How small should a document thumbnail be?

Kathrin Berkner*
Ricoh Innovations, Inc., 2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025

ABSTRACT

Viewing document images on small devices is a challenge. When showing a region of a page for reading on a small display a page overview is generally lost. If a page overview is desired, typically a low resolution version of the image fitting a small array of pixels - a thumbnail - is provided. Whereas the readability of text in thumbnails is often lost, document layout information may or may not be preserved. Preserving document layout information in thumbnails is the goal of this paper. We derive models for controlling the preservation of document layout information in thumbnails by determining the size of a thumbnail depending on the layout content of the document. The downsampling factor for a document image will depend on its layout information, such that layout units will be visually separate after scaling. The link between scaling factors and document layout information is created through novel models, White Space Graphs and White Space Trees. These models enable control over enhancement and suppression of document layout structures during scaling. Minimal scaling factors can be derived that assure visual separability of a controlled set of layout units after scaling. Those scaling factors depends on the document content as well as user and display specifications.

Keywords: Thumbnail, document layout, Voronoi diagram, White Space Graph.

1. INTRODUCTION

Documents in electronic form are part of many office workflows. They are created, stored, distributed, edited, viewed, browsed, or printed. Whereas print quality of documents has reached a satisfying level with 600dpi printers, viewing of documents on electronic displays still remains a challenge. The limited number of pixels on electronic displays creates visualization challenges: How does one trade off display of readable text in a document page versus an overview of the entire page? And how should navigation through a multi-page document or through a document collection be visualized?

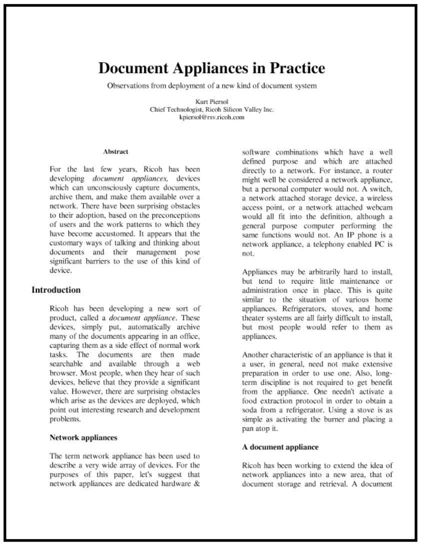
A page overview is typically provided in form of a thumbnail representation. A thumbnail is a small-size visualization of an image, where the smaller size is achieved through traditional downsampling performed on the original image intensity data. For document images, a thumbnail view of a document is not always enough to help the viewer fulfill the viewing task. For a search task, often text information is the main information used.

Thumbnails are displayed as a source for optional information with the intent that "something" in that thumbnail image may be useful to the user. What exactly the "something" is, is unclear. Whereas usefulness of textual retrieval results has been widely researched, the usefulness of thumbnail images is much less understood. Informal user surveys revealed that people often turned the thumbnail view off because they could not recognize anything in thumbnails and that the thumbnails took up too much space. These concerns lead to two central questions when thinking about creating thumbnails:

1) What document information should the thumbnail contain?
2) What size should the thumbnail be?

The first question regarding the kind of information represented in a thumbnail is addressed in Enhanced Thumbnails[16] and SmartNails[5]. Enhanced Thumbnails separate text information from the document and display keywords pasted on top of a low contrast version of a scaled document image. SmartNails focus on displaying readable

* berkner@rii.ricoh.com



Document Appliances in Practice

Observations from deployment of a new kind of document system

Kurt Piersol
Chief Technologist, Ricoh Silicon Valley Inc.
kpiersol@rsv.ricoh.com

Abstract

For the last few years, Ricoh has been developing *document appliances,* devices which can unconsciously capture documents, archive them, and make them available over a network. There have been surprising obstacles to their adoption, based on the preconceptions of users and the work patterns to which they have become accustomed. It appears that the customary ways of talking and thinking about documents and their management pose significant barriers to the use of this kind of device.

Introduction

Ricoh has been developing a new sort of product, called a *document appliance*. These devices, simply put, automatically archive many of the documents appearing in an office, capturing them as a side effect of normal work tasks. The documents are then made searchable and available through a web browser. Most people, when they hear of such devices, believe that they provide a significant value. However, there are surprising obstacles which arise as the devices are deployed, which point out interesting research and development problems.

Network appliances

The term network appliance has been used to describe a very wide array of devices. For the purposes of this paper, let's suggest that network appliances are dedicated hardware & software combinations which have a well defined purpose and which are attached directly to a network. For instance, a router might well be considered a network appliance, but a personal computer would not. A switch, a network attached storage device, a wireless access point, or a network attached webcam would all fit into the definition, although a general purpose computer performing the same functions would not. An IP phone is a network appliance, a telephony enabled PC is not.

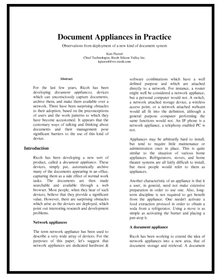
Appliances may be arbitrarily hard to install, but tend to require little maintenance or administration once in place. This is quite similar to the situation of various home appliances. Refrigerators, stoves, and home theater systems are all fairly difficult to install, but most people would refer to them as appliances.

Another characteristic of an appliance is that it a user, in general, need not make extensive preparation in order to use one. Also, long-term discipline is not required to get benefit from the appliance. One needn't activate a food extraction protocol in order to obtain a soda from a refrigerator. Using a stove is as simple as activating the burner and placing a pan atop it.

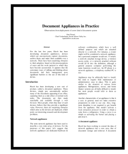
A document appliance

Ricoh has been working to extend the idea of network appliances into a new area, that of document storage and retrieval. A document

(a): s = 1/4
title readable, subsection headings barely readable



(b): s = 1/8
title is barely readable



(c): s = 1/16
headings unreadable, subsection headings barely visible

(d): s = 1/32
subsection headings not visible

Figure 1 — Document information vanishes with scaling to smaller size. Displayed scaling factors represent scaling of width and height of the original letter-size document.

text areas and recognizable images by automatically cropping, scaling and repositioning document objects. Because of those operations, layout distortion is possible which leads to a potential loss of the original page overview.

