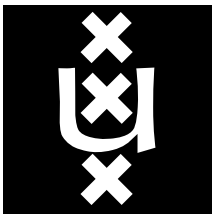


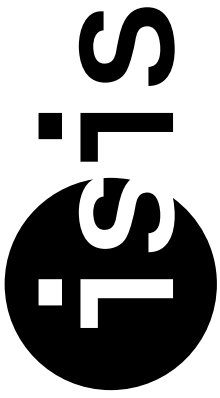
ISIS technical report series, Vol. 2002-01, February 2002

The UvA Color Document Dataset

Leon Todoran , Marcel Worring and Arnold W.M. Smeulders
Intelligent Sensory Information Systems
Department of Computer Science
University of Amsterdam
The Netherlands

Publications on color document image analysis present results on small, non-publicly available datasets. We propose in this paper a well defined and groundtruthed color dataset existing of over 1000 pages, with associated tools for evaluation. 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.

Submitted to IJDAR (International Journal on Document Analysis and Recognition)



Contents

Intelligent Sensory Information Systems
Department of Computer Science
University of Amsterdam
Kruislaan 403
1098 SJ Amsterdam
The Netherlands

tel: +31 20 525 7463
fax: +31 20 525 7490
http://www.science.uva.nl/research/isis

Corresponding author:
Leon Todoran
tel: +31(20)525 7555
todoran@science.uva.nl
http://www.science.uva.nl/~todoran

1 Introduction

Color is now playing an important role in publishing everything from scientific journals, newspapers, 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. Some tools for color documents as color OCR [18, 4, 21], color document compression [2], and color string localization [13, 25, 3, 5, 7] 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 data sets like the UW[10] and MTDB [17], freeing researchers from the labor intensive task of creating datasets to work on. Secondly, the availability of standard evaluation tools for OCR and page segmentation [11], [24], [16] allowing knowledge exchange between different researchers.

For color document image analysis, no such data set standardization has taken place. The MDTB data set does contain some colored 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 dataset for evaluating tools. Typically the data sets 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 which could be a first step in standardizing the evaluation of color document analysis.

The dataset consists of over 1000 pages with a ground truth describing the document components, their layout and 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, that existing OCR methods can extract the content from a text zone.

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, Zhong et.al. in [26] 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 is unknown”. 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 that data.

The existing evaluation methods for layout analysis can be grouped into two main categories: text-based and region-based evaluation. Text-based evaluation [8] uses textual ground truth and the edit distance to measure the errors in layout detection. Region-based evaluation methods [24, 10, 9, 11] compare the outline of the detected zones with the zone description in the ground truth.

For evaluating document analysis algorithms for color documents the region based methods are most suited as they can easily be applied to both text, pictures, and graphics. We do, however, have to extend them first to be able to evaluate color document analysis.

This paper is organized as follows. In Section 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 Section 4. Finally, Section 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 will 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 evaluation of color document analysis should be created following some guidelines. Firstly, to cover different applications, the dataset must be comprised of document pages of varying style and complexity. Secondly, 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, we have scanned full issues of the 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 one thousand 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 300dpi with 24 bits color information per pixel. In uncompressed TIFF format this requires a total space of 23.3 GB. We also have 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] preserving image quality while providing fair compression. In this format the dataset totals 1.1 Gb.

The dataset set is made available via a website[1]. 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 objects 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 the 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 representation are too limited to describe such complex relations, hence a graph-based representation is to be used.

A single graph cannot describe all possible spatial relations between the document objects. Separate graphs are used to describe relations like overlap and inclu-

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

sion. Thus the layout structure is given by a set of graphs where the vertices are the document objects $\mathcal{O}_g$ and the edges $\mathcal{R}_g$ denote a relation between the objects. The graphs can be directed or undirected and can have weights to encode attributes of the edges. As the vertices are the same for every graph, the layout structure is defined as follows:

$$\mathcal{G} = \langle \mathcal{O}_g, \mathcal{R}_g^1, \mathcal{R}_g^2, ... \rangle.$$

Similarly the logical structure is defined as:

$$\mathcal{L} = \langle \mathcal{O}_l, \mathcal{R}_l^1, \mathcal{R}_l^2, ... \rangle.$$

Although logical structure (and sometimes layout) can span different pages, 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:

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

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 documents objects namely text, image, and graphics.

