IJDAR (2005) 7: 228–240
Digital Object Identifier (DOI) 10.1007/s10032-004-0135-2

International Journal on Document Analysis and Recognition
IJDAR

The UvA color document dataset

Leon Todoran, Marcel Worring, Arnold W.M. Smeulders

Intelligent Sensory Information Systems, University of Amsterdam,
Kruislaan 403, 1098SJ Amsterdam, The Netherlands
(e-mail: {todoran, worring, smeulders}@science.uva.nl; http://www.science.uva.nl/~{todoran,worring,smeulders}

Received: 15 April 2002 / Accepted 25 February 2004
Published online: 2 February 2005 –

Abstract. Publications on color document image analysis present results on small, nonpublicly available datasets. In this paper we propose a well-defined and groundtruthed color dataset consisting of over 1000 pages, with associated tools for evaluation. As we focus on aspects specific to color documents, we leave out the document textual content in the ground truth. The color data groundtruthing and evaluation tools are based on a well-defined document model, complexity measures to assess the inherent difficulty of analyzing a page, and well-founded evaluation measures. Together they form a suitable basis for evaluating diverse applications in color document analysis. Both the dataset and the tools are available through our Web site.[1]

1 Introduction

Color now plays an important role in publishing everything from scientific journals, newspapers, and magazines to advertisements. The nature of documents in current document scanning applications is therefore rapidly shifting from simple black-and-white documents to complex color documents. Possible commercial applications of color document analysis are: analysis of advertisements, information retrieval from Internet pictures, color document compression, and reuse of information from color magazines.

Some tools for color documents such as color OCR [4, 19, 24], color document compression [2], and color string localization [3,5,7,15] have been developed. However, whereas document analysis for black-and-white documents is mature, color document analysis is still in its infancy.

Two factors have been instrumental in advancing the field of black-and-white document analysis. Firstly, the existence of public domain datasets like the UW [11] and MTDB [17] has freed researchers from the labor-intensive task of creating datasets to work on. Secondly, the availability of standard evaluation tools for OCR and page segmentation [12,18,25] has allowed for knowledge exchange between different researchers.

For color document image analysis, no such dataset standardization has taken place. The MDTB dataset does contain some color pages. Their layout is, however, so simple that their structure is not essentially different from black-and-white documents. Also the ground truth does not include any color information. As a consequence, each developer now uses its own color dataset for evaluating tools. Typically the datasets used are small, as providing a ground truth for color documents is a time-consuming task. In this paper we report on the creation of a large dataset with ground truth that could be a first step in standardizing the evaluation of color document analysis.

The dataset consists of over 1000 pages with ground truth describing the document components, their layout, and their logical structure. As we focus on aspects specific to color documents, we leave out the document textual content in the ground truth. In fact, we make the assumption that whenever a system can reliably decompose a document into its constituent components and their structure, existing OCR methods can extract the content from a text zone. Hence methods for this task are not essentially different from black-and-white methods.

The documents in the dataset show a great variety in complexity, ranging from simple one-column pages with one picture to pages with several layers of document objects with multiple overlapping pictures. It is important to be able to quantify the complexity of a document in the collection prior to evaluation. If the complexity of documents in a dataset is known and well defined, the complexity measures can be used to weight the evaluation results leading to evaluation independent of page difficulty [6].

Some papers refer explicitly to the document complexity. For instance, in [26] Zhong et al. define a complex document image as "an image where the characters cannot be segmented by simple thresholding, and the color, size, font, and orientation of the text are un-

[1] http://www.science.uva.nl/UvA-CDD

known." Chen defines complex images as "those in which text blocks are overlaid on images or graphics" [3].

It should be noted here that complexity is task dependent. A document can be simple for one task while being very difficult for another. Therefore, there is a need for a set of measures that collectively cover the whole document analysis process. Such a set of complexity measures would rank the data, but evaluation measures are needed to assess the algorithm's performance on those data. The existing evaluation methods for layout analysis can be grouped into two main categories: text-based and region-based evaluation. Text-based evaluation [9] uses textual ground truth and the edit distance to measure the errors in layout detection. Region-based evaluation methods [10–12,25] compare the outline of the detected zones with the zone description in the ground truth. As noted, we do not consider textual content. Thus, the region-based methods are best suited for evaluating document analysis algorithms. Furthermore, they can easily be applied to text, pictures, and graphics. We do, however, need to extend these measures to color document analysis.

This paper is organized as follows. In Sect. 2 we describe the dataset and a model for its content. Section 3.4 makes precise the complexity of the documents with respect to the different tasks in color document analysis. For each of these tasks an appropriate evaluation measure is derived in Sect. 4. Finally, Sect. 5 discusses how the ground truth is generated and which tools have been implemented to support ground truth definition and evaluation.

2 Document dataset

In this section we describe the documents that comprise the document dataset. We then define models to describe the content of each document.

2.1 Dataset content

A dataset for the evaluation of color document analysis must cover different applications consisting of document pages of varying style and complexity. Furthermore, in the documents considered color must be an essential component of the message the author wants to convey. Otherwise, the document is probably equivalent to a black-and-white document. We found that commercial color magazines form the most representative category of color documents. Even inside a single issue the document pages show a great variety in style, ranging from simple pages containing text only to highly complex color advertisements. Especially in the latter category of pages, the color is chosen carefully to attract the readers' attention. A system tested well on such a dataset will perform well on most other applications.

For the UvA color document dataset (**UvA-CDD**), we have scanned full issues of internationally available magazines: Cosmopolitan, Time, Newsweek, National Geographic, IEEE Spectrum, The New Yorker, and IEEE Computer. They are representatives of scientific magazines, informative magazines, lifestyle magazines, and weekly news magazines. The issues together form a dataset of more than 1000 scanned pages.

The document pages were scanned with a Hewlett Packard ScanJet Scanner. In order to reduce transparency noise, a black sheet of paper was placed on the back of the scanned page. The scanning resolution was 300 dpi with 24 bits of color information per pixel. In uncompressed TIFF format this requires a total space of 23.3 GB. We have also created a JPEG compressed version of the dataset. To that end we used a JPEG compression quality factor of 75%, which is the recommended ratio [22] for preserving image quality while providing fair compression. In this format the dataset totals 1.1 GB.

The JPEG compressed dataset is made available via a Web site. Access to this site is restricted to registered researchers. To use the images in publications, each author should individually seek permission from the magazines' publication office.

2.2 The document model

For defining the ground truth, which provides the basis for evaluation, a document model is needed that captures all essential information in the document.

The model should be based on two different views of the document: the layout information – encoding the presentation of the document – and the logical information – encoding the meaning of the document.

The basic entities in both views are the n document objects in the document object set $\mathcal{O}$:

$$\mathcal{O} = \{o_1, o_2, ..., o_n\},$$

which hold the **content** of the document. Each document object is an entity in which the content has a uniform style expressing some intention of the author. So, an element in $\mathcal{O}$ can, for example, be a single picture used as illustration, a text line in bold acting as a header, or a line in red used as a separator.

