OPTIMAL DISPLAY ADAPTATION OF ICONIC DOCUMENT VISUALIZATIONS VIA BFOS-STYLE TREE PRUNING

Kathrin Berkner and Michael J. Gormish

Ricoh Innovations, Inc.
2882 Sand Hill Rd, Suite 115, Menlo Park, CA 95025, USA

ABSTRACT

This paper introduces a new visual representation of a document or group of documents, a Dynamic Document Icon, or Dydocon. Its representation is symbolic like an icon, but changes depending on document content. A Dydocon can be used for multiple documents and thus is useful for presentation of clustered search results. Because a large number of clusters may be returned by a search it is desirable to optimally select clusters given available display space. We cluster documents into a visual taxonomy described by a tree of Dydocons, define tree functionals for icon size and distortion of a cluster, and perform display-size adaptation through optimization using a BFOS-style algorithm.

Index Terms— Document image processing, tree searching, rate distortion theory, visual languages.

1. INTRODUCTION

1.1. Visual Document Representations

The two most common visual representations for documents are thumbnails and icons. Icons are symbolic representations of either the filetype or the application used to open the document. Because icons are symbolic and typically designed for each file type or application by hand they can be fairly small. The information content of the icon, however, is also small, often equivalent to the file name extension, e.g. ".pdf".

Thumbnails are typcially a scaled version of a complete page. They are often intended to assist the user in visual differentiation of documents. This goal is only achieved, however, if the user-relevant distinct information between two documents is still distinguishable after scaling. Another drawback of thumbnails is that they can only represent one page at a time, not a group of pages or documents. Existing solutions to the problem of presenting multiple pages or multiple documents choose a "representative" thumbnail from a set of predetermined thumbnails.

In [1] iconic representations of various document layouts are introduced to query a document image database and present retrieval results. This paper introduces a new iconic representation that can be used for a single document or a collection of

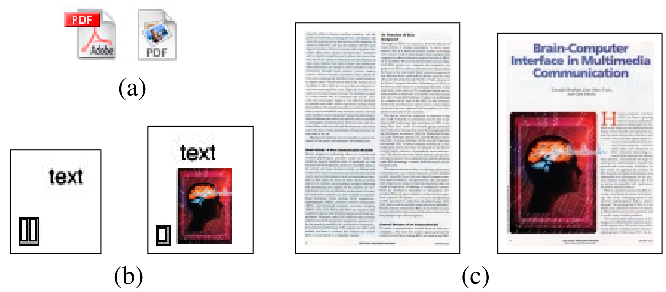

Fig. 1. (a) Icons showing filetype. (b) Dydocons and (c) thumbnails showing layout and content information.

documents. This representation, called a Dynamic Document Icon, *Dydocon*, is dependent on the content of the documents in the group. Dydocons are made up of iconic elements that can be relatively small and still be distinguished by setting a lower bound on their geometric size. A Dydocon can convey information about both layout and contents of documents. Because Dydocons can contain a variable amount of information they are useful in the creation of a visual taxonomy of a group of documents. The size of a Dydocon is a function of the amount of information (set of elements) being presented. This property enables adaptation of the iconic visualization to a target display size. Content-dependent icon size and adaptation to target displays is not considered in [1].

Examples of icons, thumbnails, and Dydocons are provided in Fig. 1. Even though a thumbnail takes roughly three times the screen area of a Dydocon it contains a similar amount of non-semantic information, e.g. the thumbnail on the left shows a two column document with no images, and the thumbnail on the right is one column of text and contains an image. The Dydocon on the left represents one or more documents with two columns text only, while the Dydocon on the right is for one column documents with text and images.

1.2. Clustering and Browsing

Since Dydocons can be used to represent a group of documents they are particularly useful for browsing a large document collection. In particular, we have in mind an application that involves browsing of a document collection on a small display. Evaluation of such a browsing scenario would require a thoughtful task definition, interface design, and user studies. Instead of addressing each of these non image procesing issues poorly, we define a framework, including properties of visual image representations, and algorithms, which is optimal under the set conditions. This framework allows future HCI and user studies to be done efficiently.