The second question regarding thumbnail size has been addressed to some extend for photographic images, not for document images. Square thumbnails of size 36 to 80 pixels were tested in a recognition task experiments[15] asking the user to identify the animal contained in the thumbnail. A similar experiment was performed with faces of people. In comparison to the animal identification task, for the face identification task often larger size thumbnails were required. This observation points to the need for an image-content-dependent thumbnail-size framework.

In this paper we investigate answers to questions regarding content and size of document image thumbnails jointly. As for thumbnail content, we select a specific kind of document information, namely document layout information. With focus on that type of document information we propose a solution to the problem of defining an appropriate thumbnail size. The approach is based on a white space analysis of a document page. White space analysis has been used in the literature in initial document layout analysis steps in order to extract structural document information from a document bitmap image[7,14]. We will proceed the opposite way, assume that document layout analysis information is available, and ask how white space analysis can be combined with document analysis information in order to derive content-dependent scaling factors for thumbnails. The goal is to assure visible layout features in a thumbnail. In the following section we first investigate what features, parameters or constrains should be considered when designing thumbnailing algorithms. Section 3 introduces a novel graph model to capture geometrical and image-content-dependent relationships between layout units, as well as display dependencies. The graph model is extended to a hierarchical model in Section 4. Example scenarios of how these models can assist in the task of layout-dependent thumbnail creation are discussed in Section 5. Finally, Section 6 concludes the paper.

2. FEATURE SPACE FOR SMALL-SIZE DOCUMENT VISUALIZATIONS

When fitting an image of pixels size (I_x, I_y) to a smaller display, a target image size (T_x, T_y) is typically fixed by an application or the user. For example, Adobe's document processing application *Acrobat*[1] uses an estimated thumbnail size of 84x102 pixels in the thumbnail view. Apple's *Preview*[13] program let's the user adjust a thumbnail size with a slider between 19x24 to 96x144 pixels. These applications all show thumbnails of the pages of one single document. Adobe's visual file navigation application *Bridge*[2] displays thumbnails of the first pages of all the documents in a certain directory and allows the user to choose between various views: File Navigator, Lightbox, or Filmstrip. Lightbox uses a thumbnail pixel size of 100x130, File Navigator a size of 200x260 and Filmstrip the size 100x130 for a sequence of first pages of documents in addition to an image of bigger size of 345x455 pixels for one user-selected first page.

Some processing on the original image is necessary to fit the image content into the target pixel dimensions. A typically processing step is scaling the image by a factor s that is determined as $s = min(T_x/I_x, T_y/I_y)$. In this case the scaling factor s depends on the pixel sizes of original and target image, but not on the image content. The amount of loss of image signal information caused by the scaling may be described by loss of frequencies in the upper $(1-s)$-range of the frequency spectrum. The scaling factor s, however, is not necessarily proportional to the amount of image information that is lost during scaling. For document images specifically, information such as reading order, column layout, or readability of text can not solely be characterized in the frequency domain. An example is shown in Fig. 1.

A document is scaled by powers-of-two, typical choices for scaling factors in multiresolution image representations, such as JPEG2000[9]. Consider the example of vanishing document specific features in Fig. 1. The readability of section headings is highly reduced when going from s=1/4 to s=1/8. A clear relationship between the chosen sequence of scaling factors and specific loss of document information, however, can not be established. Such content-independent scaling leads to the problem of not having control over the recognizability of document image features in the target image. The goal, therefore, is to derive some methods that allow control over recognizable document image information during the process of scaling.

We first investigate what kind of information loss may occur during scaling of a document image with a factor $s \ll 1$. Semantic meaning of text is lost when text is scaled below readable size. Recognizability of image features is lost when too much scaling is performed. For colored areas recognizability of different colors may be lost due to loss in color contrast during scaling. And page layout information carrying information about semantic flow of a document and hierarchical structure may be lost during scaling. These types of information all fall into categories for which loss can not simply be described by a a cut-off point in the frequency domain.

Another parameter that influences information content in a scaled version of a document image is the flexibility in formatting that is allowed for a given target image. The size (T_x, T_y) can be fixed or can be varying depending on the document content. It is also possible to fix the width, T_x, of the target display and let the T_y be a free parameter. This is done, e.g., when a document is reformatted to fit a smaller display width[6]. No constraint on T_y is given, i.e. scrolling of the text is allowed. Scrolling is also allowed in Summary thumbnails[11], a method developed for web pages that collapses text areas to smaller sizes showing only selected text portions.

A SmartNail[5] representation of a document fits document image content into a fixed target size while focusing on readability of text and recognizability of image features. Instead of choosing one scaling factor for the entire document, different scaling factors are chosen for different document objects. In order to fit a collection of individually scaled objects into the target display the original page layout might be distorted.

In Enhanced Thumbnails[16] a single scaling factor for the entire document image to fit a fixed target size. Special emphasis is placed on keyword display as keywords extracted from the document are pasted onto the scaled image. The contrast of the scaled image is lowered, which may reduce the recognizability of the page layout. Also pasted keywords may cover some document structure such as section headings.

By analyzing different types of document information that might be considered to be preserved in a scaled image we can define a feature space for small-size image visualizations consisting of the following features: readable text, recognizable features of images in documents, color contrast, and recognizable page layout. In addition to these document content features, we define the following target display features: fixed target size, fixed width and flexible height, flexible target size. A summary of published approaches mentioned above for scaling of document images is given in Table 1 categorized regarding different target display parameters and document information features. In this paper we develop a method for scaling that allows for a flexible target size and focuses on recognizable layout information in the scaled image. The method will be explained in detail in the following two sections.

Table 1 — Methods for scaling of document images categorized using feature space dimensions for small-size document image visualizations.

features considered in visualization	**readable text**	**recognizable image features**	**preserved color contrast**	**recognizable overview**	**mixed over-view+detail**	**none**
fixed target size	SmartNails	SmartNails	?	?	Enhanced Thumbnails	traditional scaling
fixed target width, flexible target height	Paper to PDA, Summary Thumbnails	?	?	?	?	?
flexible target size	?	?	?	**this paper (WSG, WST)**	?	?

The categorization in Table 1 addresses the features and visualization space for one document page only. As mentioned in the description of commercial document processing applications at the beginning of this section, thumbnails are usually displayed in groups, either of the various pages of one document or as the first pages of various documents. The expansion of the feature space to include groupings of document pages needs to be considered in the future.