The two different views of the content of a document object use different attributes to describe the content. As indicated earlier, the attributes should describe the content appearance and meaning, but not the actual content like ASCII codes for a text. Therefore, layout attributes are restricted to the geometric and color properties of the document objects. Logical attributes are functional labels expressing the function of the document object in the document. The object sets $\mathcal{O}_g$ and $\mathcal{O}_l$ denote the set $\mathcal{O}$ with geometric and logical attributes added, respectively.

An element in $\mathcal{O}$ does not appear in isolation, but an author adds structure to the set $\mathcal{O}$. At creation time the author first defines the logical structure $\mathcal{L}$ of the document. In what order are the document entities to be read? Which figure and caption belong together? Only when this has been established the author starts placing the document objects on the page yielding the layout structure $\mathcal{G}$.

In black-and-white documents the layout structure is often of a rather simple nature and document objects do not overlap. Tree-based representations have been in common use. For color documents the author can use layers to organize content, where document objects within a layer do not overlap, but between layers they do. The layer assignment is not unique, and, furthermore, the author can also move document objects forward or backward at will. Therefore, for analysis purposes, not the layers themselves should be encoded but the spatial relations between the document objects. Tree-based representations are too limited to describe such complex relations, hence a graph-based representation must be used.

A simple graph cannot describe all possible spatial relations among document objects. A directed labeled multigraph is used to describe relations like overlap and inclusion. Thus the layout structure is given by a multigraph where the vertices are the document objects $\mathcal{O}_g$ and the edges $\mathcal{R}_g$ denote a relation between the objects. The graph can be directed or undirected and can have weights to encode attributes of the edges. Thus, the layout structure is defined as follows:

$$\mathcal{G} = \langle \mathcal{O}_g, \mathcal{R}_g \rangle .$$

Similarly the logical structure is defined as:

$$\mathcal{L} = \langle \mathcal{O}_l, \mathcal{R}_l \rangle .$$

Although the logical structure (and sometimes layout) can span more than one page, we use, for simplicity, a page-based approach where every page receives a layout and logical structure. So a full document $\mathcal{D}$ is represented by:

$$\mathcal{D} = \langle (\mathcal{G}_1, \mathcal{L}_1), (\mathcal{G}_2, \mathcal{L}_2) \ldots \rangle .$$

In the following subsections we describe how the generic model defined above is instantiated to describe the ground truth for the dataset.

2.3 Geometric description

For the geometric description of a document we consider three major different categories of document objects, namely, text, image, and graphics.

In the description of the outline of these objects we make a distinction between the *perceived shape* and the *real shape* of a document object. The real shape describes the boundary of the object in the document image. The perceived shape is the boundary of the object as perceived by a human. That is, in a layered document, the perceived shape of a partially obscured document object is the whole object, without missing parts or holes. An illustrative example can be seen in Fig. 1. In the following discussion, the object itself will be indicated as o, the perceived shape of the object as $\overline{o}$, and the real shape of the object as $\hat{o}$. In a similar way $\hat{O}$, where O is a set of objects, denotes the set of real shapes of objects O.

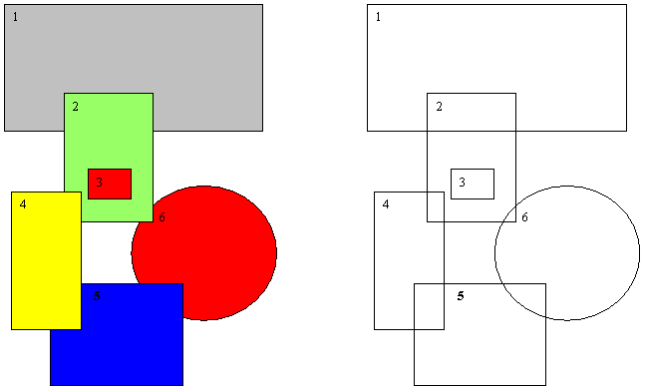

Fig. 1. In the figure on the *left* a document consisting of six document objects is depicted, where for each object the real shape is shown. In the figure on the *right* the same objects are shown, but now their perceived shape is drawn. Note that, for example, $o_2 \succ_t o_1$ and $o_3 \prec_w o_2$

Now considering the text objects, recall from the introduction that we focus on properties of the document that are specific to color documents. Therefore, we do consider color characteristics of textual document objects but not font style or size. To be precise, to describe a geometric document object, the following attributes are used:

- Geometric attributes;
  - Category: {text, image, graphics};
  - Perceived shape
    - Line: endpoints
    - Rectangle: top-left, bottom-right corners
    - Polygon: list of points
    - Ellipse: x,y-position of the center and size of short and long axes
  - Real shape: set of polygons with possible holes
  - Orientation: horizontal, vertical, other
- Color attributes for text objects
  - Text: {uniform, mixture of two or more uniform colors, texture}
  - Background: {uniform, mixture of two or more uniform colors, image, texture}

Note that any other complex shape not listed above will be represented as a polygon. For later use, let us define notations for the following subsets of geometric document objects based on individual categories and one mixed class for pictorial information:
$T = \{o \in \mathcal{O}_g | category(o) = text\}$
$G = \{o \in \mathcal{O}_g | category(o) = graphics\}$
$I = \{o \in \mathcal{O}_g | category(o) = image\}$
$P = G \cup I$

and with respect to the shape of the document object:
$\mathcal{O}_g^R = \{o \in \mathcal{O}_g | shape(o) = rectangle\}$
$\mathcal{O}_g^L = \{o \in \mathcal{O}_g | shape(o) = line\}$
$\mathcal{O}_g^P = \{o \in \mathcal{O}_g | shape(o) = polygon\}$
$\mathcal{O}_g^E = \{o \in \mathcal{O}_g | shape(o) = ellipse\}$

For text document objects we introduce some shorthand notations to indicate different classes based on the

color of the text and the background on which it is placed. To that end, we use the generic notation $\mathcal{T}_f^b$ indicating a text object with foreground type f and background type b. Choices for f and b are uniform (u), nonuniform ($\neg u$), graphic (g), image (i), or arbitrary (), the latter indicating that the foreground or background can be any of the given types. As an example, $\mathcal{T}_{\neg u}$ is the set of nonuniform text strings on an arbitrary background.

