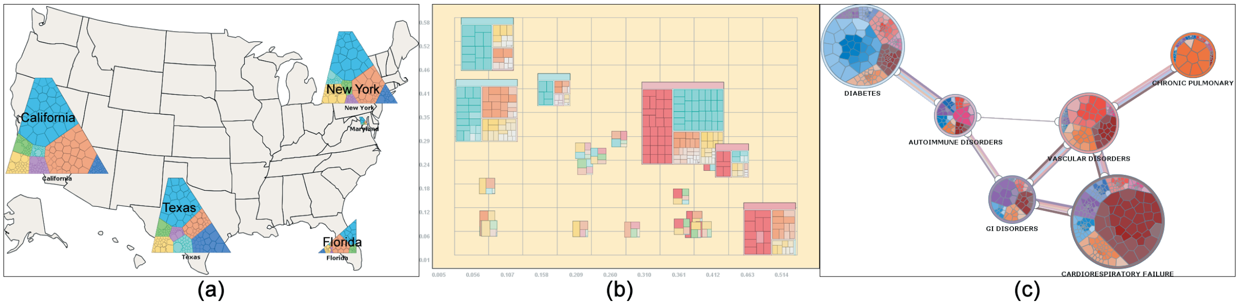


Fig. 1. DICON is a dynamic icon-based visualization technique that helps users understand, evaluate, and adjust complex multidimensional clusters. It provides visual cues describing the quality of a cluster as well as its multiple attributes, and can be embedded within many kinds of visualizations such as maps, scatter plots, and graphs.

Abstract— Clustering as a fundamental data analysis technique has been widely used in many analytic applications. However, it is often difficult for users to understand and evaluate multidimensional clustering results, especially the quality of clusters and their semantics. For large and complex data, high-level statistical information about the clusters is often needed for users to evaluate cluster quality while a detailed display of multidimensional attributes of the data is necessary to understand the meaning of clusters. In this paper, we introduce DICON, an icon-based cluster visualization that embeds statistical information into a multi-attribute display to facilitate cluster interpretation, evaluation, and comparison. We design a treemap-like icon to represent a multidimensional cluster, and the quality of the cluster can be conveniently evaluated with the embedded statistical information. We further develop a novel layout algorithm which can generate similar icons for similar clusters, making comparisons of clusters easier. User interaction and clutter reduction are integrated into the system to help users more effectively analyze and refine clustering results for large datasets. We demonstrate the power of DICON through a user study and a case study in the healthcare domain. Our evaluation shows the benefits of the technique, especially in support of complex multidimensional cluster analysis.

Index Terms—Visual Analysis, Clustering, Information Visualization

1 Introduction

Clustering is a widely used method to group data entities into subsets called clusters such that the entities in each cluster are similar in some way. A powerful feature of clustering algorithms is that they can generate clusters without any pre-defined labels or categories, which makes them an ideal choice for analyzing data with little or no *a priori* information. Unlike classification in which categories with clear semantics are pre-defined, clustering by definition works without these initial constraints on how data entities should be grouped. Users are only required to choose a distance function (e.g., Euclidean distance) that measures how similar two data items are in a feature space, and some other parameters such as the number of clusters or a maximum cluster diameter. Clustering algorithms will then automatically partition data. While this technique is powerful, users often have difficulty understanding the semantic of the resulting clusters and evaluating the quality of the results, especially for multidimensional data.

There are several issues which make understanding and evaluating clustering results difficult. First, for multidimensional data, the entities that are grouped together are close in a multidimensional feature space. However, their similarity may be mainly because of their closeness on a subset of dimensions instead of all dimensions. Understanding these abstract relationships can be challenging. Moreover, a cluster may contain several different subclusters which have different meanings for users. This subcluster structure is usually hard to detect. Second, as unsupervised learning processes using no semantic knowledge or pre-defined categories, clustering algorithms often require users to input some parameters in advance. For example, users must provide the number of clusters (i.e., k) for the well known K-means algorithm. However, it challenging to select a proper k value for the underlying data. Therefore, algorithms like K-means might group together entities that are semantically different (when k is smaller than the real number of clusters) or separate entities that are semantically similar (when k is larger than the real number of clusters). Thus, users need some way to evaluate and refine the clustering results.

Information visualization can be of great value in addressing these issues. For example, techniques such as scatter plot matrices [10], parallel coordinates [18], and RadViz [25] have been used to visually explain the results of clustering algorithms. Some algorithms focus on revealing the multi-attribute values of clusters to help users understand the semantic of clusters while others provide visual cues for the cluster quality. However, none of them offer a complete solution for cluster interpretation, evaluation, and refinement. We need a visualization which allows users to understand the meaning of various clusters, evaluate their qualities, compare different clusters, and refine clustering results as necessary. In addition, given the wide range of applications of clustering, we want a visualization that can be conveniently embedded into various visual displays or presentations.

In this paper, we propose DICON, a dynamic icon-based visualization technique that helps users understand, evaluate, and adjust complex multidimensional clustering results. DICON encodes the raw da-

- *Nan Cao and Huamin Qu are with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. E-mail: {nancao, huamin}@cse.ust.hk*
- *David Gotz and Jimeng Sun are with IBM T.J. Watson Research Center. E-mail: {dgotz, jimeng}@us.ibm.com*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

ta values in multiple dimensions as well as the statistical information related to cluster quality. It adopts an icon design which can be conveniently embedded into a wide range of presentations. Moreover, it supports intuitive user interactions for cluster refinement. The major contributions of this paper are as follows:

- A multidimensional cluster icon design that encodes multiple data attributes as well as derived statistical information for cluster interpretation and quality evaluation.
- A stabilized icon layout algorithm that generates similar icons for similar clusters for cluster comparison.
- Intuitive user interactions to support cluster refinement via direct manipulation of icons.

2 Related Work

This section provides an overview of related work. We focus on techniques most relevant to DICON, including treemap visualization, visualization of multidimensional clusters, icon-based multivariate visualization, and interactive cluster exploration and analysis. More general surveys can be found in [21,23].

2.1 Treemaps

The Treemap [19,31] is one of the most well known techniques for visualizing hierarchical information. Many algorithms [2,3,32,37] have been proposed to lay out treemaps according to different optimization criteria. The design of DICON is inspired by these techniques. However, when compared with traditional treemaps, DICON has the following significant differences. First, DICON introduces a novel encoding method which targets on the visualization of multidimensional clusters and associated statistical measures rather than hierarchial information. Second, a stabilized layout algorithm is proposed to generate similar cluster icons according to cluster similarities. Finally, several new interactions are designed to facilitate interactive cluster refinement and manipulation.

2.2 Visualizing Multidimensional Clusters

Parallel coordinate plots (PCPs), introduced by Inselberg [18], are widely used for multidimensional data. Many researchers have focused on finding innate cluster patterns over multiple dimensions. For example, Fua et al. [12] used hierarchical clustering and proposed a variation on PCPs to convey aggregate information for the resulting clusters. Novotny [26] represented each cluster as a polygonal area and used both opacity values and textures to distinguish different clusters. Zhou et al. [40] introduced visual clustering to reduce edge clutter in PCPs. Many other clutter reduction and pattern enhancement methods have been proposed including dimension reordering [1, 34] and smooth, bundled curves [39].

However, as studied by Holten et al. [17], PCPs have a limited capability when visualizing multiple clusters. If the number of clusters is over five, a rather small number, user performance in cluster identification tasks decreases dramatically. As shown in our user study, DICON overcomes this limitation and enables effective cluster identification for much larger numbers of clusters.