3. CONTENT-DEPENDENT SCALING VIA WHITE SPACE GRAPHS

3.1 The graph model

We start with a single document page **P** and assume that a page description is available as input. This page description should include descriptions of document zones z_i, describing paragraphs, lines, words, and characters. In addition to the document zones also a description of the page background is assumed. For now we assume the document to consist of non-white text, graphics, or images on a white background.

Given a collection of zones, the first step is to determine which of the zones are neighbors in a geometric sense, i.e. we have to find pairs of zones (z_i,z_j), such that the two zones are only separated by white space, not by other zones. The task of finding neighboring document zones is well known in the field of document understanding as part of the process of determining the reading order of a document. We follow the approach from Aiello et al.[4] who propose to use the Voronoi diagram for calculation of neighboring document zones. The centers c_i of the document zones z_i are chosen as the input data points to the Voronoi diagram. Zone centers c_i can be geometric centers or centers of gravity. Two document zones z_i and z_j are neighbors if they share an edge in the Voronoi diagram.

An example of document zones and the corresponding Voronoi diagram is shown in Fig. 2 (middle). Similar to Aiello's[4] approach we capture the neighborhood relationships of the zones in a graph model. The center points of the document zones are vertices. If zones z_i and z_j are neighbors then an edge is created between vertices c_i and c_j. The graph is turned into a weighted graph by associating the separating white space between neighboring zones as a weight w_{ij} to the connecting edge. The separating white space can be computed in various ways. We define a rectangular bounding box b_i of a zone z_i. For non-overlapping rectangular boxes we compute the separating white space by connecting the center points c_i and c_j of the zones by a straight line and determine the intersection of this line with the white space between the two bounding boxes b_i and b_j. The length of the extracted line segment becomes the weight w_{ij} of the edge connecting c_i and c_j. Notice that the weights w_{ij} are greater than zero by definition. As a result we construct a *White Space Graph G* of a set of vertices, edges, and weights, $G = (V, E, W)$, with

$V = \{c_1,...c_n\}$, c_i center of zone z_i,

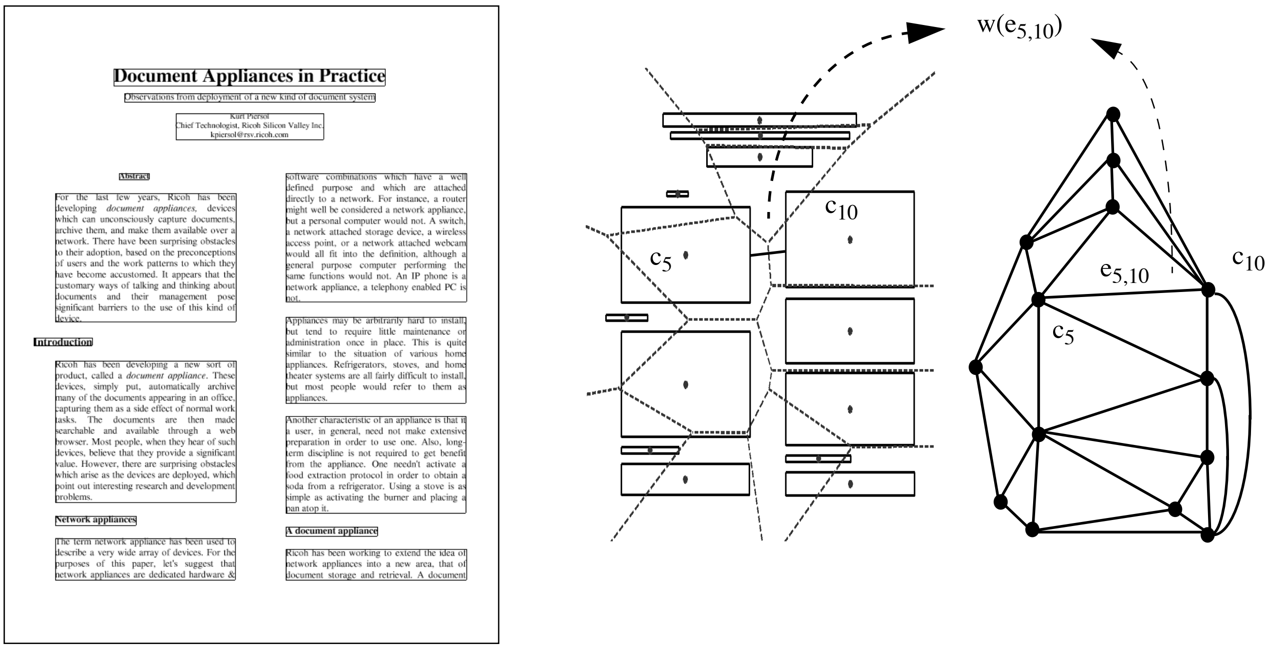


Figure 2 — Example document with bounding boxes of text zones (left), its Voronoi diagram (middle), and the vertices and edges of the corresponding WSG (right).

E = set of edges (pairs of nodes) (c_i, c_j), and

$W = \{w(e) \mid e \in E\}$.

In the weighted graph introduced by Aiello[4] the weights represent the Euclidian distance between center points. Weights in White Space Graphs (WSG), in contrast, represent separating white space. Vertices and edges of the WSG for the example document are shown in Fig. 2 (right). The graph is drawn in a way that the position of the vertices correspond to the center point locations in the document. That means that the length of the edges in the figure do not reflect the weights of the edges. Details of the graph model are explained with the example of the vertices c_5 and c_{10}. These vertices share an edge in the Voronoi diagram (middle). That means they have a connecting edge in the graph model (right). The weight of the edge is represented by the length of the line segment (bold, solid) in the Voronoi diagram.

3.2 Minimal scaling using a WSG

For a given WSG with n vertices we introduce a new variable a_{ij} for $i,j = 1,...,n$. If c_i and c_j are not connected by an edge, a_{ij} is set to zero. If the vertices c_i and c_j are connected by an edge then $a_{ij} = 1/w_{ij}$ ($w_{ij} > 0$ by definition). With this choice for a_{ij} connected vertices with larger separating white space have an value a_{ij} close to zero, the default value for a vertex pair not connected by an edge. For the purpose of this paper we set the diagonal entries a_{ii} to zero.