Any given set of documents is clustered hierarchically based on layout and content. Each level of the hierarchy provides additional information about the set of documents, and thus may have an additional element in the Dydocon. A Dydocon is created for each cluster in the tree. An example of a Dydocon tree is shown in Fig. 2. Given a constrained target display area $\mathcal{A}$, it is generally impossible to show Dydocons for all documents returned by a search. Therefore, some subtree is chosen with the same root as the entire tree. By defining an icon creation method and hierarchical structuring in a way that monotonicity of certain tree functionals is assured we are able to apply the generalized BFOS algorithm [2] in a tree pruning step that extracts subtrees with optimal display-area vs. distortion values.

2. DYNAMIC DOCUMENT ICONS

Dydocons are created to visualize document layout features of groups of documents. On the one hand the visualizations should be clearly perceived by the user, i.e. details in the visualizations such as edges or color contrast should be visible. On the other hand, visualization in form of one or more Dydocons should fit into the targeted constrained display area, e.g. a window of a specific size on a desktop or PDA.

Efficient use of display space is commonly achieved by showing small thumbnails of documents. A thumbnail, however, only visualizes an individual document, not a group of different documents. If many documents must be displayed thumbnails often become too small to be useful.

Using iconic elements it is possible to visualize layout features such as columns, title, picture regions etc. The size of such icons may be smaller than a typical thumbnail size.

Layout features of a group of documents are used to create a Dydocon as a representative visualization of the group. A document page is composed of structural units, called page zones. Extraction and labeling of those zones has long been researched in the Document Analysis field [3]. We associate a page zone with an iconic element. For example, a text paragraph, a figure unit, or the entire page is associated with a rectangle of certain size and filled with a certain texture. The texture may depend on the zone content: white or gray for text zones, an iconified graphic for graphic zones, an iconified photo for photo zones.

To insure visibility of features on a particular display, each iconic element carries a *minimal size attribute* s^*. For example, a column element has $\mathsf{s}^* = 10$, a white-space element $\mathsf{s}^* = 1$, and a figure element $\mathsf{s}^* = 7$ pixels. A final Dydocon is composed out of a set of iconic elements, such that proportions between iconic elements in the icon should be the same as the proportions between corresponding page zones. Thus the dimensions of an icon are content-dependent (see Fig. 1).

3. HIERARCHICAL DOCUMENT CLUSTERING WITH ICONIC LABELS

In this paper, we consider the visualization of layout features of a set of documents given constrains on the target display area $\mathcal{A}$. Those constraints allow only for a selected subset of icons to be fitted into $\mathcal{A}$. Selected icons should show representative features of the document collection. Therefore the question is how to select icons for final display.

Our approach to this problem is to organize the layout features of a document set in a tree structure by performing hierarchical clustering of the feature space. The resulting tree tree $\mathcal{T}$ reflects a visual taxonomy of the document collection. Individual tree nodes $\mathsf{n} \in \mathcal{T}$ reflect groups of documents containing similar specific layout features. Those layout features are visualized by icons as described in Section 2. That means that each node n has an iconic visualization $\mathsf{V}(\mathsf{n})$ associated with it. Such a visualization is a label for the document cluster represented by the associated node. The set of all leaf nodes of $\mathcal{T}$ is denoted by $\mathcal{L}(\mathcal{T})$. A parent node of a node n is denoted by $\mathsf{p}(\mathsf{n})$ and a child node of n by $\mathsf{c}(\mathsf{n})$. The set of all children of a node n is denoted by $\mathsf{C}(\mathsf{n})$.