The scatter plot is another well known visualization technique. It is simple in design, very familiar to users due to its long history, and has a high degree of visual clarity [35]. Typical scatter plots depict data distributions across two dimensions. For multidimensional data, scatter plot matrices can be used. Scatter plot matrices [4] represent all pairwise combinations of dimensions to provide an overview of an entire dataset. However, finding multidimensional clusters using a scatter plot matrix is tedious and time consuming. Some dimension reduction methods such as Principal Component Analysis, Multidimensional Scaling and RadVis [25] can project multidimensional data onto a 2D plane where data clusters can be more easily identified using distance measures. Unfortunately, it is difficult for users to understand the semantics of the resulting clusters. DICON is designed to leverage the advantages of both traditional scatter plots and dimension reduction algorithms to provide more effective cluster interpretation capabilities. Another problem of scatter plot technique is the unavoidable overlaps. HeatMaps [6, 9, 11] has been designed to tackle this problem. It uses 2D tables with color-coded cells to identify meaningful correlations. Despite their utility in many scenarios, these techniques cannot effectively convey all attributes of multidimensional data. In contrast, DICON is designed specifically to handle multidimensional clusters.

2.3 Icon-based Multivariate Visualizations

Icon or glyph-based designs have been studied for many years. For example, Chernoff faces [5, 35] were proposed in the 1970s and use human facial features to encode multiple data dimensions with a single icon. Similarly, stick figure techniques [28] employ relatively simple icon designs where data values are mapped to visual features such as angle, length and thickness. Post et al. [29] proposed 3D iconic techniques for feature visualization. Keogh et al. [24] proposed to use colored bitmap to encode time series data.

Other designs like [22, 38], depict individual feature values using colored rectangular cells or pixels. The cells/pixels can then be packed together into an icon using various layout arrangements. These approach are perhaps most similar to the DICON design. However, these existing designs often obscure the semantic of a cluster. DICON introduces a design that conveys more cluster information, helping users better understand, compare, and adjust multidimensional clusters.

2.4 Interactive Cluster Exploration and Analysis

Many visualization tools leverage the power of rich interactivity to facilitate cluster exploration and analysis. For example, Henry et al. proposed NodeTrix [15] which combines a matrix representation for graphs with traditional node-link graph visualization techniques. Users can select and group nodes to generate an adjacency matrix view to highlight relational patterns. Seo et al. designed HCE [30] for hierarchical multidimensional cluster analysis. Elmqvist et al. developed an interactive scatter plot matrix [10] which leverages animated transitions to smoothly switch between different user selected dimensions. In these systems, interactivity provides an important role in allowing users to perform exploratory visual analysis. DICON follows a similar approach and includes its own powerful interactive capabilities that allow users to compare and refine clusters as they explore their data.

3 Visualization Design

In this section we first present several design guidelines that influenced our development of the DICON technique. We then provide a detailed description of DICON's visual encoding methodology. Finally, we introduce a number of interaction features for cluster manipulation.

3.1 Design Guidelines

Motivated by the challenges of cluster interpretation, evaluation, and comparison, we identified a few key design guidelines to follow during the development of DICON.

A cluster's visual representation should present different levels of granularity. Clusters contain information at several scales, ranging from specific entity data features, to individual entities, to overall clusters. An effective visual representation must convey each of these levels of detail. DICON adheres to this guideline by converting clustered data into an entity-feature-cluster hierarchy and using a treemap-based technique to represent them. Connections between features for a single entity are preserved via interactive highlights.

A multidimensional cluster's representation should employ consistent encodings across entity dimensions and scales. A cluster icon should uniformly apply visual encoding techniques across data dimensions and scales so that users can smoothly navigate across data dimensions and to reduce visual complexity. DICON uses the same encoding technique, based on the size and color of areas, to represent all feature dimensions. This approach is repeated at the cluster level, providing a consistent representation across scales.

Icons for similar clusters should appear visually similar while dissimilar clusters should have icons that are easily distinguishable. Icons should provide at-a-glance representations that allow users to easily determine which clusters are different and which are similar. This design requirement is critical for both cluster identification and comparison tasks. DICON satisfies this design guideline by using

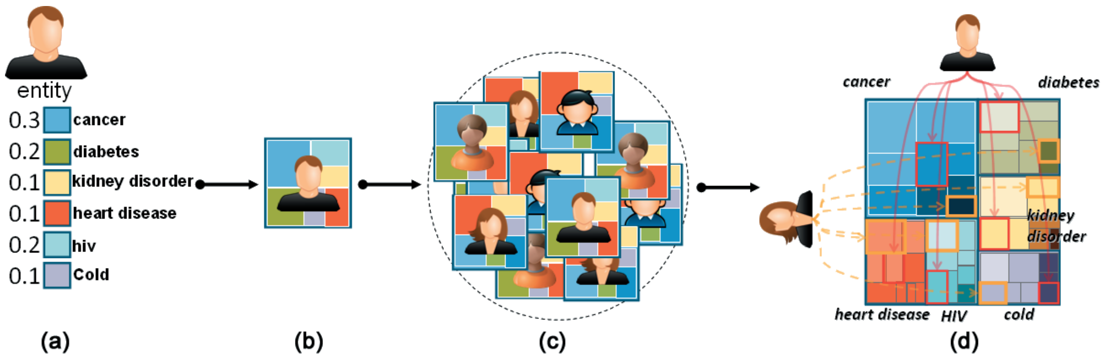


Fig. 2. Visual encoding for a patient dataset. In this encoding, an individual entity is described by a feature vector. Each feature in the vector is a numerical value depicted by a small cell. The cells are packed together to generate an individual icon. Individual icons are grouped together by splitting and re-grouping their features into categories.

a novel stable layout algorithm. This algorithm maintains consistent feature locations both within and across icons.

The visual representation should allow users to interactively manipulate clusters for refinement and exploration. Users should be able to select clusters to be merged, select entities to be removed from a cluster, and select individual clusters for subdivision into finer grained sets. All changes in cluster membership should be visually reflected in a stable manner to maintain a user's mental map as much as possible. DICON satisfies this guideline by providing a number of interactive cluster refinement features.

3.2 Visual Encoding

Following the design guidelines listed above, we designed DICON, a dynamic icon-based visualization technique which represents clusters of multidimensional data as compact glyphs. As multidimensional clusters naturally contain information at multiple scales, we adopt a treemap-like visual encoding scheme. Our icon design, summarized in Fig. 2, uses a combination of spatial size, position, shape and color to convey key cluster properties.

Size Encoding. Generally speaking, an n-dimensional data cluster contains a number of *entities*, each of which is described by a set of *features*, noted as $F = f_0 \dots f_n$. For example, Fig. 2(a) depicts an entity from a healthcare dataset which corresponds to a single patient's medical record. That record contains six features, including severity scores for various co-morbidities such as cancer and diabetes. DICON requires quantitative features and the visual encoding process begins by globally normalizing the range of all features to the interval $[0,1]$. This enables the comparison of multiple features regardless of scale. An optional local normalization step is performed on each entity such that the total value of all features equals one (i.e., $\sum_{i=0}^{n} f_i = 1$). The feature values are then mapped to color-coded cells whose sizes represent these values. As depicted in Fig. 2(b), the cells are packed together to form an iconic representation of the entity. A group of entities are further packed as clusters as shown in Fig. 2(d). If performed, the local normalization step produces an icon in Fig. 2(b) with a total area of one unit. This makes cluster icon sizes correspond to the number of items in a cluster, but makes feature value comparison across entity sizes difficult. Thus, color opacity can be used to display actual non-normalized values, or when cluster icon size is less important the optional local normalization can be skipped.