The goal is now to find a scaling factor such that the separations between all the neighboring document zones are still visually separate in the scaled image. Visual separation may depend on a variety of parameters. Those parameters can be divided into user-specific parameters (e.g. degree of far-sightedness) as well as display specific parameters (e.g. contrast, brightness, resolution) and document specific parameters (e.g. background and foreground colors of a document). In this paper we neglect user-specific parameters, but consider display characteristics as well as background and foreground colors. If the set of display parameters is denoted by D and the set of background and foreground colors is denoted by C we introduce a constant $\varepsilon_{C,D}(i,j)$ that reflects the minimal number of pixels necessary to visually separate two neighboring document zones i and j. Given $\varepsilon_{C,D}(i,j)$ we define a minimal scaling factor s* of a document as

$s^* = \arg\min_{s>0} \{s / a_{ij} \geq \varepsilon_{C,D}(i,j),\ i,j \in \{1,\ldots,n\},\ a_{ij} > 0\}$.

A closed-form expression for s* is given by

$$s^* = \max \{\varepsilon_{C,D}(i,j) \bullet a_{ij},\ a_{ij} > 0\}. \qquad (1)$$

We call the value s* the *minimal scaling factor* for the specific document page **P**. For different documents the minimal scaling factors may differ. In order to emphasize this dependency we denote the minimal scaling factor by s*(**P**).

Computing s* as in Eq. (1) assures that all neighboring document zones are visually separable by the viewer after scaling. For some applications it may, however, be sufficient to assure that selected zones only are visually separate, not all zones. In case of an academic paper formatted in a two-column format, it may be, e.g. desired to just assure visual separations between title and author zones and between the two columns, but not between paragraphs nested in the two columns, or between footer or header zones. In another scenario, it may be desired to show separation of a section heading and the first paragraph of the section, but not necessary the separation between a figure and its figure caption. Both scenarios have in common that separation between elements of a subset of all document zones is desired.

Given a subset $S = \{z_{i1},\ldots,z_{ik}\}$ of all the zones $\{z_1,\ldots z_n\}$, a *conditional minimal scaling factor* $s^*(\mathbf{P}|_S)$ for a document page **P** given the condition that zone elements of S should be visually separate after scaling is derived as

$$s^*(\mathbf{P}|_S) = \max \{\varepsilon_{C,D}(i,j) \bullet a_{ij} \mid z_i, z_j \in S,\ a_{ij} > 0\}. \qquad (2)$$

4. EXTENSION OF WHITE SPACE GRAPHS TO WHITE SPACE TREES

A typical characteristic of document pages is that semantic document information is organized in a hierarchical form. Document layout is used to transform this semantic hierarchical information onto a page image. A title typically stands out by being on the top of the page and is formatted using larger font or a different style, such as bold face or capitalized characters, and is typically followed by a 'body' section. A section heading inside the body section may have smaller font size than the title, but still larger font size or different font style than the following text body paragraph. A figure unit typically consists of a graphic or a photo and a caption. The semantic separation of a figure unit from body text is commonly enforced by placing the figure caption below the associated figure, closer to the figure than to the following text paragraph. That way document layout takes on the role of transforming the semantic document hierarchy information into a visual representation. Separating white space plays an important role in such a visualization of semantic hierarchy.

After reviewing how hierarchically organized semantic information may be transformed into an image we ask the question of how much of that semantic information is in fact included in the white space distribution of a page. Document zones close to each other (e.g. with smaller value in the adjacency matrix) should have a tighter semantic link than document zones far away from each other (e.g. with larger value in the adjacency matrix). In order to analyze the document structure that is created via separating white space we derive a hierarchical white space representation by merging zones with least separating white space together to form a new composite unit, and continue to merge composite units with least separating white space together. As a result, a *White Space Tree (WST)* based on a WSG *G* is created in the following way:

```
Initialize the list of open (available for merging) vertices of G as V_open = V
Set b_ij = a_ij for all v_i,v_j in V_open with a_ij > 0
while V_open ≠ ∅
       {v_i_1,... v_i_k} = arg max_{vi,vj ∈ V_open}(ε_C,D(i,j) • b_ij)
       create new node v* with associated scaling factor
              s(v*) = max ε_C,D(i,j) • b_ij, i,j∈{i_1,...i_k}
       add v_i_1,... v_i_k as children to v*
       remove v_i_1,... v_i_k from V_open
       add v* to V_open
```