The visualization $\mathsf{V}(\mathsf{n})$ of a node n is an image covering some spatial region. That region can be interpreted as a resource for $\mathsf{V}(\mathsf{n})$. With this interpretation we define a *resource tree functional* R on a subtree S of $\mathcal{T}$ as the smallest spatial area that can contain the visualizations $\mathsf{V}(\mathsf{n})$ for all $\mathsf{n} \in \mathcal{L}(\mathcal{T})$. In the following we constrain the fitting problem of the visualizations into $\mathcal{A}$ to a one-dimensional problem: The width of a region containing all leaf node visualizations in a horizontally ordered line-up should fit into the width of $\mathcal{A}$. For this case we define R as

$$\mathsf{R}(\mathcal{S}) = \sum_{\in \mathcal{L}(\mathcal{S})} \mathsf{w}(\mathsf{V}(\mathsf{n})) \tag{1}$$

where $\mathsf{w}(\mathsf{V}(\mathsf{n}))$ is the width of the visualization of the node n. With this definition R is a linear tree functional [2]. Through construction of the icons by successively adding iconic elements of guaranteed minimal size when expanding a parent to children nodes, we obtain $\mathsf{w}(\mathsf{V}(\mathsf{n})) \geq \mathsf{w}(\mathsf{V}(\mathsf{p}(\mathsf{n})))$. Together with the linearity from Eq. (1) it follows that R is monotonically increasing, i.e. if $\mathcal{S}_1$ is a subtree of $\mathcal{S}_2$ with both trees having the same root node (denoted by $\mathcal{S}_1 \preceq \mathcal{S}_1$) it follows that $\mathsf{R}(\mathcal{S}_1) \leq \mathsf{R}(\mathcal{S}_2)$.

Tree growing is performed by divisive, monothetic clustering of layout features. Use of a single feature (monothetic) ensures that simple Dydocons exist which express differences between clusters (see also [4]). Many different methods for splitting a set of feature vectors into a collection of subsets have been used in the literature, such as k-means, graph partitioning, etc. [5]. In this paper the requirement imposed on the splitting criterion is that the clustering split decision should have a visual representation in form of Dydocons.

4. DISPLAY ADAPTATION OF THE ICON TREE

4.1. Stopping criteria for growing of the icon tree

When growing a tree, stopping or pruning of the tree is often applied to avoid overfitting of data or to satisfy some constraints. In general, pruning is considered to be more effective than stopping [6]. In this paper, the additional constraint given by $\mathcal{A}$ enables an interesting interplay of pruning and stopping of the Dydocon tree.

In each tree-growing (cluster dividing) step a former leaf node is expanded into children nodes that become leaf nodes of the current tree. Leaf nodes that are candidates for turning into parent nodes in the next growing step are denoted as *open leaf nodes*. Leaf nodes that can not be turned into parent nodes are called *closed leaf nodes*. Using the monotonicity of the resource functional R we can define the following stopping criterion for the growing process of the Dydocon tree. Given an upper resource bound R^* (e.g. the width of $\mathcal{A}$), if $\sum_{(\ )\in(\ )} \mathsf{R}(\mathsf{V}(\mathsf{c}(\mathsf{n}))) \quad \mathsf{R}^*$ for an open leaf node n, that node becomes a closed leaf node, i.e. stopping is performed.

4.2. Display-adaptive tree pruning with generalized BFOS

In a data compression application, optimal quantization with respect to rate-distortion performance is achieved using the generalized BFOS algorithm [2]. Two tree functionals are defined on the quantizer tree T: an affine monotonically increasing bit rate functional u_1 and an affine monotonically decreasing distortion functional u_2. Optimal tree pruning for a given bit rate is performed by tracing the lower boundary of the convex hull of the set $\{\mathsf{u}_1(\mathsf{S}), \mathsf{u}_2(\mathsf{S}), \mathsf{S} \preceq \mathsf{T}\}$. Bit rate can be interpreted as a resource for the signal representation.

The functional R from Eq. (1) satisfies the requirements for u_1 in [2] being affine and monotonically increasing. If, in addition, we can construct an affine monotonically decreasing distortion functional for the icon tree we can adapt the BFOS method to the Dydocon tree pruning application in this paper.

We define a *distortion tree functional* D of a subtree S of $\mathcal{T}$ as the sum of the dissimilarities of all the clusters associated with leaf nodes of S

$$\mathsf{D}(\mathcal{S}) = \sum_{\in \mathcal{L}(\mathcal{S})} \mathsf{d}(\mathsf{n}), \qquad (2)$$

where $\mathsf{d}(\mathsf{n})$ is the dissimilarity of the cluster of n.