Position and Shape Encoding. When a set of entities are grouped together into a cluster, as illustrated in Fig. 2(c), the entity icons must be combined into a single aggregate iconic representation. We generate a cluster icon by (1) splitting each entity icon into individual feature cells, (2) regrouping the feature cells by feature type, and (3) packing the regrouped cells into a single overall cluster icon. One example is shown in Fig. 2(d) which uses the traditional treemap layout to create a cluster icon. A more advanced implementation is also proposed. With the help of a stabilized Voronoi layout described in Section 4.1.2, it embeds cluster kurtosis and skewness using an icon's shape to simulate the entity distributions within the cluster as illustrated in Fig. 3 and Fig. 4. The width of the icon's top edge encodes kurtosis (the ***"Peakness Cue"***) while the horizontal position of the peak of the icon encodes skew (the ***"Asymmetry Cue"***).

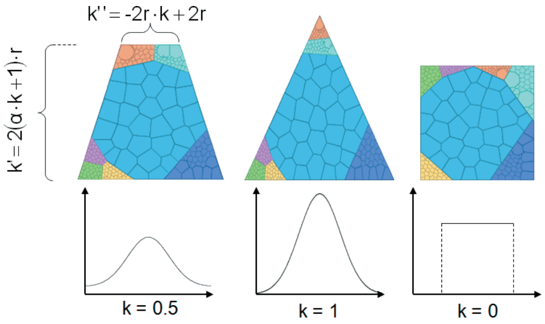


Fig. 3. Encoding the normalized kurtosis k using icon shape. The shape of the icon intuitively shows the distributions of underlying data.

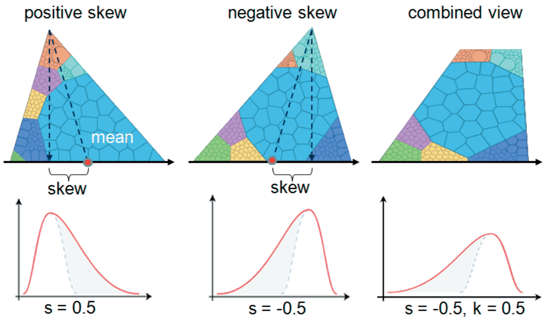


Fig. 4. Encoding the normalized skew using icon shapes. The first two figures are the icons with different skew values. The third one is an icon generated by combining both kurtosis and skew together.

It should be noted that shape as a preceptive visual property provides high efficiency for cluster comparison in multiple scales. For example, Fig. 1(a) clearly shows that all of the depicted state clusters have similar distributions except for Florida, which has greater peakness and opposite asymmetry. Unfortunately, the irregular icon shapes can make precise size comparisons more difficult. DICON thus allows users to turn this feature on or off as needed during their analysis.

Color Coding. By default, each cell in a cluster icon is rendered using the color assigned by its corresponding feature. For instance, all "cancer" cells in Fig. 2 would be rendered in the same shade of blue. In addition, three other color schemes are provided to convey various

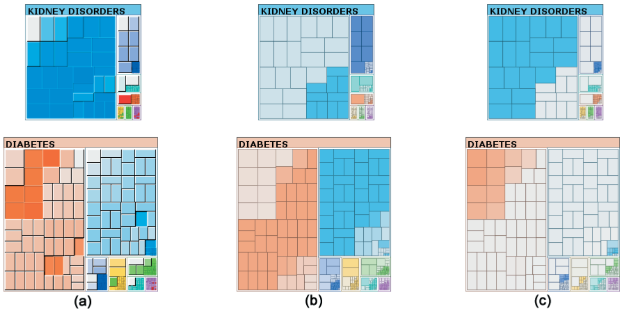


Fig. 5. Color encoding and visual cues in two example clusters: (a) Cluster quality cue; (b) Co-occurrence cue; (c) Domination cue.

visual cues. They are critical for some tasks, but also introduce visual complexity. Thus, we allow users to switch between color schemes.

The first scheme uses cell color saturation to convey globally normalized feature values to facilitate quantitative comparison over clusters in case the cell size encodes the locally normalized values.

Our second scheme is designed to depict cluster quality. Feature variance is selected as the quality measure and encoded by shading as illustrated in Fig. 5(a). Multi-level cushion shading is also used to ease interpretation even for small icons. For a given cell, feature variance is positive if the feature value is larger than the cluster mean and negative if smaller than the mean. Positive and negative variances are depicted using rising and sinking shades, respectively. Under this scheme, high quality clusters appear more homogeneous which provides a clear ***"Cluster Quality Cue"***. An ***"Outlier Cue"*** is also represented under this scheme. Outliers have large variances from the cluster mean and are visually highlighted with intense and sharp shading.

The final color scheme depicts feature co-occurrences to facilitate multidimensional analysis. We define feature co-occurrence as two features f_i and f_j that both have a value greater then zero within the same entity. We define a co-occurrence score $C_i = \sum_j p(f_j > 0|f_i > 0)^2$ which measures how often a feature f_i co-occurs with all other features ($p(\cdot)$ is the conditional probability). The result of this metric is normalized and encoded using color saturation to highlight the most co-occurred features within the cluster. The resulting color scheme serves as the ***"Co-occurrence Cue"***. As illustrated in Fig. 5(b), many patients are highlighted by dense colors in the bottom cluster because they have both the diabetes (colored in orange) and the kidney disorders (colored in blue). In contrast, most of the patient cells in the top cluster are encoded use light colors as most of them merely have the single disease, kidney disorder. Thus, reversing the color, we have the ***"Dominate Cue"*** that highlights the features that have not co-occurred with any other features as shown in Fig. 5(c).

Discussions. This design provides a number of key advantages. First, it provides intuitiveness and efficiency by leveraging several well established techniques such as a space filling layout [31] and the use of color saturation to depict data variance and diversity [27]. Second, DICON uses color and positions for cluster identification which are both high efficiency cues as described in [35]. As a result the icons clearly depict which clusters are similar to each other while still providing visual cues for more detailed analysis. Third, encoding components such as color and shape can be well scaled without significant loss of information as described, respectively, in [20] and [36]. This allows the design to remain effective for both large and small sized icons. In addition, the approach scales to work effectively with large numbers of icons as shown in Fig. 12. Fourth, the icons enable interactive manipulation which will be described in the next section. Finally, our design compresses multidimensional cluster information into relatively small cluster icons which can be easily embedded within other visualizations as illustrated in Fig. 1.

Yet there are also some constraints on our approach. In particular, the number of feature dimensions that can be visualized at any one time is limited because each must be represented by a unique user-distinguishable color. To alleviate this problem, feature selection can be used to identify the key features that should be included in a given visualization. Another challenge is that it can be hard for users to obtain precise feature values from our representation because comparisons of size and color across clusters can be difficult. In addition, the splitting of entities into parts may impede analysis for applications focusing on entities. However, we believe these limitations are a reasonable trade-off for the benefits of representing multidimensional cluster information using small, compact icons. We have also introduced some interactive features, such as highlights and tooltips, to target some of these concerns.

3.3 Interactions

As expressed in our design guidelines, a key requirement for DICON is that the visualization must allow users to interactively explore and refine the multidimensional clusters. DICON allows users to interactively perform the following cluster manipulation actions.

Merge. Users can merge cluster icons in two ways. First, users can drag and drop one icon onto another. Performing this action will merge the two corresponding clusters and create a single new icon to represent the newly created cluster. Second, users can merge two or more clusters by drawing a lasso around the corresponding icons. DICON will then merge all clusters selected by the lasso. DICON will animate the transition between states during the merging process to clearly illustrate the changes being made.

Split. Given a cluster icon, users can perform several types of split operations. To remove specific outlier entities, users can simply click on an entity and drag it out of the cluster. Releasing the mouse finalizes the split. As a result, DICON creates a new icon to represent the split entity and updates the existing cluster icon to reflect the split. Users can also perform algorithmic split actions via a pop-up context menu. After right clicking on a cluster, users can choose to perform either a binary split or an outlier split. The binary split operation breaks a cluster into two different sub-clusters. K-means is chosen as an example for the binary split in our implementation. Any other algorithms can also be used. The outlier split operation removes the one percent of entities that are farthest from the cluster center

Attribute Grouping. Users can explicitly request that data entities be re-grouped along various data dimensions. This feature allows users to consider non-feature entity attributes. For example, in an electronic medical record use case where diseases are features, patients could be grouped into clusters by non-feature attributes such as age, sex, or location. DICON can handle attribute grouping for categorical, numerical, and temporal attributes.