The geometric structure of the document is the structure induced by the layers in the document. From there one can also define the structure within a layer, but that is not considered here. Edges in the geometric structure graph are defined by the *on-top* relation, indicating that the object is in a higher layer. The relation is formally defined as:

$$o_1 \succ_t o_2 \Leftrightarrow \overline{o}_1 \cap \overline{o}_2 \cap \hat{o}_1 \neq \emptyset \,.$$

The above formulation is applicable both when the perceived shape of the two objects have a partial overlap and when one fully contains the other. To make the distinction, we explicitly introduce the relation *within*, denoted by $\prec_W$, which indicates that the perceived shape of one object is fully contained within the area of the other:

$$o_1 \prec_w o_2 \Leftrightarrow \overline{o}_1 \subset \hat{o}_2 \,.$$

On the basis of the above relation, we define two layout structure relations, the first dealing with overlapping objects, the other with included objects:

$$\mathcal{R}_g^s = \{(o_1, o_2) \in \mathcal{O}_g \times \mathcal{O}_g | \overline{o}_1 \succ_t \overline{o}_2 \wedge \neg(o_1 \prec_w o_2)\} \,,$$

$$\mathcal{R}_g^w = \{(o_1, o_2) \in \mathcal{O}_g \times \mathcal{O}_g | \overline{o}_1 \succ_t \overline{o}_2 \wedge o_1 \prec_w o_2\} \,.$$

Finally, $\mathcal{R}_g = \mathcal{R}_g^s \cup \mathcal{R}_g^w$.

The two relations are explained in Fig. 1.

In the creation of the document the author is free to define as many layers as desired, only adhering to all desired on-top relations. For a consistent definition of the ground truth a well-defined layer definition is required.

Layers are defined based on the graph of on-top relations $\mathcal{R}_g$ as follows. In the graph $\mathcal{R}_g$ all paths connecting document objects $o \in \mathcal{O}_g$ are detected. Each layer is identified by an index. The layer with index zero, also called the "paper layer," is the lowest in the layer hierarchy. A document object $o \in \mathcal{O}_g$ is assigned to the layer with index z, where z is the maximum number of predecessors on any of the paths that reaches o in the graph. When a cycle exists in the graph of on-top relations, no consistent layer definition exists. We restrict ourselves to documents in which there are no cycles in the graph.

Note that other layout relations, like "above," "to the left" can be easily defined later as the ground truth information already has all the required spatial information.

2.4 Logical description

After an analysis of the magazines in the dataset, for each type of document object a set of possible representative logical labels are selected. Object classes that do not appear frequently in the dataset receive the label "**Other**." Of course, they could be refined later. This leads to:

- Logical attributes
  - Category: {text, image, graphics}
  - Logical label
    - Text: {author, abstract, bibliography, caption, equation, header, footer, footnote, list, table, title, quote, paragraph, page number, advertisement, note, other}
    - Image: {advertisement, image containing scene text,[2] other}
    - Graphics: {separator, border, logo, map, barcode, graph, other}

All of the above document objects with their logical labels could be part of the logical structure of the document. As reading order is most important, we focus on this particular structure.

We have chosen reading order also because it is representative of page-based analysis. The reading order is based on the relation *before in reading* denoted by $\preceq_r$. So the logical structure graph has as vertices the logical document objects O_l, and there is a directed edge between $o_1, o_2 \in \mathcal{O}_l$ whenever $o_1 \preceq_r o_2$. To be a proper reading order graph it should be acyclic. Then, a path in the graph is an independent reading order in the document. When there are multiple paths in the graph, they are related to groups of document objects that can be read in arbitrary order. So for the logical structure we have:

$$\mathcal{R}_l = \{(o_1, o_2) \in \mathcal{O}_l | o_1 \preceq_r o_2\} \,.$$

3 Document complexity

The performance of an algorithm on a given dataset depends on two things: the quality of the algorithm itself and the complexity of the data. This complexity is task dependent. When the ground truth is available, the complexity can be computed beforehand. It can then be used to order the documents in the dataset so that one can choose a certain level of complexity for designing and testing the algorithm.

Before defining such a set of complexity measures we first consider which steps are performed when doing color document analysis.

3.1 Document analysis steps

We decompose color document analysis into four major steps. The first two deal with the geometric aspects of the documents; the third and fourth steps deal with the logical content of the document.

[2] It can be argued that this is a geometric rather than a logical label. However, to find scene text, substantial interpretation of the image is required.

– *Page segmentation*: determination of the set of geometric document objects $\mathcal{O}_g$.

In this step the page is decomposed into text zones, image zones, and graphics zones. For the resulting objects the attributes are computed.

– *Layout detection*: determination of the relation $\mathcal{R}_g$.

This process yields the layered structure of the document captured in the relations between document objects.

– *Logical object classification*: determination of the set of logical document objects $\mathcal{O}_l$.

Logical labels for each of the different categories of objects are assigned to the document objects.

– *Reading order detection*: determination of the relation $\mathcal{R}_l$.

At this point in the process the vertices and edges of both the geometric and logical graphs are computed.

For each of the steps a complexity measure will be derived:

– C_1: Complexity of page segmentation
– C_2: Complexity of layout detection
– C_3: Complexity of logical object classification
– C_4: Complexity of reading order detection

The above measures are all defined for a document page and can be computed from the ground truth graphs corresponding to the page. For a document, the complexity of each task is computed by averaging the complexities of individual pages. The different tasks and their complexity measures are illustrated in Fig. 2.

3.2 Document complexity for page segmentation

In analyzing the difficulties of the page segmentation algorithms described in the literature [14, 16, 23], we identify four main factors that influence the quality of the results. They are:

1. *Nonuniformity in color*: If the color of a text string is nonuniform or placed on a colored background, it is much harder to segment the text from its background.
2. *Shape irregularity*: Most documents are based on rectangular document objects. If documents do not conform to this general style they are more difficult to segment.

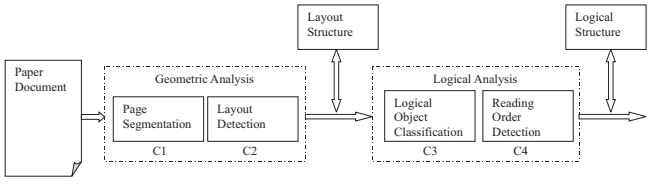

Fig. 2. The four main tasks in color document image analysis, with their associated complexities

3. *Picture/text ratio*: Pictures contain a much wider range of colors than most text strings. They are much harder to identify by their color characteristics.
4. *Amount of pictorial document objects containing text*: Scene text or text in a graphical object can cause problems as they have similar characteristics as genuine text strings in the document.

Taking into account the above factors, we consider a document page containing only uniformly colored text objects and having rectangular shapes on a uniform background to have a complexity of zero. An example of a document page of maximum complexity is one containing an image in the background, completely covering the page, with text objects with nonuniform color and irregularly shaped boundaries placed on top of it. For each of the four factors we have designed a complexity measure.