In case clustering is performed via the k-means method and dissimilarity is measured as $\mathsf{d}\ (\mathsf{n}) = \sum_{\in} (\mathsf{x} - \mathsf{x}^*(\mathsf{n}))^2$, where $\mathsf{x}^*(\mathsf{n})$ is the centroid of the cluster of the node n, it is shown in [7] that the monotonicity

$$\mathsf{d}(\mathsf{n}) \geq \sum_{\in\ (\ )} \mathsf{d}(\mathsf{m}) \qquad (3)$$

holds. From Eq. (2) it follows for the choice of $\mathsf{d} = \mathsf{d}$ that D is linear and monotonically increasing, i.e. $\mathsf{D}(\mathcal{S}_1) \geq \mathsf{D}(\mathcal{S}_2)$ for tree functionals $\mathcal{S}_1$ and $\mathcal{S}_2$ with $\mathcal{S}_1 \preceq \mathcal{S}_2$, and therefore fits the BFOS framework.

5. EXPERIMENTAL RESULTS

We start with a test collection of 13 documents. The collection contains article pages formated in 1-, 2-, or 3-column format. Some documents consist of text zones only, others of text and picture zones. We define the feature space to contain the *number of columns* feature g and the *content* feature f reflecting the distribution of area coverage of text f and picture zones f relative to the page area. Document analysis information describing those features is assumed to be available.

Clustering is performed using the k-means algorithm with $\mathsf{k} = 2$. At each tree node, clustering is performed in each feature g and f separately. The output of the two clustering steps is compared and the one with smallest cluster split cost is chosen. The cluster split cost is defined as $\cdot\ (\mathsf{n}_1, \mathsf{n}_2)[1/\mathsf{d}_2(\mathsf{n}_1)) + 1/\mathsf{d}_2(\mathsf{n}_2))]$, where $(\mathsf{n}_1, \mathsf{n}_2) = \sum_{\in\ _1\ \in\ _2} (\mathsf{x} - \mathsf{y})^2$ for children nodes $\mathsf{n}_1, \mathsf{n}_2$ of n. To avoid bias introduced by different units of g- and f-features the weight is included and set to $= 0.5$ for g-features and to $= 1$ for f-features.

Given a clustering result, a Dydocon needs to be created for the cluster. The set of iconic elements used for icon composition consists of rectangular elements for page (gray) and column units (white) each with $\mathsf{s}^* = 10$ pixels, as well as white space between columns with $\mathsf{s}^* = 2$ pixels. Composited icons show a 1-, 2-, or 3-column layout. The width of the icon increases with the number of columns. Thus the column layout feature of node n is transformed into a geometric property: the width of the visualization $\mathsf{V}\ (\mathsf{n})$.

For visualization of the content feature $(\mathsf{f}\ , \mathsf{f}\ )$ first the cluster centroid $(\mathsf{c}\ , \mathsf{c}\ )$, as determined by the k-means algorithm, is selected. Depending on those centroids content labels are added to the right of $\mathsf{V}(\mathsf{g}(\mathsf{n}))$. If $\mathsf{c} \quad 0$, i.e. the cluster contains text zones, a label containing the word "text" is added. This label has a minimal size $\mathsf{s}^* = 30 \times 20$ pixels. If $\mathsf{c} \quad 0$, i.e. the cluster contains picture zones, the largest picture zone in the cluster is cropped, scaled and pasted below the text label. For a picture element $\mathsf{s}^* = 35 \times 35$ pixels. Depending on the proportions of text or picture content between two children the associated labels are scaled accordingly to visualize the difference in content distribution between the two icons. The resulting content label visualization of a node n is denoted by $\mathsf{V}\ (\mathsf{n})$. The final visualization consist of the composition of V and V, denoted by V, where composition is performed via horizontal concatination.

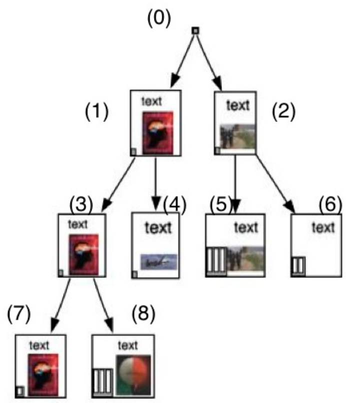

Fig. 2. Dydocon tree for the document collection.