Filtering. DICON allows users to filter the set of feature categories used for cluster icon generation. By default, all data attributes selected as features are used to generate cluster icons. For multidimensional datasets with many such features, users can apply filters to reduce visual complexity and to focus in on a subset of the feature space.

Highlights. Because our encoding method spatially distributes an entity's feature cells across the cluster icon, DICON supports entity highlights. When a user's mouse hovers over a specific feature cell, all of the corresponding entity's feature cells are highlighted. A tooltip can also be shown to depict the entity's key attributes.

4 System Overview and Implementation

The DICON system's architecture, shown in Fig. 6, consists of three primary components. First, the *preprocessing module* extracts key features of the multidimensional dataset and optionally conducts an initial cluster analysis based on these features to transform raw data into a set of entity records in the form of $< id|cid|f_1, f_2, ..., f_n|a_1|a_2|...|a_n >$, where id is the record id, cid is the cluster id (optional), f_i is the ith feature and a_i is the ith non-feature attribute. The *visualization module* first generates entity or cluster icons via an icon layout algorithm. It then performs a global layout process to arrange the generated icons within the overall visualization canvas. The *user interaction* module supports user manipulations of the icons as described in Section 3.3. These operations feed back into the preprocessing and visualization modules to enable user-driven data exploration and cluster refinement.

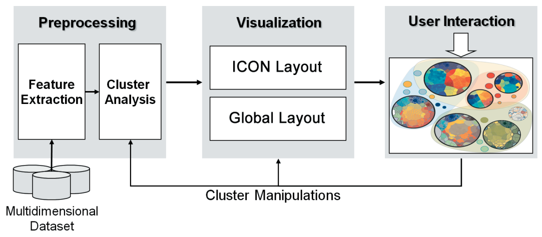


Fig. 6. The overview of the DICON visual analysis system.

In a typical visual analysis process by using the above DICON system, the users are given a set of data entities or some initial clustering results which are visualized as icons. Their correlations are disclosed by using different globle layouts and several visual cues such as the visual similarity of the icons. The users can iteratively merge, group, split the icons to generate or refine existing clusters until the high quality clusters are found. The whole process is rich in interactions which are not highly depend on the automatic clustering algorithms.

The entire system was implemented in Java and has been used with both traditional displays and on a Windows-based touch screen device. In the remainder of this section, we provide additional details about the icon and global layout algorithms. We also describe DICON's approach to handling animated transitions during cluster manipulation.

4.1 Icon Implementation

As mentioned previously, we adopt a treemap scheme to encode the entity-feature-cluster hierarchy in our icons. Treemap layouts have been heavily studied and many existing techniques can be leveraged. However, traditional layouts cannot satisfy all of our requirements. Therefore, we further develop a stabilized Voronoi-based layout.

4.1.1 Traditional Treemap Icons

The traditional treemap [31] is a well established technique used to visualize hierarchical structures. Many algorithms [3, 32, 37] have been proposed to lay out traditional treemaps according to different optimization criteria. These algorithms can be directly applied to generate layouts for our icons. The results are quite promising as illustrated in Fig. 5 and the process is very efficient. This makes them suitable for real time icon manipulation.

However, despite its computational efficiency, the traditional treemap icon also has some major limitations. First, the layout for rectangular treemaps may not be stable during the cluster refinement process. After users add or remove some entities to/from the cluster icon, the positions of cells may be shuffled and the layout may change dramatically. Second, there is no guarantee that similar icons will be generated for similar clusters. Traditional layout algorithms only do optimization within a single treemap. For multiple cluster icons, more constraints are needed to guarantee that the same features in different clusters are positioned in similar locations. Third, traditional treemaps produce rectangular icons which cannot be shaped to embed global cluster statistics as described in Section 3.2. These limitations make traditional treemaps inefficient for cluster comparison, and global statistic embedding.

4.1.2 Voronoi Icons

To overcome the limitations of traditional treemaps, we introduce a new Voronoi icon layout that satisfies all of DICON's design principles. Our algorithm embeds statistical information into the icon shapes and also introduces a stability factor while leveraging the centroidal Voronoi tessellation [7] and weighted Voronoi diagrams that are also used by Balzer et al. [2]. Before describing our algorithm in detail, we briefly review weighted and centroidal Voronoi tessellation.

Given a set $P = p_1, ..., p_n$ of sites (initial points), a Voronoi Tessellation is a subdivision of the space into n cells, one for each site in P, with the property that a point q lies in the cell corresponding to a site p_i iff $d(p_i, q) < d(p_j, q)$ for i distinct from j (d is a distance metric function). The segments in a Voronoi Tessellation correspond to all points in the plane equidistant to the two nearest sites. Weighted Voronoi diagrams use a weight w_i assigned to each point in p_i as part of the distance measure. The following additively weighted power distance measure can be used to create Voronoi tessellations with straight line boundaries:

$$d(p_i, q) = ||p_i - q||^2 - w_i^2 \tag{1}$$

Intuitively, one can consider the weighted Voronoi diagram as using using circles as sites instead of points where the circles' radii are a function of the corresponding weight w_i.

A Voronoi tessellation is called centroidal when all of the tessellation's sites are located at the center of mass for their respective regions. It can be viewed as an optimal partition corresponding to an optimal distribution of sites. A number of algorithms can be used to generate centroidal Voronoi tessellations, including Lloyd's algorithm and the K-means algorithm (see [7]). Recently, Balzer et al. introduced an optimization algorithm for weighted centroidal Voronoi tessellation to generate Voronoi treemaps [2]. We further extend Balzer's algorithm by introducing a stabilized centroid.

Statistic Embedding. To strengthen the visual design, our layout method first embeds statistical measures into an icon's bounding shape before applying the stabilized Voronoi layout. Generally speaking, two standard moments, the skewness ($\gamma = \mu_3/\sigma^3$) and the kurtosis ($\kappa = \mu_4/\sigma^4 - 3$), are embedded into cluster icons via an icon's overall shape that facilities analysis. In the layout, we use a ladder shape to simulate the underlying data distribution as illustrated in Fig. 3 and Fig. 4. The height and the top width of the ladder are defined, respectively, by the following functions:

$$k'(k) = 2(\alpha \cdot k + 1) \cdot r,\ k''(k) = -2r \cdot k + 2r \tag{2}$$

where k is the kurtosis, r is the original radii of the icon and $\alpha \in [0, 1]$ is a factor that controls the sharpness of the ladder. The width of the bottom end is automatically adjusted to keep the icon size proportional to the number of its containing entities. The cluster skewness is further used to adjust the position of the top vertex of the triangle to intuitively represent the data's asymmetry.

Stabilized Voronoi Icon Layout. Voronoi tessellation is computed within each shaped icon. We provide a stabilized Voronoi-based icon layout algorithm which maintains the stability of Voronoi regions when cluster changes occur and maintains a predefined order for sites within an icon. It places Voronoi regions next to each other according to their semantic similarities. In the layout process, we first arrange the feature types in an order that is followed in all cluster icons. For example, we can order the feature types according to their importance or follow a predefined order with certain semantics.

We maintain this site order during layout by carefully controlling the initial positions of their corresponding sites. Different strategies are used for different icon shapes. For example, for circular icons we initially layout the sites on a spiral line centered at and within the boundary circle. For rectangular icons, the sites are laid out line by line from left to right in order. A weighted CVT optimization is then performed which assigns a weight to each site based on the corresponding value and adjusts their positions and weights to obtain a proper tessellation.