In the description of the outline of these objects we make a distinction between the *shape* and the *region* of a document object. With shape we denote the perceived shape of the object which in a layered document could be partly obscured by another document object. The object region is the true shape of the object. In the following, the object itself will be indicated as o, the shape of the object as $\overline{o}$ and the region of the object as $\hat{o}$. In a similar way $\hat{O}$ where O is a set of objects denotes the regions of all objects in the set.

Now considering the text objects, recall from the introduction that we focus on properties of the document which are specific for 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};
  - shape
    - line: end-points
    - rectangle: top-left, bottom-down corners;
    - polygon: list of points;
    - ellipse: x,y-position and size of short and long axis;
  - object region: 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 }

For later use, let us define notations for the following subsets of geometric document objects based on individual category and one mixed class for pictorial information:
$T = \{o \in \mathcal{O}_g | category(o) = text\}$
$G = \{o \in \mathcal{O}_g | category(o) = graphics\}$
$I = \{o \in \mathcal{O}_g | category(o) = image\}$
$P = \mathbf{G} \cup \mathbf{I}$

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 notation 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 textobject with foreground type t and background type b. Choices for f and b are uniform (u), non-uniform ($\neg u$), graphic (g), image (i), or nothing (), the latter indicating that the foreground or background can be any of the given types. So as an example, $\mathcal{T}_{\neg u}$ is the set of non-uniform textstrings 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:

$$o_1 >_t o_2 <=> \overline{o}_1 \cap \overline{o}_2 \cap \hat{o}_1 \neq \emptyset$$

The above is applicable both when the shapes 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 $<_w$ which indicates that the shape of one object is fully contained within the area of the other.

$$o_1 <_w o_2 <=> \overline{o}_1 \subset \hat{o}_2$$

On the basis of the above we define two layout structure relations, the first dealing with overlapping shapes of objects, the other with the remaining on-top relations.

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

$$\mathcal{R}_g^w = \{(o_1, o_2) \in \mathcal{O}_g | \overline{o}_1 >_t \overline{o}_2 \wedge o_1 >_w o_2\}$$

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

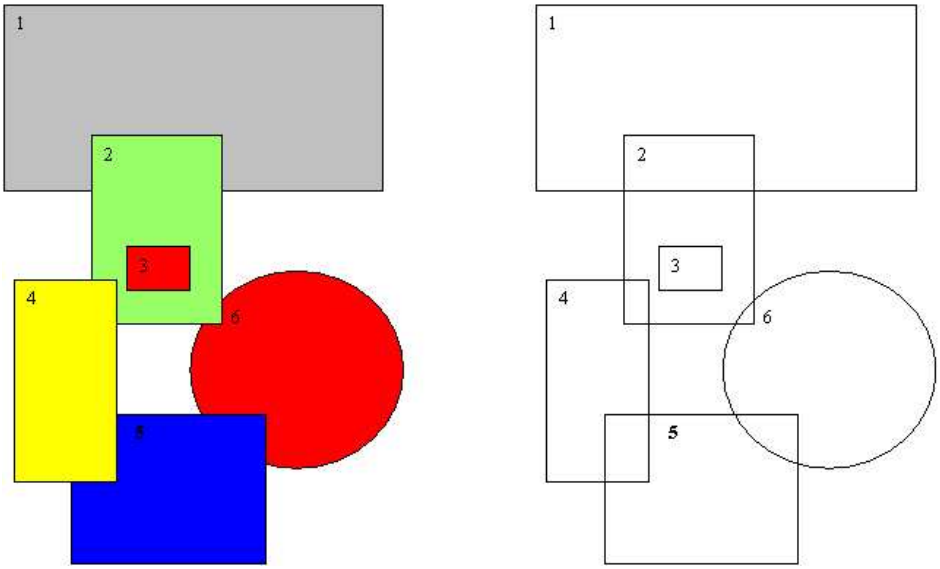


Figure 1: In the left figure a document consisting of 6 document objects is depicted, where for each object the area is shown. In the right figure the same objects are shown, but now their perceived shape is drawn. Note that for example $o_2 >_t o_1$ and $o_3 <_w o_2$

The two relations are explained in Figure.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 "paper layer", is the lowest in the layer hierarchy. A document object $o \in \mathcal{O}_g$ is assigned to the layer of 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 relation, no consistent layer definition exists. So we restrict ourselves to documents in which there are no cycles in the graph.