The first measure is based on the text strings that are either not uniformly colored or have a nonuniform background. Using the shorthand notations from Sect. 2.3:

$$c_1^1 = \frac{Area(\mathcal{T}_u^g) + Area(\mathcal{T}_u^i) + Area(\mathcal{T}_{\neg u})}{Area(\mathcal{T})}\,. \qquad (1)$$

The second measure considers the percentage of irregular shapes:

$$c_1^2 = \frac{Area(\mathcal{O}_g^P) + Area(\mathcal{O}_g^E)}{Area(\mathcal{O}_g)}\,. \qquad (2)$$

The third complexity measure considers the area of the geometric union of all the shapes corresponding to pictorial document objects, normalized by the width (w) and height (h) of the page:

$$c_1^3 = \frac{Area(\bigcup_{o \in P} \hat{o})}{w * h}\,. \qquad (3)$$

Finally, the fourth measure considers the subset of graphics and image objects containing text, denoted by P^{ct}:

$$c_1^4 = \frac{Area(P^{ct})}{Area(P)}\,. \qquad (4)$$

Each of the four complexity measures are normalized to yield values in the range [0,1]. Note that we use the area of objects in a set instead of the cardinality of the set. This can be seen as a weighted mean value, where large objects contribute more to the complexity than small objects.

For the sake of simplicity the complexity C_1 for page segmentation is defined as a linear combination of the four complexity features defined above. Weights could be used to emphasize one of the four components. Here, we consider them equally important:

$$C_1 = \frac{c_1^1 + c_1^2 + c_1^3 + c_2^4}{4} . \quad (5)$$

3.3 Document complexity for layout detection

The problem of detecting multiple layers in color documents has, to our knowledge, not been addressed. The DjVu system [2] could be seen as an exception; however, the system is restricted to one foreground and one background layer, and, more importantly, the goal is compression, not analysis.

As defined in Sect. 2.3, the geometric structure is based on the observation that we perceive a regularly shaped object as the full object even if it is partly occluded. Clearly the larger the occlusion, the less applicable this observation. Therefore, to measure the complexity of the decision on whether two elements overlap, we consider the area of the intersection relative to the union of the two objects. Subsequently this is summed over all object pairs:

$$C_2 = \frac{1}{|\mathcal{R}_g|} \sum_{o_1 \neq o_2} \left\{ \frac{Area(\bar{o}_1 \cap \bar{o}_2)}{Area(\bar{o}_1 \cup \bar{o}_2)} \right\} . \quad (6)$$

It follows that for layout structure detection a document has a complexity of zero when none of the objects in the document have an overlap. A document of maximum complexity (1.0), although not realistic, is a document consisting of two objects having a partial overlap almost equal to one of the object's perceived shapes.

3.4 Document complexity for logical object classification

In general, logical object classification is based on layout features, i.e., visual appearance, content, and possible a priori information about the document class. As indicated earlier, we do not consider document content. Furthermore, a priori information cannot be made part of the ground truth as it is user and application dependent. Therefore, for deriving a complexity measure we use visual appearance only.

The complexity of the classification problem is determined by the similarity in visual appearance within a logical class and the dissimilarity between different logical classes. However, variability and separability depend on the geometric features used and on the classification method. As we want the complexity measure to be independent of the specific method used, we focus on the number of different classes on the page that have to be distinguished. We do so separately for text, images, and graphics so that they can be weighted differently.

To be precise, let L_t denote the set of possible text labels for logical objects and let L_i and L_g be defined likewise for image labels and graphics labels. Furthermore, let L' denote the set of labels actually present on the page. Then the complexity measure for logical labeling is given as:

$$C_3 = \frac{1}{3} \left\{ \frac{|L'_t|}{|L_t|} + \frac{|L'_i|}{|L_i|} + \frac{|L'_g|}{|L_g|} \right\} . \quad (7)$$

Obviously the only documents with $C_3 = 0$ are empty pages. The most complex ones ($C_3 = 1$) are documents with all classes of text, image, and graphics appearing at least once in the document.

3.5 Document complexity for reading order detection

In analyzing existing methods for reading order detection [20, 21], it is observed that methods work well if document objects are nicely ordered, e.g., in a column. Performance degrades if the reading order "jumps" from one object to another in an irregular way. To that end we derive a complexity measure that measures the irregularity of the reading path when visiting the different text objects in the document.

Recall that the reading order is defined through the before in reading order relation $\preceq_r$. Each maximal path in the graph with edges defined through the before in reading relation gives an independent reading path. Thus we can write the relation $\mathcal{R}_l$ as $\{r_o, r_1, \ldots\}$, where each

$$r_i = (o_1, o_2, \ldots, o_{m(i)})$$

is such a maximal path in the graph.

We now define a measure of irregularity for a path r_i. First, note that we cannot rely on the first and last word of the block as we aim at measures that are independent of the content. Therefore, we consider the polyline with vertices p_j for $j = 1, m(i)$ that results if one connects the centers of gravity of the subsequent document objects in r_i. For analysis of reading order, based on geometric information, the simplest assumption one can make is that for finding p_{j+1} from p_j one continues in the direction of the vector from p_{j-1} to p_j. If this is the case we assign a complexity of zero. In general cases, the point is found in a different direction. Therefore, we define the turning angle α_j at p_j as the angle between the expected direction and the actual direction in which p_{j+1} can be found. Locally the complexity is maximal if one has to search in exactly the opposite direction that one came from. The turning angle can be computed using the inner product as:

$$\alpha(j) = \cos^{-1} \frac{|\overrightarrow{p_{j-1}, p_j} \cdot \overrightarrow{p_j, p_{j+1}}|}{|\overrightarrow{p_{j-1}, p_j}||\overrightarrow{p_j, p_{j+1}}|} . \quad (8)$$

Note that the angle is defined for all but the first and last point on the path.

For a page, the average turning angle on any path is computed. Normalizing to [0,1], the complexity measure for reading order detection is given by:

$$C_4 = \sum_{i=1}^{|\mathcal{R}|} \left(\frac{1}{(m(i)-2)\pi} \sum_{j=2}^{m(i)} \alpha(j) \right) . \quad (9)$$

Note that C_4 cannot be computed for a reading order containing two elements. As in such cases deriving the reading order is mostly trivial, we assign $C_4 = 0$ in such cases.