The individual entity features are laid out inside the regions for each feature type by carefully controlling the positions and movements of their corresponding sites $S = s_1, ..., s_n$ during the CVT iteration. Intuitively, in each iteration, we move a site s_i towards to its region v_i's center of mass c_i while trying to balance two other constraints. First, we aim to position all similar sites as close as possible to each other while positioning dissimilar sites far apart (the screen distance $|X_i - X_j|$ is close to their semantic distance d_{ij} with a proper scaling factor). Second, as entities are added or removed from a cluster, we strive to maintain icon stability by minimizing any changes in location from a site's previous optimal position $pre(s_i)$. Formally, we capture

Algorithm 1: VoronoiIconLayout()

```
Data: $S(s_i,...s_n)$, $V$, $\varepsilon$, $pre(s_1)$, ..., $pre(s_n)$
Result: coordinates of each site $X(X_1,X_2,...,X_n)$
begin
    if $pre(S)$ is not empty then
        $X'_i = pre(s_i)$;
    else
        $X'_i =$ random locations within $V$;
    $stress' = 10000$ //give a very large value;
    while $ratio > \varepsilon$ do
        //the coordinate update based on stress majorization;
        $X_i = \frac{\sum_{i<j} \omega_{ij}(x_j + d_{ij}(x'_i - x'_j) inv(||X'_i - X'_j||))}{\sum_{i<j} w_{ij}}$;
        compute Voronoi tessellation $v_i$ according to $X_i$;
        compute $c_i$ according to $v_i$;
        $X_i = \mu_1 \cdot (X_i - c_i) + \mu_2 \cdot X_i + \mu_3 \cdot (X_i - pre(s_i))$;
        $w_i = w_i \cdot (1 + (desired_i - a_i)/desired_i)$;//adjust weight
        if $w_i < 1$ then
            $w_i = 1$
        $r = min\{|X_i - X_j|^2/(w_i + w_j)\}$;
        if $r < 1$ then
            $w_i = w_i \cdot r$
        $str_1 = \sum_i |X_i - c_i|^2$; $str_2 = \sum_i |X_i - pre(X_i)|^2$;
        $str_3 = \sum_{i<j}(|X_i - X_j| - d_{ij})^2$;
        $stress = \sum_k (\mu_k \cdot str_k)$;
        $ratio = (stress' - stress)/stress'$;
        $\mu_k = \mu_k \cdot (1 + (stress - str_k)/stress)$;
        normalize $\mu_k$;
        $X'_i = X_i$;
        $stress' = stress$;
```

these constraints in a layout model which tries to minimize follow the objective function:

$$\mu_1 \sum_i |X_i - c_i|^2 + \mu_2 \sum_{i<j} (\omega_{ij}|X_i - X_j| - d_{ij})^2 + \mu_3 \sum_i |X_i - pre(X_i)|^2 \tag{3}$$

where X_i is the coordinate of s_i. $c_i(c_x, c_y)$ is the mass center of the region v_i which can be computed by following equations:

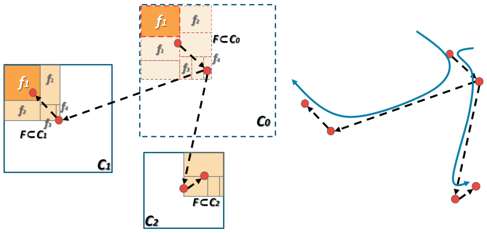
$$\begin{aligned} c_x &= \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ c_y &= \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{aligned} \tag{4}$$

where A is the area of v_i, and (x_i, y_i) is the ith vertex of polygon v_i.

In the layout objective function (3), d_{ij} is the semantic distance between two features f_i and f_j. It is defined by their corresponding feature vectors $f_i \in F_i$ and $f_j \in F_j$ as $(1 - cos(F_i, F_j))$ in our implementation. The weights μ_k ($\sum_k \mu_k = 1$ and $0 < \mu_k < 1$) balance the three parts of our layout model. They are changed adaptively during each CVT iteration using several heuristic strategies. Intuitively, we always keep μ_1 larger than the other two weights since we want the iteration to stop at a position where s_i at or close to its mass center c_i. Then we compute the errors of each part in the formula and increase the weight of the part that has the largest error and decrease the weight of the part that has smallest error. In this way, the part with largest error is the focus for minimization during the next iteration. Our proposed algorithm is specified in Algorithm 1. It leverages the stress majorization [13] technique to provide a local minimization of the model.

4.1.3 Discussions

The *VoronoiIconLayout*() algorithm satisfies all of the design principles outlined earlier in this paper. It has a time complexity of $O(n^2)$ for each iteration which is the same as Balzer's algorithm but worse than Lloyd's CVT algorithm ($O(nlog(n))$). Some acceleration techniques such as Sud et al. [33] and Gotz et al. [14] are available but it still remains a challenging task to layout Voronoi treemaps for large datasets in real time. To have the benefits of both real time interaction and high-quality layouts, DICON supports both rectangular treemap icons as well as the optimized Voronoi icons. The first are used to support real time exploratory interactions. Because of its efficiency, users can group any set of entities and clusters to generate new icons in real time. Switching to the Voronoi view helps users better understand and compare the clustering results.



(a) (b)

Fig. 7. Transition path bundling while splitting C0 into C1 and C2. The transition paths of feature cells are bundled based on their hierarchical cluster centers to reduce visual clutter during animation.

4.2 Global Layout

After the icon layout process completes, a global layout algorithm is used to position the icons and uncover their correlations. Various layout algorithms can be used for different purposes as illustrated in Fig. 1. For example, when icons are used to represent geographical clusters, they can be globally laid out based on their locations as shown in Fig. 1(a). DICON icons can also be used in conjunction with scatter plots to uncover the correlations among various dimensions as shown in Fig. 1(b). We can also apply our technique to a multi-relational graph visualization shown in Fig. 1(c) to reveal both patient communities and their relationships. The communities are generated according to patient similarities over multiple diagnoses and represented using DICON icons. The link colors and thicknesses encode different types of relations and their strengths, respectively. The layout of the icons in the graph can be computed using a force-directed model.

Beyond these applications which embed the icons within another visualization, DICON also provides a MDS-based projection to layout cluster icons based on their similarity. A fast overlap removal algorithm [8] is adopted to avoid overlapping icons. It eliminates overlaps while retaining each icon's original position as much as possible. Some improvements were made to these algorithms to facilitate interactive cluster manipulations. First, we minimize icon movement when clusters change by smoothing positional changes based on the icons' previous positions. Second, an incremental layout technique is used for split and merge commands. For example, when entities are split off from a cluster, only modified clusters (including any newly created clusters) are re-laid out in a sub-area followed by a global overlap removal. Thus, the positions for far away cluster icons are not affected.

4.3 Animated Transitions

When a cluster manipulation interaction such as attribute grouping or merging is applied, the icons may be reorganized and re-laid out to generate a new presentation of the data. In our system, this changing process is smoothly conveyed using a multi-step animated transition. First, feature cells for entities that change clusters are split from their original icon. Second, all of the feature cells are moved to their new location and their shapes are changed accordingly. Finally, the feature cells are repacked together under a new organization. During the second step, a naive approach to moving feature cells can create complex visual movements that are often confusing and hard to follow.