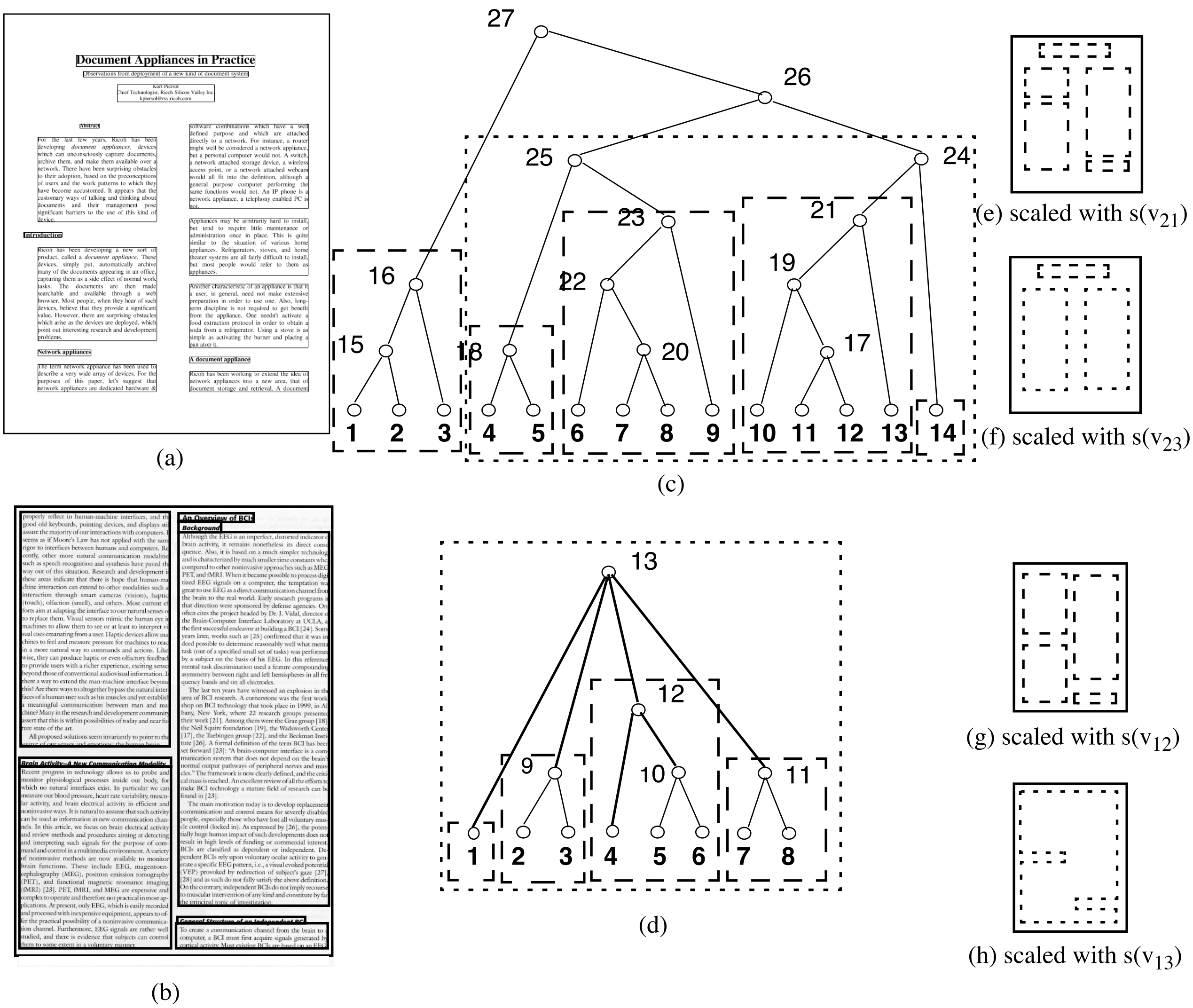


Figure 3 — WST scaling for two example documents (top) and (bottom): document images and zone bounding boxes (a),(b), their White Space Trees (c),(d), and visual zone separations when scaling at a specific node in the tree (e),(f),(g),(h).

As a result, a WST is a tree of nodes v with scaling factors $s(v)$ characterizing coarse-to-fine visual separability of document zones. A WST provides a sequence of document-specific significant scaling factors for a document. Walking the tree from the root to a leaf node means that the scaling factors associated with the nodes will provide a coarse-to-fine scaling of the image where in each step an additional separations between composite units or zones will be visible in the image.

For a good page layout, the composite nodes of a WST should coincide with the nodes in a semantic document tree, but that is not always the case. Examples of WST are shown in Fig. 3. The nodes are numerated with respect to monotonic decreasing scaling factors of the nodes. In the top example the hierarchy represented in the WST correspond to the semantic hierarchy of the document: Section headings are first merged with body text, title lines are merged into title unit (nodes in dashed boxes). Then section+body units are merged into columns (nodes in dotted boxes). Visible zone separations after scaling with scaling factors of two specific nodes are shown. In the second example the WST hierarchy differs from the logical hierarchy: Section headings are merged with body text into new composite units (nodes in dashed boxes). Then some of the composite units from different columns are merged together (nodes in dotted boxes), not following the reading orderAnother use of a WST is the following. Document thumbnails may be displayed as the result of some processing, e.g. grouping of documents with respect to certain features[10]. These features may be of statistical, semantic, or layout specific nature. The thumbnail size is often fixed by the application. If grouping is performed based on features representing layout structures then thumbnails should make sure that those features are visible in them. Using a WSG it can be checked whether the grouping layout feature is actually visible in thumbnails of a specified size or what layout features should be used for grouping in order to assure visibility in the thumbnails.

5. EXPERIMENTS AND DISCUSSION

5.1 Single document WSG scaling factor - dependency on dpi, brightness, colors

In order to demonstrate the dependency of visually separate white space on the display we consider three different displays: an Apple Cinema display (96dpi, 1680x1050 pixels), the display of an iBook laptop (91 dpi, 1024x768 pixels), and an Optoma EP739X DLP projector driven from an iBook at an absolute pixel resolution of 800x600. The brightness on the Cinema and the laptop displays was set to the maximum. Document images used in the experiments are letter-size and have a resolution of 300dpi (2550x3300 pixels).

In order to determine values for $\varepsilon_{C,D}(i,j)$ for all three displays the example document image from Fig. 2 was opened with Adobe Photoshop[3] and a pair of body text lines with an average gray scale intensity for words (20%K) and white background (1%K) selected. Then the image was scaled at various percentage rates using bicubic interpolation. For the cinema display the smallest percentage that still permitted visually distinct lines of plain text was 5%. For the laptop display it was 7.5%, and for the projector 15%. The minimal separate white space between two lines of plain text in those downsampled images was 1 pixel on the cinema display, 1.5 for the laptop display, and 3 pixels on the display for the projector. From this investigation we set $\varepsilon_{C,D} = 1$ for the cinema display, $\varepsilon_{C,D} = 1.5$ for the laptop display, and $\varepsilon_{C,D} = 3$ for the projector. Dependency of $\varepsilon_{C,D}(i,j)$ on specific zone pairs was not used in the experiment, i.e. $\varepsilon_{C,D}(i,j) = \varepsilon_{C,D}$ for all zone pairs. The difference between the constant $\varepsilon_{C,D}$ for Cinema and laptop display did not seem to be related to small difference in dpi resolution, but rather to the different ways the devices are illuminated. The laptop display looked less crisp.

The necessary page descriptions in form of document zones of examples in this paper are obtained from the document layout analysis preformed by the *Expervision* OCR engine[8]. For the example document in Fig. 2, the maximum a_{i*j*} of all values a_{ij} is $a_{i*j*} = a_{1,2} = 0.032242$ measuring white space between the title and the subtitle line. Using Eq. (1), the minimal scaling factor for the example document is $s^* = \varepsilon_{C,D} \bullet 0.032242$. For the Apple Cinema Display with $\varepsilon_{C,D} = 1$ this results in $s^* = 0.032242$, for the laptop display in $s^* = 0.048363$, and for the projector in $s^* = 0.096726$.