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 were selected. Object classes which are not frequently appearing in the dataset receive the label "**Other**". Of course these could be refined later. It leads to:

- logical attributes;
  - category: {text, image, graphics};
  - logical label
    - text: {Author, Abstract, Bibliography, Caption, Equation, Header, Footer, Foot Note, 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.

The reading order is based on the relation *before in reading* denoted by $<_r$. 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 $o_1 <_r o_2$. To be a proper reading order graph it should be a-cyclic. Then, any path in the graph is an independent reading order in the document. If there are multiple paths in the graph they are related to groups of document objects which can be read in arbitrary order. So for logical structure we have:

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

3 Document Complexity

The performance of an algorithm on a given dataset depends on two things. Namely the quality of the algorithm itself and the complexity of the data. This complexity of the data is task dependent. When a 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 step deal with the logical content of the document:

- *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.

[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.

- *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 object. Finally we have the step of

- *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 graph are computed.

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

- C_1 - Complexity for page segmentation;
- C_2 - Complexity for layout detection;
- C_3 - Complexity for logical object classification;
- C_4 - Complexity for 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.

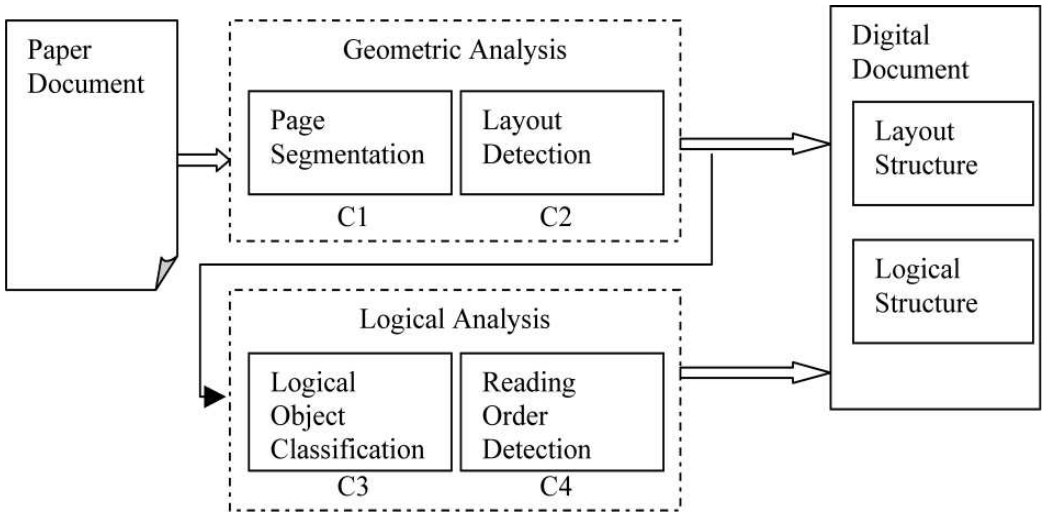


Figure 2: The four main tasks in color document image analysis, with their associated complexities.

3.2 Document complexity for page segmentation

Analyzing the difficulties of the page segmentation algorithms described in literature [12, 15, 23], we identified four main factors that influence the quality of the results. These factors are:

1. *non-uniformity in color*; if the color of a textstring is non-uniform or it is 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 apply to this general style they are more difficult to segment.

3. *picture/text ratio*; pictures contain a much wider range of colors than most textstrings, and 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 textstrings in the document.

Taking into account the above, we consider a document page containing only uniformly colored text objects, having rectangular shapes, on a uniform background to have complexity 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 non-uniform color and irregularly shaped boundaries placed on top of it. For each of the four factors we have designed a complexity measure which is normalized to the range [0,1].

The first measure is based on the textstrings that are either not uniformly colored or have a non-uniform background. Using |.| to denote the cardinality of a set and using the shorthands from section 2.3:

$$c_1^1 = \frac{|\mathcal{T}_u^g| + |\mathcal{T}_u^i| + |\mathcal{T}_{\neg u}|}{|\mathcal{T}|} \tag{1}$$

The second measure considers the percentage of non-regular shapes:

$$c_1^2 = \frac{|\mathcal{O}_g^P| + |\mathcal{O}_g^E|}{|\mathcal{O}_g|} \tag{2}$$

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:

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

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

$$c_1^4 = \frac{|P^{ct}|}{|P|} \tag{4}$$

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 or another component. Here, we consider them equally important.

$$C_1 = \frac{c_1^1 + c_1^2 + c_1^3 + c_2^4}{4} \tag{5}$$

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] can 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 section 2.3 the geometric structure is based on the observation that we perceive a regularly shaped objects as the full object even if it is partly occluded. Clearly the larger the occlusion the less clear this observation can be made. Therefore, to measure the complexity of the decision 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.