To overcome this problem, we use a transition path bundling technique. It aggregates the feature cells for each cluster into transition

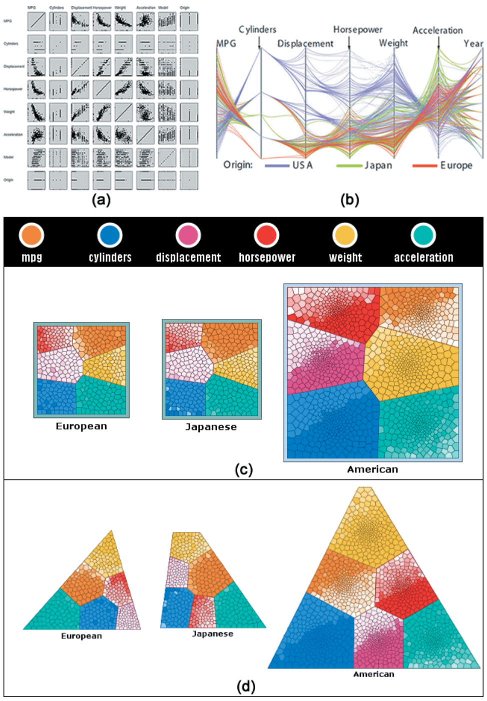


Fig. 8. Visualizations of the cars dataset using (a) scatter plot matrices with dimension reordering [10], (b) parallel coordinates with edge clustering [40], (c) DICON using square-shaped Voronoi-based icons, and (d) DICON using icon shapes to encode statistical cluster measures.

groups according to their movement trends. Each trend is defined using a polyline that describes the overall direction of movement. All the transition paths in a group are bundled together using a B-spline based on the control points of their associated trends. This spline guides the animation path. We compute the trends by using the innate hierarchy of our icon design. This algorithm is inspired by edge bundling [16].

To illustrate our algorithm, we consider a sample split interaction. Suppose a cluster C_0 is to be split into two smaller clusters C_1 and C_2 as shown in Fig. 7(a). Feature cell f_1, along with other feature cells f_i, will be split from C_0 and packed into a new icon for cluster C_1. Similarly, the remaining feature cells from C_0 will move to cluster C_2. The trend for feature f_1 is then defined as a polyline that connects the centers of f_1's feature type region in the C_0 icon, the C_0 icon, the C_1 icon, and its corresponding feature type region in C_1. The transition curves defined by the features' polyline trends are used to smoothly animate the feature cells as shown in Fig. 7(b).

5 Applications

This section presents examples of how DICON can be used to analyze multidimensional data. We discuss use cases for two different datasets.

5.1 Visualization of the Cars Dataset

We applied DICON to a cars dataset which has also been used to evaluate both parallel coordinates (PCP) and scatter plot matrices (SPM) (see Fig. 8(a) and (b)). The cars dataset contains 407 cars described by 7 different dimensions from which we selected 5 quantitative dimensions as features. The two remaining dimensions, year and origin, were used as additional attributes. The visualization results are illustrated in Fig. 8(c). Compared with PCP and SPM, our technique compresses the multidimensional information into a small number of compact cluster icons which require very little space for display. DICON immediately conveys the size of each cluster which is usually hidden in both PCP and SPM. For example, the number of cars produced by Americans is about 3 or 4 times larger than the number of cars produced by European or Japanese manufacturers. From Fig. 8(c), we can easily see that the European cars and Japanese cars have similar features. However the statistical distributions are different as shown in Fig. 8(d). American cars, when compared to European and Japanese models, usually have a larger weight (depicted in yellow), larger displacement (depicted in purple), more cylinders (depicted in blue), somewhat reduced acceleration (depict in cyan), much greater horsepower (depicted in red), and much fewer miles-per-gallon (depict in orange). We believe that doing a similar multidimensional comparison is relatively difficult with PCP and SPM techniques because line crossings and data overlaps are unavoidable in both of these alternative visualizations.

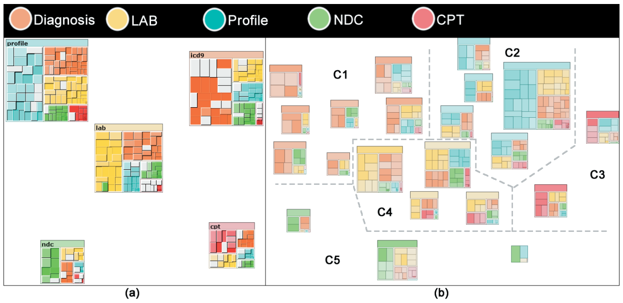


Fig. 9. Exploration of patient similarities with DICON: (a) Low quality in the initial five clusters generated by grouping the patient icons based on their visual similarity; (b) 21 user identified clusters in the co-occurrence view after refining the clusters, which can be roughly separated into five regions by their positions and visual similarities.

5.2 Case Study in the Healthcare Domain

We also applied DICON within the healthcare domain to visualize a dataset containing more than 10,000 patient records. The data includes claims, labs, pharmacy, and patient profile information. To augment this data, we applied a patient similarity algorithm to compute patient similarity scores across multiple dimensions (e.g., diagnoses, lab results, etc.). We also indexed the patient records to make the data searchable. We then invited two physicians to participate in a case study. Both physicians were required to complete two exploratory tasks that were motivated by real use cases proposed by domain experts. The tasks were as follows: (1) given a patient with a challenging disease, find the records of similar patients to serve as reference points during diagnosis; and (2) find the most prominent diseases over different age groups, geographic locations, and sex for a given a set of patient data.

5.2.1 Patient Similarity Task

In this task, we selected the similarity scores with respect to diagnosis, labs, profiles, NDC, and CPT as the features of each patient. Our users started with a query using a target patient and the system returned a large set of similar patients that were represented as icons and laid out based on the MDS projection. The users were required (a) to identify several patient clusters, (b) to refine their quality according to the data and cluster correlations, and (c) to answer how similar they are to the target query. At the beginning, the users grouped the patients according to their visual similarity and relative positions. This initial grouping generated clusters with low quality as shown in Fig. 9(a). The follow-up binary split and outlier split operations were used to quickly refine the clusters at a rough granularity. Then a series of more precise split and merge interactions moving small numbers of patients were conducted to fine tune the clusters according to both icons' screen potions and visual similarities. After several iterations of cluster adjustment, they finally found 21 meaningful and homogeneous clusters. When the users switched to the co-occurrence view (see Fig. 9(b)), they found that the patient clusters are roughly separated into 5 parts as annotated by the dashlines. The clusters in region C1 shared similar diagnoses with the target patient; the clusters in region C4 were similar to the target patient both in labs and diagnoses. This approach also helped to detect several non-medically-relevant clusters in C2. They were similarly to the target patient mainly by profiles.

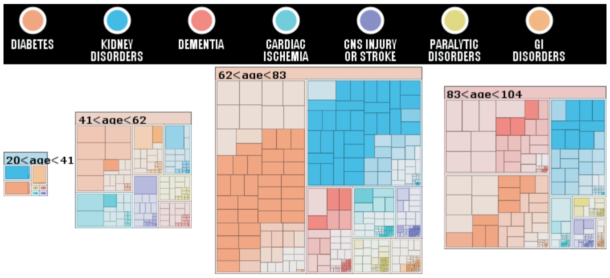


Fig. 10. DICON showing clusters of patients grouped by age. Here color depicts the co-occurrence cue to highlight the degree of disease co-occurrence within each age group.

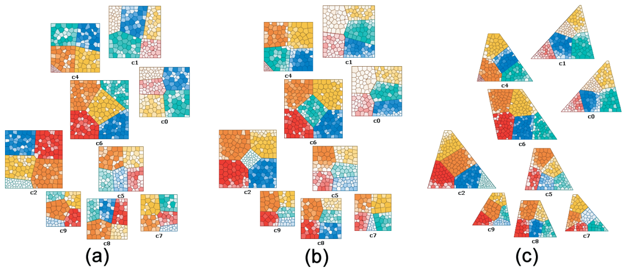


Fig. 11. The synthetic data used in user study task 1. Nine cluster icons are arranged using MDS projection. Icon types are evaluated using (a) random order packing, (b) ordered packing following the DICON design guidelines, and (c) ordered packing combined with icon shapes.