5.2 Conditioned WSG scaling factor

To illustrate the effect of the conditioned minimal scaling factors introduced in Eq. (2) we choose an example document (see Fig. 4) containing various document zones: title, authors' names, authors' addresses, body text paragraphs, figure, figure caption, and footers. In the first scenario we assume that the semantic units title and authors' names shall be visually separate after scaling. The resulting minimal scaling factor for the Cinema display is 0.01316. In a second scenario the units figure and figure caption are required to be visually separate after scaling. In this case the resulting minimal scaling factor for the Cinema display is 0.03125 (see Fig. 4).

5.3 WST to determine layout features suitable for grouping

We consider a group consisting of documents all formatted in a two-column layout (Fig. 5). This grouping may be the result of some clustering performed on a larger document collection[10]. We assume further that a specific document viewing or browsing application allocates a specific pixel size for each document thumbnail. We estimated the thumbnail sizes for the following viewing applications: Adobe Acrobat[1], Apple's Preview[13], Apple's Pages[12], and Adobe Bridge[2]. The pixel sizes for these environments are shown in Table 2. In order to check whether the two-column separation is visible in thumbnails of those sizes WSTs were created for each document. From a WST subtrees T_1 and T_2 were selected such that the set leaf nodes of T_1 and T_2 are disjoint and that all zones belonging to the first column are contained in the leaf node set of T_1 and all zones belonging to the second column are contained in the leaf node set of T_2. In addition T_1 and T_2 are selected such that the distance from the root nodes of the subtrees to the root node of the WST is minimal. Now

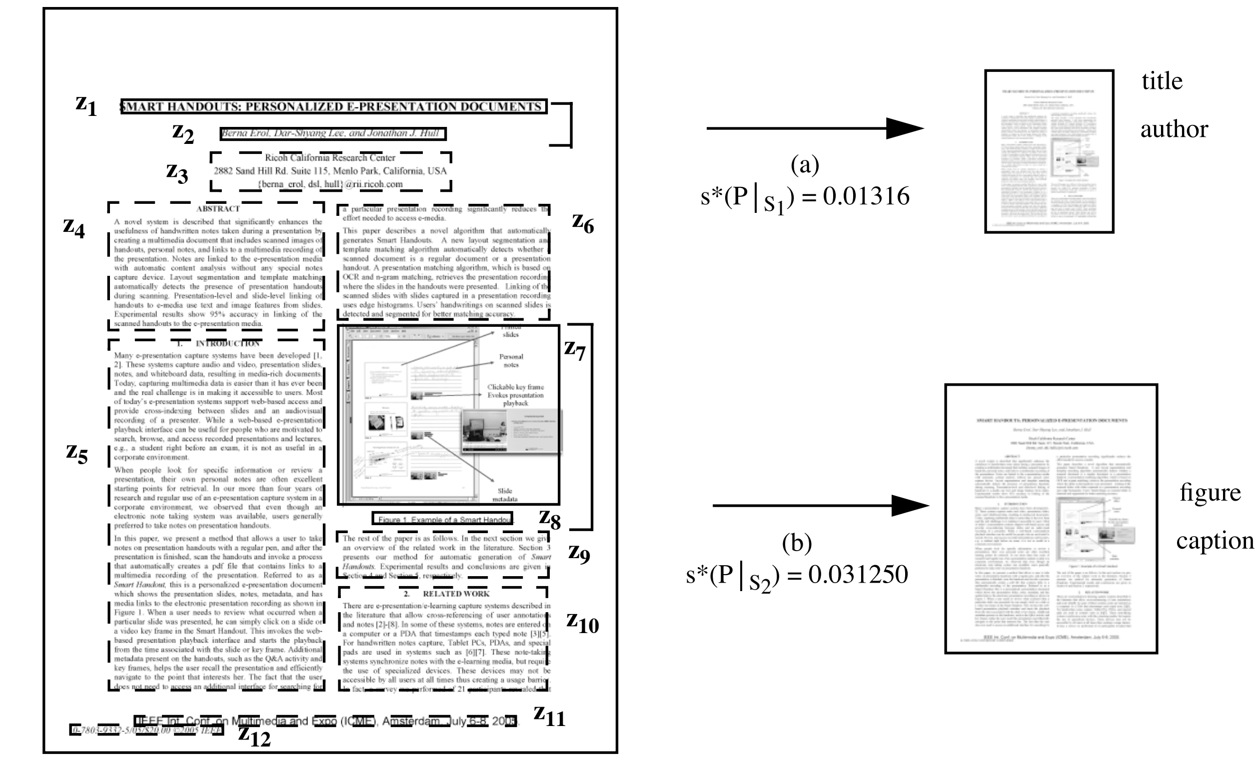


Figure 4 — Example of conditional scaling. Selected zones are in (a) title and authors' names ($S_1 = \{z_1, z_2\}$), and in (b) figure and figure caption ($S_2 = \{z_7, z_8\}$). Conditional minimal scaling factors are for (a) $s^*(P|_{S_1}) = 0.031250$ and for (b) $s^*(P|_{S_2}) = 0.01316$.

the root node n* of the smallest subtree containing both T_1 and T_2 is determined. The scaling factor s_{col} to visually separate the columns is then computed as

$$s_{col} = s(n^*). \tag{3}$$

In order to assist the user to see visual separation beyond the common two-column layout we use the WSG of a document to determine a scaling factor that shows the next largest white space separation inside the two column units. Two subsets S1 and S2 of all vertices are formed by the zones that are contained in column one and two. Then the conditioned minimal scaling factors $s^*(\mathbf{P}|_{S1})$ and $s^*(\mathbf{P}|_{S2})$ are determined from Eq. (2). The combined minimal scaling factor is then chosen as

$$s_{col+} = \max\{s^*(\mathbf{P}|_{S1}), s^*(\mathbf{P}|_{S2})\}. \tag{4}$$

Results for the scaling factors for the column separation (s_{col}), the column subdivision (s_{col+}), and the separation of all zones (s*) for the documents in Fig. 5 are summarized in Table 3. Interpreting the results in the table it becomes clear that the thumbnail sizes used by Acrobat and Bridge are appropriate for most of the scaling tasks when viewing the thumbnail images on a Cinema display, whereas the mean Preview size and the Pages size are not suitable for the task of showing all zone separations. For less crisp displays as for the projector display even Acrobat and Bridge thumbnail sizes may not be sufficient to show selected separations between zones in a column or all zone separations.