In the tree example in Fig. 2 proportional scaling of the text label is performed, e.g., for the two children of the root node, reflecting more text content in the right cluster than in the left cluster. Proportional scaling of images is performed, e.g., for the bottom most leaf nodes: the width of the picture at node (8) is larger than the width in the picture at node (7).

Due to the construction of the icon part containing the column visualizations the width of the icon is monotonically increasing with the distance of a node to the root node. Therefore, setting $V = V$ in the definition of R in Eq. (1) assures monotonic increase of R. Scaling of the content labels is defined in a way such that the monotonically increasing behavior of R also holds for $V = V$. Since the two parts column layout and labels are concatinated horizontally, setting $V = V$ consequently assures monotonic increase of R.

Dissimilarity d at a node is defined as the sum of individual the k-means dissimilarities of g- and f - features in the node cluster, i.e.

$$d(n) = \sum_{\in}(x| \ - c^*)^2 + \sum_{\in}(x| \ - c^*)^2 \qquad (4)$$

where $x \in n$ is a 3-dimensional vector representing features g, f and f of a document in the node n. From the observations in Section 4.2 it follows that the distortion functional D from Eq. (2) using the distortion measure d from Eq. (4) is monotonically decreasing. Icon width and dissimilarity value for the nodes in the example in Fig. 2 are displayed in Table 1.

Table 1. Icon width and dissimilarity values for the nodes in Fig. 2.

icon	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$w(V(n))$	5	47	40	45	45	55	49	48	60
$d(n)$	11.14	1.00	0.37	0.45	0.04	0.19	0.01	0.00	0.00

Given the width W of the constrained area $\mathcal{A}$, the goal is now to determine which of the icons in the tree should be displayed in the available display area. The collection of leaf node icons provides the most detailed division of layout features in the collection. However, the combined width of all the leaf node icons may larger than W. Therefore, pruning of the tree is applied using the generalized BFOS algorithm to select the subtree with best R/D value. The leaf node icons of this optimal subtree form the icons for the final display. Results in form of leaf node icons of this optimal pruning are shown in Fig. 3 for two display widths $W = 140$ and $W = 200$.

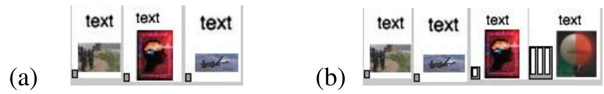

Fig. 3. Pruning results: (a) Nodes {(2)(3)(4)} adapting to $W = 140$ and (b) nodes {(7)(8)(4)(2)} adapting to $W = 200$.

6. CONCLUSIONS

We have introduced Dydocons, a new visualization for documents and groups of documents, a clustering technique that allows a hierarchy of Dydocons to be used for a visual taxonomy, and defined tree functionals allowing optimal selection of a subtree for display purposes. This is a step in the general problem of maximizing perceived information subject to constrains on physical display characteristics.

7. REFERENCES

[1] Cullen, J. et al., "Image database browsing and query using texture analysis," in *ICDAR*, 1997, pp. 718–721.

[2] Chou, P.A., et al., "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, 1989.

[3] Aiello, M. et al., "Document understanding for a broad class of documents," *IJDAR*, vol. 5, pp. 1–16, 2002.

[4] Kummamuru, K. et al., "A hierarchical monothetic document clustering algorithm for summarization and browsing search results," in *Proc. WWW2004, New York, NY*, 2004, pp. 658–665.

[5] Zhao, Y., Karypis, G., "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Machine Learning*, vol. 55, pp. 311–331, 2004.

[6] Li, X.-B. et al., "A dynamic programming based pruning method for decision trees," *INFORMS Journal of Computing*, vol. 13, pp. 332–344, 2001.

[7] Ding, C.H.Q., X. He, "Cluster merging and splitting in hierarchical clustering algorithms," in *Proc. Intern. Conf. on Data Mining*, 2002, pp. 139–146.