5.2.2 Study on Disease Distributions

In this task, we selected the severity of various diseases as the features to visualize with DICON. Our users found the requested disease distributions by using attribute grouping to cluster the patients on non-feature data attributes. For example, they grouped patients based on location and combined the icons with a map as illustrated in Fig.1(a). This clearly depicted the disease distributions for different states. When grouping patients by age, several disease co-occurrence over different age groups were found in the dataset. As illustrated in Fig. 10, "diabetes" was co-occurred with "kidney disorder" in patients aged from 62 to 83. In the oldest patient group, these two diseases were also co-related with "dementia". Similar groupings were also conducted by our users on other attributes such as race and sex.

6 User Study and Interviews

To evaluate DICON's ability to facilitate cluster interpretation and comparison for a large dataset, we conducted a controlled user study. The study included 30 participants (24 males, 6 females) all of whom were either graduate or undergraduate students. Participant ages ranged from 19 to 30. We also interviewed domain experts.

6.1 Study Setup

In order to conduct controlled experiments, we generated two synthetic datasets since the study contained too many parameters such as the skewness, the kurtosis and the number of clusters. The datasets were generated based on the multinomial distribution that made our synthetic datasets close to real datasets. The first dataset contained 9 clusters and 300 entities and was visualized using MDS layout as shown in Fig. 11. The second dataset contained 50 clusters and 1000 entities (see Fig. 12). With these datasets, we had users perform the following tasks: **(T1)** *identify the two most similar clusters in Fig. 11 and determine which features made them similar;* **(T2)** *identify groups of similar clusters from the large set of icons in Fig. 12.* We provided multiple answers as choices for both T1 and T2 and also allowed users to write down their own answers. Both of them evaluates the cluster comparison which is the most important feature of DICON. The first task evaluated DICON's support for distinguishing between clusters. The second task evaluated DICON's effectiveness for comparisons over a large set of clusters. We selected Voronoi icons for our study because it satisfied all our design guidelines. For both tasks, we compared user performance among three different Voronoi icon types: type 1 icons had feature types packed in a random order; type 2 icons obeyed all design guidelines including consistent ordering; type 3 icons were the same as type 2 but added shapes to encode data distributions.

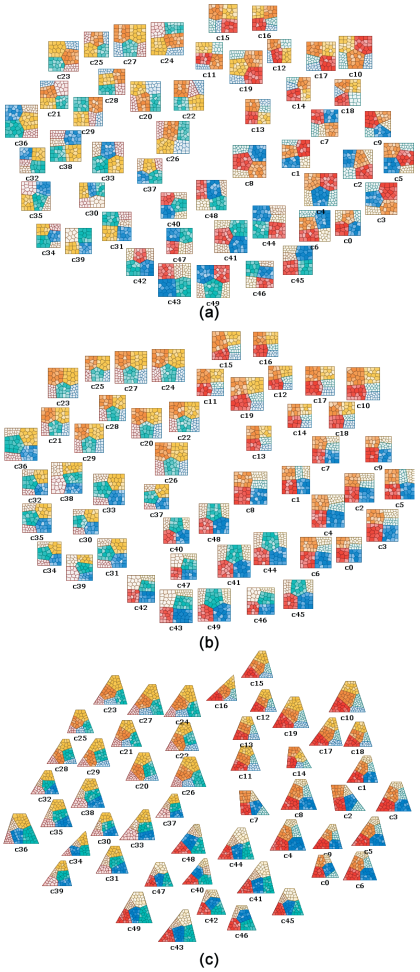
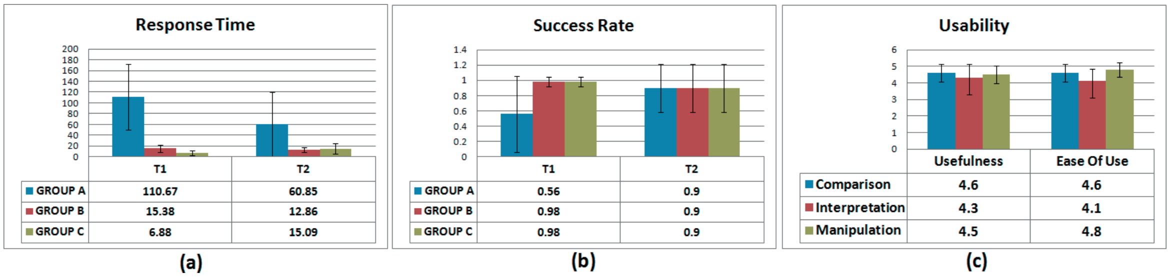


Fig. 12. The synthetic dataset with 50 clusters used in user study task 2. Icons generated by (a) random order packing, (b) DICON's ordered packing, and (c) ordered packing combined with icon shapes.

Before the study, users were given a 10 minute introduction to the icon design. Users were then allowed to try DICON for themselves to explore the patient dataset. Users were encouraged to group and regroup clusters, and to view different statistic measures for the results. This introduction was followed by a brief interview session to gather initial feedback. Users then performed the formal study tasks. We conducted a between subject study in which we separated the users into three groups of 10. Group A used type 1 icons, group B used type 2 icons, and group C used type 3 icons. All users were required to answer the questions as accurately as possible. Their response time and task success rate were recorded. Finally a questionnaire survey on system usability was conducted.

6.2 Study Results and Analysis

The study results are summarized in Fig. 13. The benefits of the proposed design principles were evident in both the task response time and the task success rate. A two-way repeated measures ANOVA analysis showed that when compared with the random type 1 icon, both type 2 and type 3 icons had significant improvements in response time



	T1	T2
GROUP A	110.67	60.85
GROUP B	15.38	12.86
GROUP C	6.88	15.09

	T1	T2
GROUP A	0.56	0.9
GROUP B	0.98	0.9
GROUP C	0.98	0.9

	Usefulness	Ease Of Use
Comparison	4.6	4.6
Interpretation	4.3	4.1
Manipulation	4.5	4.8

Fig. 13. User study results including (a) task response time, (b) task success rate, and (c) subjective feedback on “ease of use” and “usefulness”.

for both T1 (type 2: $0.0002 < .05$; type 3: $0.0001 < .05$) and T2 (type 2: $0.017 < .05$; type 3: $0.024 < .05$). The response time using type 2 and type 3 icons are similar in both tasks. In T1, the response time of type 3 icons was slightly better than that of type 2 icons, while for T2 the result was reversed. The task success rate of T1 also had significant improvements when compared with the baseline type 1 icons ($0.004 < .05$ of both type 2 and 3). There was no difference on task success rate between the type 2 icons and the type 3 icons on T1. We found the success rate of T2 was the same (90%) for all icon types.

The study results verified the efficiency of our design principles. They showed that a well organized multidimensional cluster icon visualization provided high efficiency on cluster comparison and interpretation. In addition, they showed no negative impact of using shape to encode additional cluster statistics in type 3 icons. Moreover, even when visualizing a large number of clusters, the type 2 and 3 icons remained highly efficient. It took an average of only 12.86 seconds to compare 50 multidimensional clusters with a task success rate of 90%. This was a surprising finding which would be difficult to achieve using PCP techniques according to Holten et al.’s study result [17]: when the number of clusters are over five (far fewer than the 50 in our study), the ability to identify clusters with PCP is dramatically decreased.

Furthermore, we found that the guidelines for generating similar icons for similar clusters plays a very important role in making a precise comparison as required in T1. Several of the users were confused when using type 1 icons to make the comparison. After the study, many of them said that they gave the answers mainly by guessing. The correctness is thus far worse than the results observed when using icon types 2 or 3.