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

It follows that for layout structure detection a document has complexity 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 shapes.

3.4 Document complexity for logical object classification

In general, logical object classification is based on layout features (visual appearance), content, and possible apriori information about the document class. As indicated earlier, we do not consider the content of the document. Furthermore, 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, the variability and separability depend on the geometric features used and 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 rather. We do so separately for text, images, and graphics so that they could 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:

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

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 at least appearing at one place in the document.

3.5 Document complexity for reading order detection

Analyzing existing methods for reading order detection [20, 19], 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 the other in a non-regular way. To that end we will derive a complexity measure that measures the irregularity of the reading path when it is visiting the different text objects in the document.

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

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

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 which 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 centres of gravity of the subsequent document objects in r_i. Now 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 would always be the case we assign a complexity 0. In general cases the point will be 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. So locally the complexity is maximal if one has to search in exactly the opposite direction where one came from. The turning angle can be computed using the innerproduct as:

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

Note that this is not defined for 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:

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

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.

For the four complexity measures, examples of increasing complexity are presented in Figure 3-6.

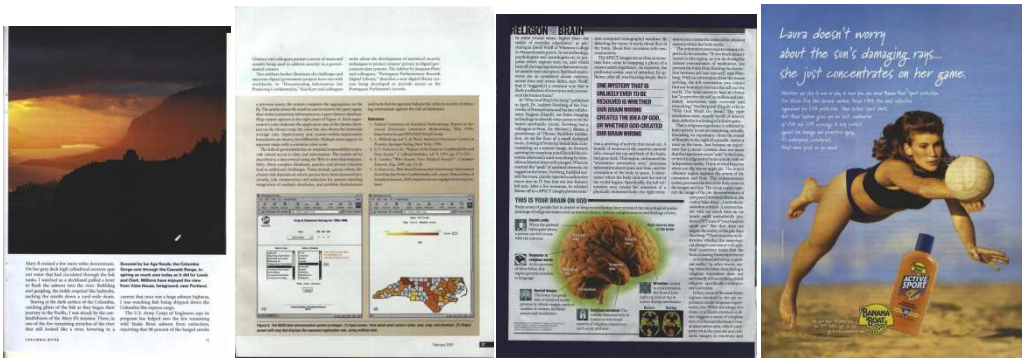


Figure 3: Images of increasing complexity for page segmentation ($C_1 = 0.0, 0.07, 0.66, 0.74$), ranging from a simple page containing only text with a uniform background and an image, to a page with an image as background with text zones on top, each of which has a polygonal shape.

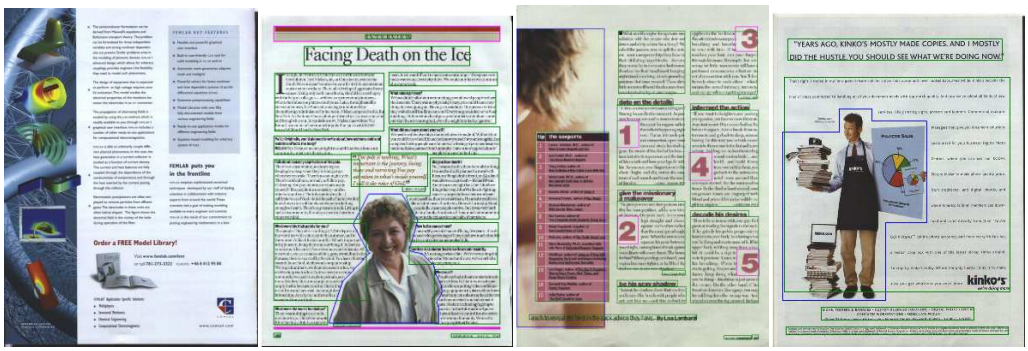


Figure 4: Examples of documents with increasing complexity for layout detection ($C_2 = 0.0, 0.02, 0.12, 0.42$). The simplest examples has document objects which all have a rectangular outline which is fully visible. In the most complex example the occluded area is a significant part of the occluded object.