5.4 Discussion

With the development of WSG and WST models for a document we developed the concept of layout information dependent scaling of documents based on white space information. Looking at this concept in more detail, there are many areas for future research. We will discuss a few points in the following. As an input we assumed to have a page description in form of text units available. How to obtain such a page description is not a solved problem yet. Whereas layout analysis

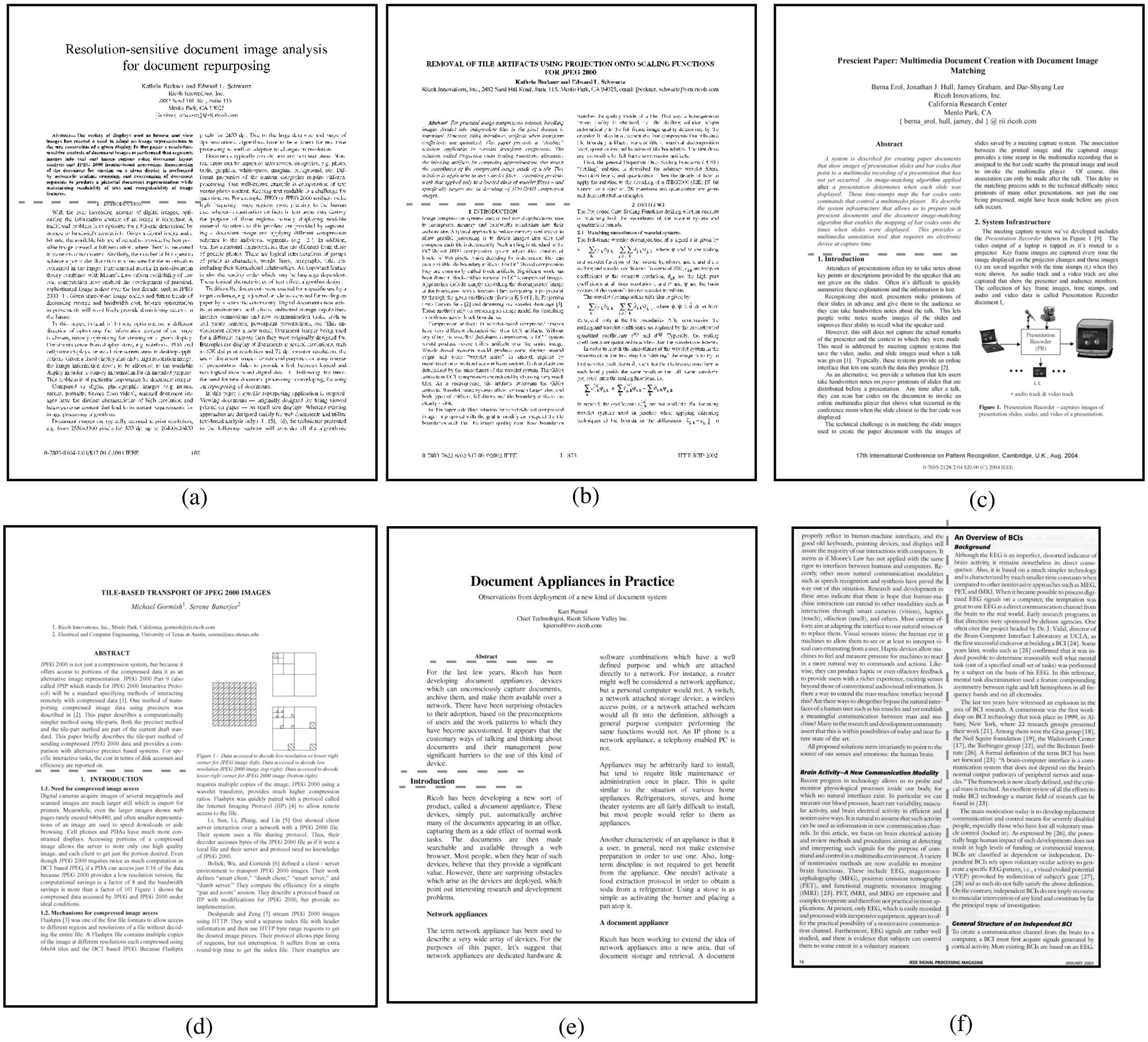
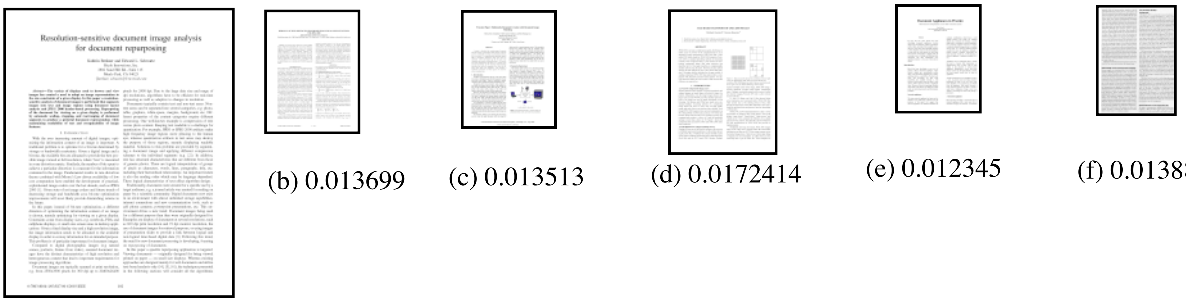


Figure 5 — Example documents with common two-column layout feature. The dotted lines reflect the separations targeted by the scaling in Fig. 6.

tools designed for OCR are sufficient for normal size dark text on light background, detection tools for title units consisting of large, graphical fonts, dark text detection on textured background, and colored text on background of different color than the text are still being researched and analysis tools are not yet widely available.

In this paper we proposed a simple metric for measuring white space separating two text regions by assuming the text to fit into non-overlapping rectangles. For more complicated page layout, this assumption may not hold and more sophisticated distance measures need to be developed. For the display and document dependent parameter $\varepsilon_{C,D}(i,j)$ we fixed the color parameters to be the same for all text zones and all pages of black text on white background, and only investigated dependency only display characteristics, i.e. we set $\varepsilon_{C,D}(i,j) = \varepsilon_{C,D}$. In a next step dependency on color characteristic of individual zones and separating background needs to be included. As a last point, the set of test documents consisted of only one category of documents, namely academic papers. For other categories, e.g. scanned books or handwritten documents, newspapers or letters the WSG/WST framework may have to be modified.