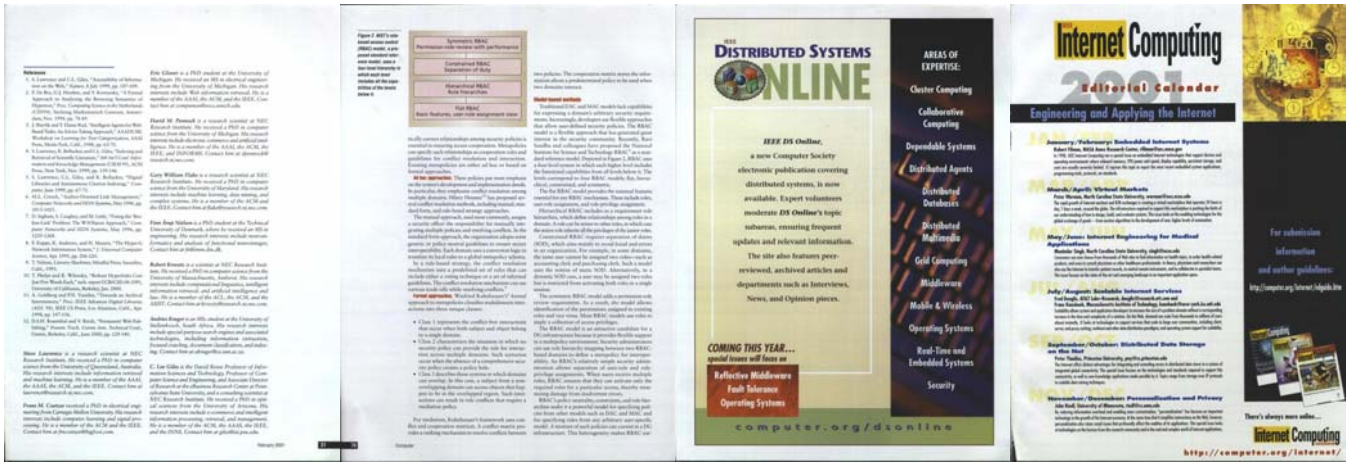

Fig. 3. Images of increasing complexity for page segmentation ($C_1 = 0.0, 0.08, 0.13, 0.15$), ranging from a simple page containing only text with a uniform background and an image to a page with text written on graphics and polygonal shaped objects. All images copyright ©2001 IEEE

Fig. 4. Examples of documents with increasing complexity for layout detection ($C_2 = 0.0, 0.14, 0.17, 0.29$). The simplest examples have document objects all of which have a rectangular outline that is fully visible. In the most complex examples, the occluded area is a significant part of the occluded object. All images copyright ©2001 IEEE

Fig. 5. Different documents with increasing complexity ($C_3 = 0.03, 0.10, 0.17, 0.48.$) for logical classification. The first document has one logical label only, whereas the last document has 12 different labels. All images copyright ©2001 IEEE

Fig. 6. Documents with increasing complexity for reading order detection ($C_4 = 0.0, 0.23, 0.51, 0.65.$). Paths clearly range from regular to very irregular. All images copyright ©2001 IEEE

Table 1. Average complexity values for the UvA color document dataset

Magazine	*Pages*	C_1 *Page segm.*	C_2 *Layout detect.*	C_3 *Log. obj. detection*	C_4 *Reading order det.*
Cosmopolitan	362	0.29	0.09	0.11	0.05
National Geographic	160	0.19	0.03	0.09	0.03
Time	94	0.18	0.06	0.23	0.16
Newsweek	64	0.18	0.06	0.25	0.29
IEEE Spectrum	106	0.08	0.05	0.26	0.27
The New Yorker	96	0.06	0.04	0.07	0.01
IEEE Computer	132	0.02	0.04	0.08	0.03

3.6 Document statistics

For the four complexity measures, examples of increasing complexity are presented in Figs. 3–6.[3]

To get an insight into the overall distribution of documents in the dataset, Table 1 gives the four complexity values averaged over all documents in the UvA dataset.

Furthermore, Table 2 shows the histogram of number of layers per page. Thus, in the UvA dataset 61% of the pages have two layers, 10% have three layers and 1.5% have four layers. The remaining 27% are "simple" pages with only one layer.

4 Evaluation measures

Complexity measures give an indication of the expected difficulty of a task based on the data prior to the use of an algorithm. Evaluation measures are needed to compare different algorithms performing the task.

[3] Note that here we show only images from IEEE Computer, issue February 2001, as the other editors did not grant us permission to publish pages from their magazines. Please see the complete UvA-CDD dataset available on our Web site for more complex examples.

Table 2. Distribution of number of layers per page in UvA dataset

Number of layers/page	*Number of pages (%)*
1	0.274162
2	0.610454
3	0.100592
4	0.014793

4.1 Precision and recall

Using the graph-based document model, evaluation measures can be posed as a graph matching problem between a ground truth graph and the detected graph.

The decomposition of the problem into the four subtasks leads to an important simplification as in each step either vertices or edges are used.

For each task, two major aspects of a specific algorithm should be evaluated. First, is the result correct – are these indeed elements the system was supposed to find? Second, is the result complete – have any elements been missed?

Precision and recall are well known in information retrieval [1] to be indicators of these two often conflicting factors. They are used explicitly [8] or implicitly [10, 11] in the evaluation of document analysis tasks. Let us first

consider the general definition. Let S be a set of ground truth elements and S' be the result of any task aiming at deriving the ground truth elements. Then precision and recall are given by:

$$p = \frac{\mid S' \cap S \mid}{\mid S' \mid} \qquad r = \frac{\mid S' \cap S \mid}{\mid S \mid}. \tag{10}$$

Obviously, precision and recall are always in the range [0,1]. Maximum precision is achieved when all the elements in the detected set are indeed part of the ground truth set. Or, in other words, there are no false alarms detected. The maximum value for recall is reached when all the elements in the ground truth set are also present in the detected set, i.e., no false negatives.

When results are not discrete sets, but correspond to regions in the image, the same definitions can be used by using the area of the regions instead of counting the number of elements in a set.

To identify how elements contributed to the precision and recall measures, we can derive the following sets:

- *Correct* $= S \cap S'$
- *Misdetection* $= S \backslash S'$
- *False alarm* $= S' \backslash S$

In the following discussion, the sets S and S' will be made specific for the evaluation of the different tasks.

4.2 Page segmentation

For the evaluation of page segmentation we are faced with the problem that there is no one-to-one correspondence defined between the areas found by the algorithm and the areas given in the ground truth. The same problem was encountered in the evaluation of segmentation of a page into text lines by Liang et al. [10,11]. We base our measures on the method proposed in [10, 11] and extended by Mao and Kanungo in [12]. It is straightforward to use the definitions for the more general objects we consider.

So let us make this more precise. The ground truth objects are given by $\mathcal{O}_g$. Let the result of the page segmentation be given by $\mathcal{O}'_g$. To find the likelihood of a match between elements in the two sets, we consider the pairwise precision and recall between the object with index i in $\mathcal{O}'_g$ and the object with index j in $\mathcal{O}_g$ as follows:

$$p_1^{ij} = \frac{Area(\hat{o}'_i \cap \hat{o}_j)}{Area(\hat{o}'_i)} \qquad r_1^{ij} = \frac{Area(\hat{o}'_i \cap \hat{o}_j)}{Area(\hat{o}_j)}. \tag{11}$$

Based on the analysis of the values for all possible pairs, Liang et al. introduced six categories to measure the quality of detection. The first three are similar to those we encountered, but the imprecision of the match between two objects is taken into account.

To identify the correctly detected elements, let us define the approximate intersection $X \tilde{\cap} Y$, which gives the pairwise area intersection of all elements for which $r_1^{ij} \approx 1$ and $p_1^{ij} \approx 1$.

Further categories are:

- *Misdetection* if for all $j : r_1^{ij} \approx 0$.
- *False alarm* if for all $i : p_1^{ij} \approx 0$.