In addition to the quantitative results, we collected qualitative user feedback on the key features of the DICON system (visual comparison, interpretation and interactive cluster manipulation) by considering “ease of use” and “usefulness”. As illustrated in Fig. 13(c), all of the features had a high average score over four (five is the highest score). The cluster comparison capability was rated as the most useful feature and interactive cluster manipulation was considered the easiest feature by the study participants. Interpretation was rated slightly lower when compared with other two features. We believe that this is because it is a feature based on comparison and requires additional effort to understand the meaning of the feature categories and clusters.

6.3 Interviews with Domain Experts

Using the patient dataset described previously, we conducted extended one-on-one interviews with two medical doctors with very strong domain expertise. The first doctor is a former emergency physician with over 30 years of hospital-based experience. He has published multiple articles and book chapters on both clinical and management subjects. The second doctor is a health care and biotechnology executive who has, in addition to clinical experience, more than 30 years of expertise in sophisticated managed care organizations, strategic planning, and operations management.

Both physicians were intrigued by the interactive visualization that DICON provides for examining and manipulating similar patient cohorts. The physicians were attracted to the interactivity and felt that the iconic representations provided significant value. At first, one physician remarked that the icon was more “complex” than he was used to (e.g., bar charts). However, after brief demonstrations of the tool the representation became clear.

Referring to DICON’s refinement capabilities, one physician said “it provides an interesting way to define cohorts.” He was especially interested in the drag-and-drop nature of the technique which he felt provided a “very intuitive interface” for manipulating sets. He very much liked the icon design which provided a concrete object for him to analyze and manipulate. When referring to the interactive refinement of cohorts, one physician stated that “as a medical director, this is exactly what I would want to do.” The icon design let him “do it rapidly [via] drag and drop” instead of “giving it to a programmer” to generate a new report. One physician commented that the ability to overlay statistical measures of quality onto the visualization was very useful. When asked if that helped him gauge the quality of clusters, he responded “absolutely.”

In addition to commenting on DICON’s current functionality, the participants also made a number of suggestions for future improvements. For example, the physician wanted the ability to perform attribute grouping for combinations of dimensions (e.g., both sex and age). This is a feature that we hope to introduce in future revisions of the tool. A more complicated request made by one physician was the ability to drag the icon for a cohort from our tool onto icons for other system functionality. His suggestion was to use this approach to issue requests for additional analytics to be applied to a given group of patients. The user’s request for this feature shows that the tangible icons we designed for representing cohorts form a very powerful representation in the minds of our users. The icon itself becomes the object that the user wishes to operate on. We believe this is a very powerful design approach and we are exploring ways to adopt it.

7 Conclusion and Future Work

In this paper, we presented DICON, an interactive icon-based multidimensional cluster visualization. It provides a novel approach to visual cluster comparison, interpretation and adjustment. Compared with traditional visualization techniques, DICON encodes additional derived statistical information that provides visual cues to facilitate cluster evaluation and adjustment. DICON also scales well to effectively support large numbers of clusters. DICON’s design follows several predefined guidelines and leveraging several well established designs. Strong information scent and visual cues for cluster quality evaluation, interactive adjustment and exploration are provided by this design. Interactions are further supported for cluster manipulation. Finally, new layout algorithms as well as animated transition techniques were introduced to satisfy DICON’s design requirements. Our evaluation, including a case study, user study and feedback from domain experts, demonstrates the effectiveness of DICON. Future work includes performing additional user studies to evaluate other aspects of the design and proposing new layout algorithms and features.

Acknowledgments

We thank all the user study participants and doctors for their contributions to the system evaluation, Dr. Tim Dwyer for his help on the node overlap removing and the anonymous reviewers for their valuable comments. This work was supported in part by grant HK RGC GRF 619309 and an IBM Faculty Award.

REFERENCES

[1] M. Ankerst, S. Berchtold, and D. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *IEEE Symposium on Information Visualization*, pages 52–60, 1998.

[2] M. Balzer and O. Deussen. Voronoi treemaps. In *IEEE Symposium on Information Visualization*, pages 49–56, 2005.

[3] M. Bruls, K. Huizing, and J. Van Wijk. Squarified treemaps. In *Proceedings of the joint Eurographics and IEEE Symposium on Visualization*, pages 33–42, 2000.

[4] D. Carr, R. Littlefield, W. Nicholson, and J. Littlefield. Scatterplot matrix techniques for large N. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[5] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[6] S. Climer and W. Zhang. Rearrangement clustering: pitfalls, remedies, and applications. *Journal of Machine Learning Research*, 7:919–943, 2006.

[7] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

[8] T. Dwyer, K. Marriott, and P. Stuckey. Fast node overlap removal. In *Graph Drawing*, pages 153–164, 2006.

[9] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 14863–14868, 1998.

[10] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1141–1148, 2008.

[11] M. Friendly. Corrgrams. *The American Statistician*, 56(4):316–324, 2002.

[12] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings of IEEE Conference on Visualization*, pages 43–508, 1999.

[13] E. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *Graph Drawing*, pages 239–250, 2005.

[14] D. Gotz. Dynamic voronoi treemaps: a visualization technique for time-varying hierarchical data. Technical Report RC25132, IBM, 2011.

[15] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1302–1309, 2007.

[16] D. Holten. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[17] D. Holten and J. Van Wijk. Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

[18] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE conference on Visualization*, pages 361–378, 1990.

[19] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009.

[20] D. Keim. Designing pixel-oriented visualization techniques: theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2000.

[21] D. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[22] D. Keim and H. Kriegel. VisDB: database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49, 2002.

[23] D. Keim and H. Kriegel. Visualization techniques for mining large databases: a comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):923–938, 2002.

[24] E. Keogh, L. Wei, X. Xi, S. Lonardi, J. Shieh, and S. Sirowy. Intelligent icons: integrating lite-weight data mining and visualization into gui operating systems. In *Proceedings of the International Conference on Data Mining*, pages 912–916, 2006.

[25] L. Nováková and O. Štepánková. Multidimensional clusters in RadViz. In *Proceedings of WSEAS International Conference on Simulation, Modelling and Optimization*, pages 470–475, 2006.

[26] M. Novotny. Visually effective information visualization of large data. In *Proceedings of the Central European Seminar on Computer Graphics*, pages 41–48, 2004.

[27] T. Pham, R. Hess, C. Ju, E. Zhang, and R. Metoyer. Visualization of diversity in large multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1053–1062, 2010.

[28] R. Pickett and G. Grinstein. Iconographic displays for visualizing multidimensional data. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 514–519, 1988.

[29] F. Post, T. van Walsum, F. Post, and D. Silver. Iconic techniques for feature visualization. In *Proceedings of IEEE Conference on Visualization*, pages 288–295, 1995.

[30] J. Seo and B. Shneiderman. Interactively exploring hierarchical clustering results. *IEEE Computer*, 35:80–86, 2002.

[31] B. Shneiderman. Tree visualization with treemaps: 2d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[32] B. Shneiderman and M. Wattenberg. Ordered treemap layouts. In *IEEE Symposium on Information Visualization*, pages 73–80, 2001.

[33] A. Sud, D. Fisher, and H. Lee. Fast dynamic voronoi treemaps. In *IEEE International Symposium on Voronoi Diagrams in Science and Engineering*, pages 85–94, 2010.

[34] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66, 2009.

[35] E. Tufte and G. Howard. *The visual display of quantitative information*. 1983.

[36] C. Ware. *Information visualization: perception for design*. 2004.

[37] J. Wood and J. Dykes. Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, 2008.

[38] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, and W. Ribarsky. Value and relation display: interactive visual exploration of large data sets with hundreds of dimensions. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):494–507, 2007.

[39] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1001–1008, 2009.

[40] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.