(a) 0.034483 (b) 0.013699 (c) 0.013513 (d) 0.0172414 (e) 0.012345 (f) 0.013888

Figure 6 — Scaling for the documents with common two-column layout feature from Fig. 5 with the task to show the largest subdivision inside column units (dotted lines in Fig. 5), as described by $_{scol+}$ in Eq. (4).

Table 2 — Estimated thumbnail sizes in pixels for various document viewing applications.

Viewing application	**Acrobat**	**Preview**			**Pages**	**Bridge**		
		min	**mean**	**max**		**Lightbox**	**File Navigator**	**Filmstrip**
thumbnail size	84x102	24x19	60x78	96x144	60x78	101x130	199x257	345x455

Table 3 — Scaling factors s*, s_{col} and s_{col+} and corresponding thumbnail sizes for the documents in Fig. 5 for viewing on Cinema and projector display.

	(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
s_{col} (Eq. (3)) for Cinema display	0.02127	0.011364	0.010204	0.010526	0.003831	0.013888
s_{col} (Eq. (3)) for projector	0.06381	0.034092	0.030612	0.031578	0.011493	0.041664
thumbnail pixel size for Cinema display	54 x 70	29x38	26x34	27x35	10x13	35 x 46
thumbnail pixel size for projector	162x210	87x152	78x102	81x105	30x39	105 x 138
s_{col+} (Eq. (4)) for Cinema display	0.034483	0.013699	0.013513	0.0172414	0.012345	0.013888
s_{col+} (Eq. (4)) for projector	0.103449	0.041097	0.040539	0.0517242	0.037035	0.041664
thumbnail pixel size for Cinema display	88x114	35x45	34x44	44x57	31x41	35 x 46
thumbnail pixel size for projector	264x342	105x135	102x132	132x171	93x123	105 x 138
s* (Eq. (1)) for Cinema display	0.034483	0.028571	0.02	0.038461	0.032242	0.038461
s* (Eq. (1)) for projector display	0.103449	0.085713	0.06	0.115383	0.096726	0.115383
thumbnail pixel size for Cinema display	88x114	73x94	51x66	98x123	82x106	98x123
thumbnail pixel size for projector	264x342	219x282	153x198	294x369	246x318	294x369

Looking back at the feature space categorization in Table 1, WSG and WST just filled one of the entries that had not been considered in the literature up to now. How to develop thumbnail creating methods that would fill the remaining "?"-cells in the table we did not address in this paper.

6. CONCLUSIONS

This paper introduces the concept of document layout dependent scaling of document images in order to enable control over visible layout structures in thumbnails. Content-dependent scaling factors were derived based on the analysis of white space separating neighboring document zones. As a result of this analysis White Space Graph and White Space Tree models have been developed and usage of those models in selected application scenarios has been demonstrated. Although only a narrow selection of documents has been analyzed and a limited set of application scenarios discussed, many research questions for future work are identified in the area of content-dependent scaling of documents.

REFERENCES

[1] Adobe Acrobat 7.0 Professional, Version 7.0.0, www.adobe.com/products/acrobat/, Adobe Systems Incorporated, San Jose, CA.

[2] Adobe Bridge CS2, Version 1.0.1.46, www.adobe.com/products/creativesuite/bridge.html, Adobe Systems Incorporated, San Jose, CA.

[3] Adobe Photoshop CS2, Version 9.0, www.adobe.com/products/photoshop/main.html, Adobe Systems Incorporated, San Jose, CA.

[4] Aiello M., Monz, C., Todoran, L., Worring, M., "Document Understanding for a Broad Class of Documents," International Journal on Document Analysis and Recognition, **5(1)**, pp. 1-16, 2002.

[5] Berkner, K., Schwartz, E. L., Marle, C., "SmartNails - Image and Display Dependent Thumbnails," *Proceedings of SPIE*, San Jose, CA, vol. 5296, pp. 53-65, 2004.

[6] Breuel, T., Janssen, W., Popat, K., Baird, H., "Paper to PDA," *Proceedings of 16th International Conference on Pattern Recognition*, Quebec City, Canada, vol. 4, pp. 476-479, 2002.

[7] Breuel, T., "An Algorithm for Finding Maximal Whitespace Rectangles at Arbitrary Orientations for Document Layout Analysis," *Proceedings of International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, vol. 1, pp. 66-70, 2003.

[8] Expervision, www.expervision.com, Fremont, CA.

[9] Gormish, M.J., Lee, D., Marcellin, M.W., "JPEG 2000: Overview, Architecture, and Applications," *Int. Conf. on Image Processing*, Vancouver, Canada, vol. 2, pp. 29-32, 2000.

[10] Hu, J., Kashi, R., Wilfong, G., "Document Image Layout Comparison and Classification," *Proceedings of the International Conference on Document Analysis and Recognition*, Bangalore, India, pp. 285-288,1999.

[11] Lam, H., Baudisch, P., "Summary thumbnails: readable overviews for small screen web browsers, "*Proceedings of the SIGCHI conference on Human factors in computing systems*, Portland, Oregon, pp. 681 - 690, 2005.

[12] Pages, Version 1.0.2, www.apple.com/iwork/pages/, Apple Computer Inc., Cupertino, CA.

[13] Preview, Version 3.0.3, www.apple.com, Apple Computer Inc., Cupertino, CA.

[14] Rus, D., Summers, K., "Using White Space for Automated Document Structuring," Technical Report: TR94-1452, Cornell University, Ithaca, NY, 1994.

[15] Suh, B. et al., "Automatic Thumbnail Cropping and its Effectiveness," *Proceedings of the 16th annual ACM symposium on User interface software and technology*, Vancouver, Canada, pp. 95 - 104, 2003.

[16] Suh, B., Woodruff, A., Rosenholtz, R., Glass, A., "Popout Prism: Adding Perceptual Principles to Overview+Detail Document Interfaces," *Proceedings of the SIGCHI conference on Human factors in computing systems*, Minneapolis, Minnesota, pp. 251-258, 2002.