In addition, more sets are identified to give the category of error:

- *Split* if for all $j : r_1^{ij} < 1$ and $\sum_{j=1}^{N} r_1^{ij} \approx 1$.
- *Merge* if for all $j : p_1^{ij} < 1$ for all i and $\sum_{i=1}^{M} p_1^{ij} \approx 1$.
- *Spurious* for any other detection.

Note that the above definition requires two thresholds T_l and T_h to judge whether values are close to 0 or 1, respectively. The actual values for these two thresholds were selected by analyzing the p_1^{ij} and r_1^{ij} matrices, for seven randomly selected pages from each of the magazines in the dataset, groundtruthed twice. We found $T_h = 0.80$ and $T_l = 0.05$ to be the appropriate threshold values for the UvA dataset.

The above-described measures give accurate local information. The definitions of global precision and recall for a page are:

$$p_l = \frac{Area(\hat{O}_g \tilde{\cap} \hat{O}'_g)}{Area(\hat{O}'_g)} \qquad r_l = \frac{Area(\hat{O}_g \tilde{\cap} \hat{O}'_g)}{Area(\hat{O}_g)}. \tag{12}$$

After this task we assume that we have found the match between $\mathcal{O}$ and $\mathcal{O}'$ defined by the pairs of elements in the two sets for which $r_1^{ij} \approx 1$ and $p_1^{ij} \approx 1$. The objects in the matched graphs will be denoted by $\tilde{O}$ and $\tilde{O}'$, respectively. Likewise, relations between those objects in the result and the ground truth are indicated by $\tilde{R}_g$ and $\tilde{R}'_g$. Further evaluation is restricted to those two object sets and relations to assure that errors made in the page segmentation do not propagate into further evaluation. Note, however, that in some cases it might be better to apply subsequent steps of the algorithm to the ground truth data from the previous step.

4.3 Evaluation of layout detection

In the layout detection for color documents, what needs to be evaluated is whether the geometric relations between document objects are found correctly. In our case this corresponds to evaluating whether the edges corresponding to pairs in the overlap relation $\mathcal{R}_g$ are correct.

Following the notation just introduced, this gives the following precision and recall measures for step 2 of the analysis process:

$$p_2 = \frac{|\tilde{R}'_g \cap \mathcal{R}_g|}{|\tilde{R}'_g|} \qquad r_2 = \frac{|\tilde{R}'_g \cap \mathcal{R}_g|}{|\tilde{R}_g|}. \tag{13}$$

4.4 Evaluation of logical object classification

To evaluate the classification of objects into logical classes, we must find the objects in both the ground

truth and the results with a specific label. Respectively we define:

$$\tilde{O}_l^i = \{o \in \tilde{O} | \text{logical label}(o) = i\}\,,$$
$$\tilde{O}_{l'}^i = \{o \in \tilde{O}' | \text{logical label}(o) = i\}\,.$$

Furthermore, we need the intersection m of the objects in the result and the ground truth according to the labels:

$m_{ij} = \tilde{O}_l^i \cap \tilde{O}_{l'}^i$.

By considering the cardinality of each m_{ij} we get the well-known confusion matrix for classification.

To evaluate the performance on the whole page, we need to identify the set of objects M that were classified correctly, i.e., all elements in m_{ii}. This leads to the following overall measures:

$$p_3 = \frac{|\bigcup_i(\tilde{O}_l^i \cap \tilde{O}_{l'}^i)|}{|\bigcup_i(\tilde{O}_{l'}^i)|} \qquad r_3 = \frac{|\bigcup_i(\tilde{O}_l^i \cap \tilde{O}_{l'}^i)|}{|\bigcup_i(\tilde{O}_l^i)|}\,. \quad (14)$$

Note that in our case $p_3 = r_3$ as we only consider those elements that were matched previously. Hence, the two object sets have the same cardinality.

4.5 Evaluation of reading order detection

Evaluation of the final step in the analysis is similar to the layout detection as both are directly computed from the match between the edges of the graph. Again to avoid error propagation, only those elements that received the correct label in the previous step are considered when matching the edges in the logical graph. Following the same notation conventions as earlier the relations between those objects in the result and the ground truth are indicated by $\tilde{R}_l$ and $\tilde{R}_{l'}$, respectively.

So the final evaluation measures are given by:

$$p_4 = \frac{|\tilde{R}_{l'} \cap \tilde{R}_l|}{|\tilde{R}_{l'}|} \qquad r_4 = \frac{|\tilde{R}_{l'} \cap \tilde{R}_l|}{|\tilde{R}_l|}\,. \quad (15)$$

5 Implementation

Groundtruthing a complex color document is a difficult task first because of the many relations between the different objects, and second because some subjective choices have to be made. We have therefore defined a set of rules the groundtruther has to obey.

Even when these guidelines are strictly obeyed there will always be a variation between different evaluators as the boundary of an object has to be indicated manually.

5.1 Guidelines for ground truth creation

As there are many geometric relations between document objects, it is more convenient to use layers to define the geometric structure. Later in the process the relations defining the geometric structure can be derived easily from the layer-based definition.

The rules for geometric description are as follows:

- *Rule g1:* Put overlapping objects in different layers.
- *Rule g2:* Put objects having different background in different layers.
- *Rule g3:* If objects do overlap, specify that the top one is on a higher layer.

The easiest way to make sure that the above holds is to start with all objects that are fully visible, i.e., their perceived shape is the same as their real shape. These form the top layer. From there continue downwards.

- *Rule g4:* Prefer regular shapes over polygonal shapes, i.e., whenever possible, use rectangles or ellipses. If the use of regular shapes would produce a false overlap, use a polygon to indicate the shape.
- *Rule g5:* Specify the "background color" for textual document objects placed on images, based on local rather than global information, i.e., if the text falls in a uniform part of a picture, consider the background to be uniform.
- *Rule g6:* Mark tables as a whole, not as independent cells and lines.

Finally, the set of rules for logical groundtruthing are:

- *Rule l1:* Assign logical labels based on visual appearance only, without considering the content.
- *Rule l2:* If two zones have a different background, consider them independent in the reading order.
- *Rule l3:* Link objects in one reading order iff they are in the same layer.

5.2 Variability

To measure the inherent variability in the ground truth definition, we perform a variability test. From each magazine we select 4 document pages for each of the four complexity classes, thus 16 document pages in total. For each complexity class, we select randomly document pages of lowest, highest, and two other intermediary complexities, respectively. They are groundtruthed 4 times in total by two different evaluators.

The four ground truth files obtained by the four evaluation runs are evaluated in pairs, each of them playing the role of ground truth and result, respectively. We use the same evaluation measures as before for each step to compute the variability.

The evaluation results for all six possible pairs are averaged to obtain the variability measure. This is expressed as average value.

Table 3 summarizes the observed variability in ground truth specification.

If the operators in the variability experiment follow carefully the guidelines for ground truth specification, there should be no variation between their ground truth definitions. As shown in Table 3, the variability error is quite small. As expected, the largest variability errors are reported for document object specification – p_1/r_1. This is due to human imprecision in specification of the document objects' boundaries. The smallest variability is reported for reading order specification – p_4/r_4. For

Table 3. Variability in ground truth definition for the UvA color document dataset

Magazine	p_1/r_1	p_2/r_2	p_3/r_3	p_4/r_4
Cosmopolitan	0.98/0.99	0.92/1.00	1.00/1.00	1.00/1.00
IEEE Computer	0.97/0.94	0.96/0.96	1.00/1.00	1.00/1.00
IEEE Spectrum	0.99/0.99	0.92/0.94	0.99/0.99	0.96/0.98
National Geographic	0.97/0.93	1.00/1.00	1.00/1.00	0.99/1.00
Newsweek	0.97/0.98	0.96/0.94	0.99/0.99	1.00/1.00
The New Yorker	0.99/0.92	0.95/0.95	0.97/0.97	1.00/1.00
Time	0.97/0.90	0.94/0.93	0.98/0.98	0.96/1.00

several human observers it is far easier to indicate the correct reading order consistently than to click on the same boundary points.

We conclude from Table 3 that the ground truth in UvA-CDD is reproducible up to 97–99% depending on the task.

5.3 GT-UvA – The ground truth editor

Following the guidelines defined in Sect. 5.1, the ground truth is manually generated for every page in the UvA-CDD dataset using the **GT-UvA** ground truth editor software.

All the pages are processed in two steps. First they are groundtruthed by a student, then they are checked by the author. The author corrects wrongly assigned labels or features and reshapes the contours of document objects in case of visually estimated significant error.

The **GT-UvA** software is implemented in VisualC++ using MFC and Visual SDK [13] classes. The user interface allows the user to draw a rectangular, circular, elliptical, or polygonal shape around the document objects. The layout and logical descriptions are then introduced via a property dialog box.

The ground truth can be exported in plain ASCII or in XML format. Figure 8 shows the document type definition for the UvA color document dataset, where all the possible document objects are defined. For visualization of the geometric and logical descriptions, we store the ground truth in SVG format. A screenshot of the application is shown in Fig. 7.

5.4 Eval – The evaluation toolkit

The evaluation measures described in Sect. 4 are implemented as a program in C, called **Eval**, to be run in batch mode. **Eval** has two operating modes: page evaluation and dataset evaluation. In page evaluation mode, **Eval** takes as arguments two text files, one containing the ground truth information, the other the result description of a document page. In dataset evaluation mode, the input argument is the directory where the dataset is located. For this case, evaluation is performed for each individual page. Statistics are generated at the end for the entire dataset.

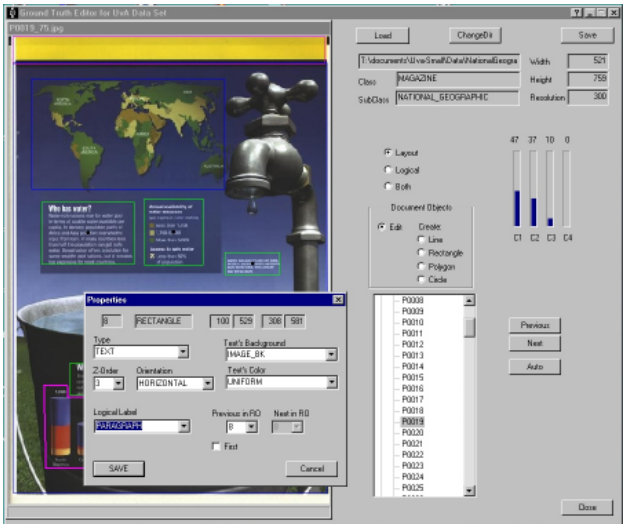

Fig. 7. User interface of the application used for ground truth generation and visualization

6 Conclusion

To advance the field of color document analysis a well-defined dataset is essential. We have created the UvA color document dataset consisting of over 1000 document pages, groundtruthed at the geometric and logical levels.

To describe the document pages, a graph-based model is proposed. Based on the model, the process of document analysis has been decomposed into four steps dealing with the vertices or edges of either the geometric graph or the logical graph describing the document.

As the variety of color documents ranges from very simple to complicated structures, we have defined four complexity measures that rank the document complexity for each of the four steps independent of the algorithm used for analysis.

For each of the four steps, evaluation measures are defined. All of the measures are derived from the general evaluation measures precision and recall. The complexity and the evaluation measures are scale independent. They are also independent of the textual content of the document.

From our variability experiment we conclude that the ground truth in UvA-CDD is valuable up to 97–99% depending on task reproducibility.

```
<!-- DTD file for the UvA  dataset -->
<!ELEMENT uvadoc (doc-descr, page+) >
<!ELEMENT page (pag_descr, (text-do|image-do|graphics-do)+ ) >
<!ELEMENT doc-descr (id, name, xres, yres, width, height)>
<!ELEMENT pag-descr (id, pag-nr)>
<!ELEMENT text-do (advertisement|author|abstract|bibliography|
caption|header|footer|foot-note|list|note|page-number|
paragraph|table|title|other_txt) >
<!ELEMENT image-do (advertisement|image-containing-scene-text|
regular-image|other_img) >
<!ELEMENT graphics-do (border|barcode|graph|logo|map|separator|
other_graph)>


<!ATTLIST text-do
    id ID #REQUIRED
    shape-type ENTITY #REQUIRED
    position ENTITY #REQUIRED
    orientation ENTITY #REQUIRED
    layer CDATA #REQUIRED
    bkgnd-color ENTITY #REQUIRED
    txt-color ENTITY #REQUIRED
    perceived-shape ENTITY #REQUIRED
    real-shape ENTITY #REQUIRED
    overlap-list ENTITY #REQUIRED
    prev-ro CDATA #REQUIRED
    next-ro CDATA #REQUIRED
>

<!ATTLIST image-do
    id ID #REQUIRED
    shape-type ENTITY #REQUIRED
    position ENTITY #REQUIRED
    orientation ENTITY #REQUIRED
    layer CDATA #REQUIRED
    perceived-shape ENTITY #REQUIRED
    real-shape ENTITY #REQUIRED
    overlap-list ENTITY #REQUIRED
>


<!ELEMENT shape-type (line|rectangle|ellipse|polygon)>
<!ELEMENT line (#PCDATA)>
<!ELEMENT rectangle (#PCDATA)>
<!ELEMENT ellipse (#PCDATA)>
<!ELEMENT polygon (#PCDATA)>
<!ELEMENT position   (x1, y1, x2, y2) >
<!ELEMENT x1 (#PCDATA)>
<!ELEMENT y1 (#PCDATA)>
<!ELEMENT x2 (#PCDATA)>
<!ELEMENT y2 (#PCDATA)>
<!ELEMENT orientation (horizontal|vertical|other) >
<!ELEMENT layer (#PCDATA)>
<!ELEMENT bkgnd-color (uniform|image|other)>
<!ELEMENT txt-color (uniform|image|other)>
<!ELEMENT prev-ro (#PCDATA)>
<!ELEMENT next-ro (#PCDATA)>

<!ELEMENT abstract  (#PCDATA)>
<!ELEMENT author    (#PCDATA)>
<!ELEMENT body      (#PCDATA)>
<!ELEMENT caption   (#PCDATA)>
<!ELEMENT list      (#PCDATA)>
<!ELEMENT other     (#PCDATA)>
<!ELEMENT page-number   (#PCDATA)>
<!ELEMENT title     (#PCDATA)>
```

Fig. 8. The document type definition used for the UvA dataset

Finally, the documents and associated tools are available on a restricted basis to the research community via a special Web site.

Acknowledgements. Leon Todoran was supported by Senter den Haag and Océ Technologies BV, Venlo (IOP project IBV 96008). The authors would like to thank Andrew Bagdanov and Marco Aiello for their comments on the manuscript.

References

1. Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison-Wesley, Reading, MA
2. Bottou L, Haffner P, Howard PG, LeCun Y (1999) Djvu: analyzing and compressing scanned documents for internet distribution. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 625–628
3. Chen WY, Chen SY (1998) Adaptive page segmentation for color technical journals' cover images. Image Vis Comput 16(3):855–877
4. Garcia C, Apostolidis X (2000) Text detection and segmentation in complex color images. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Istanbul, pp 75–78
5. Hase H, Shinokawa T, Yoneda M, Sakai M, Maruyama H (1999) Character string extraction from a color document. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 75–78
6. Hua XS, Wenyin L, Zhang HJ (2001) Automatic performance evaluation for video text detection. In: Proceedings of the 6th international conference on document analysis and recognition (ICDAR'01), Seattle, pp 545–550
7. Jain AK, Yu B (1998) Automatic text location in images and video frames. Pattern Recog 31(12):2055–2076
8. Junker M, Hoch R, Dengel A (1999) On the evaluation of document analysis components by recall, precision and accuracy. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September, 1999, pp 713–716
9. Kanai J, Rice SV, Nartker TA, Nagy G (1995) Automated evaluation of ocr zoning. IEEE Trans Pattern Anal Mach Intell 17(1):86–90
10. Liang J, Phillips IT, Haralick R (2001) An optimization methodology for document structure extraction on latin character documents. IEEE Trans Pattern Anal Mach Intell 23(7):719–734
11. Liang J, Rogers R, Haralick R, Phillips I (1997) Uw-isl document image analysis toolbox: an experimental environment. In: Proceedings of the 4th international conference on document analysis and recognition, Ulm, Germany, August 1997, pp 984–988
12. Mao S, Kanungo T (2001) Empirical performance evaluation methodology and its application to page segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 23(3):242–256
13. Microsoft Research (2000) The Microsoft Vision SDK library. http://www.research.microsoft.com/projects/VisSDK
14. Nagy G (2000) Twenty years of document image analysis in PAMI. IEEE Trans Pattern Anal Mach Intell 22(1):38–62
15. Perroud T, Sobottka K, Bunke H, Hall L (2001) Text extraction from color documents – clustering approaches in three and four dimensions. In: Proceedings of the 6th international conference on document analysis and recognition (ICDAR'01), Seattle, pp 937–941
16. Ryu DS, Kang SM, Lee SW (2000) Parameter-independent geometric document layout analysis. In:

Proceedings of the 2000 international conference on pattern recognition (ICPR'00), Barcelona, Spain, pp 397–400
17. Sauvola J, Kauniskangas H (1998) MediaTeam document database II. CD-ROM collection of document images, University of Oulu, Finland. http://www.mediateam.oulu.fi/MTDB/index.html
18. Sauvola J, Haapakoski S, Kauniskangas H, Seppanen T, Pietiklainen M, Doermann D (1997) A distributed management system for testing document image analysis algorithms. In: Proceedings of the 4th international conference on document analysis and recognition (ICDAR'97), Ulm, Germany, pp 989–995
19. Sobottka K, Bunke H, Kronenberg H (1999) Identification of text on colored book and journal covers. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), September 1999, Bangalore, India, pp 57–60
20. Todoran L, Aiello M, Monz C, Worring M (2001) Logical structure detection for hetrogeneous document classes. In: Kantor PB, Lopresti DP, Zhou J (eds) Proceedings of SPIE, Document Recognition and Retrieval VIII, San Jose, CA, 3407:99–111
21. Tsujimoto S, Asada H (1992) Major components of a complete text reading system. Proc IEEE 80(7):1133–1149
22. Wallace GK (1991) The JPEG still picture compression standard. Commun ACM 34(4):30–44
23. Watanabe T, Sobue T (2000) Layout analysis of complex documents. In: Proceedings of the 2000 international conference on pattern recognition (ICPR'00), Barcelona, Spain, pp 447–450
24. Wu V, Manmatha R, Riseman EM (1999) Textfinder: an automatic system to detect and recognize text in images. IEEE Trans Pattern Anal Mach Intell 21(11):1224–1229
25. Yanikoglu B, Vincent L (1997) Pink panther: a complete environment for ground-truthing and benchmarking document page segmentation. Pattern Recog Lett 31(9):1191–1204
26. Zhong Y, Karu K, Jain AK (1995) Locating text in complex color images. Pattern Recog 28(10):1523–1535

Leon Todoran received his M.Sc. in computer science from the Technical University, Cluj-Napoca, Romania in 1993. Between 1993 and 1997 he worked as a teaching assistant at the Technical University Cluj-Napoca. He is now a Ph.D. student at the University of Amsterdam, The Netherlands. His research interests include image processing, document analysis, and real-time video analysis.

Marcel Worring received a degree in computer science (honors) from the Free University Amsterdam. He earned his Ph.D. from the University of Amsterdam, The Netherlands in 1993, and his Ph.D. thesis was on digital image analysis. He became assistant professor in 1995 and associate professor in 2003 in the Intelligent Sensory Information Systems group at the University of Amsterdam. His current interests are in multimedia information analysis, in particular document and video analysis. He has been a visiting researcher in the Department of Diagnostic Imaging at Yale University (1992) and at the Visual Computing Lab at the University of California, San Diego (1998).

Arnold W.M. Smeulders has been in image processing since 1975 when he received his M.Sc. in physics. He received his Ph.D. in medical image analysis in 1982. He is currently a full professor of multimedia. He heads the 25-person Intelligent Sensory Information Systems group (ISIS) working on the theory of computer vision, image retrieval, and industrial vision. He has published extensively on vision and recognition. His current topics are: image retrieval, color, intelligent interaction, the role of engineering in vision, and the relation between language and vision. Previously, he served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and currently serves as an associate editor of Cytometry and BioImaging. He is a senior member of the IEEE.