To get an insight in the overall distribution of documents in the dataset, Table 1 gives the four complexity values averaged for each document of the UvA Data Set.

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.

Figure 5: Different documents with increasing complexity ($C_3 = 0.03, 0.17, 0.27, 0.48.$) for logical classification. The first document has 1 logical label only, whereas the last document has 12 different labels.

Figure 6: Documents with increasing complexity for reading order detection ($C_4 = 0.0, 0.20, 0.54, 0.91.$). Paths clearly range from regular to very irregular.

Table 1: The average complexity values for UvA Color Document Dataset.

Magazine	*Pages*	C_1	C_2	C_3	C_4
Cosmopolitan	362	0.29	0.09	0.11	0.05
Time	94	0.22	0.22	0.24	0.16
NewsWeek	64	0.22	0.20	0.25	0.29
National Geographic	160	0.20	0.04	0.09	0.02
IEEE Spectrum	106	0.10	0.15	0.26	0.27
The NewYorker	96	0.08	0.04	0.07	0.01
IEEE Computer	132	0.02	0.03	0.07	0.01

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 graph detected.

The decomposition of the problem into the four subtasks leads to and 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. Firstly, is the result correct, are these indeed elements the system was supposed to find? Secondly, is the result complete, aren't there elements missed?

Precision and recall are well known in Information Retrieval [1] to be indicators of these two often conflicting factors. So let us first consider the general definition. Let S be a set of ground truth elements and let S' be the result of any task aiming at deriving the ground truth elements. Then precision and recall are given by:

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

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:

- $Correct = S \cap S'$
- $Misdetection = S \backslash S'$
- $FalseAlarm = S' \backslash S$

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

4.2 Page Segmentation

For 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 evaluation of segmentation of a page into text lines by Liang et al. [10, 9]. We base our measures on the method proposed in the reference and extended by Mao and Kanungo in [11]. It is straightforward to use the definitions for the more general objects we consider.

So let us make this more precise. Whereas 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 the ones 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 some 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* 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 a randomly selected page from the dataset, groundtruthed twice. We found $T_h = 0.80$ and $T_l = 0.05$ to be the appropriate threshold values for the UvA data set.

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 Objects Classification

To evaluate the classification of objects into logical classes we have to find the objects in both the ground truth and the results with a specific label. Respectively we define:

$$\begin{aligned} \tilde{O}^i_l &= \{o \in \tilde{O} | \text{logical label}(o) = i\} \\ \tilde{O}^i_{l'} &= \{o \in \tilde{O}' | \text{logical label}(o) = i\} \end{aligned}$$

Furthermore, we need the intersection m of the objects in the result and the ground truth according to the labels:
$m_{ij} = \tilde{O}^i_l \cap \tilde{O}^i_{l'}$.

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 which were classified correctly i.e. all elements in m_{ii}. It leads to the following overall measures:

$$p_3 = \frac{|\bigcup_i (\tilde{O}^i_l \cap \tilde{O}^i_{l'})|}{|\bigcup_i (\tilde{O}^i_{l'})|} \qquad r_3 = \frac{|\bigcup_i (\tilde{O}^i_l \cap \tilde{O}^i_{l'})|}{|\bigcup_i (\tilde{O}^i_l)|} \tag{14}$$

Note that in our case $p_3 = r_3$ as we only consider those elements which 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 the elements which 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:

$$p_4 = \frac{|\tilde{R}_{l'} \cap \tilde{R}_l|}{|\tilde{R}_{l'}|} \qquad r_4 = \frac{|\tilde{R}_{l'} \cap \tilde{R}_l|}{|\tilde{R}_l|} \tag{15}$$

5 Implementation

5.1 Guidelines for Ground Truth Creation

Groundtruthing a complex color document is a difficult task. Firstly, because of the many relations between the different objects. Secondly, some subjective choices

have to be made. We have therefore defined a set of rules the groundtruther has to obey.

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:* don't put overlapping objects in the same layer.
- *rule g2:* objects having different background cannot be put in the same layer.
- *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 which are fully visible, i.e. their shape is the same as their area. 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 polygonal shape the indicate exactly 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.
- 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:* don't link objects in the reading order if they are in different layers.

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

5.2 Variability

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

The 4 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 used before for each step to compute the variability.

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

Table 2 summarized the observed variability in ground truth specification.

Table 2: The 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 NewYorker	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

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 one can see in Table 2 the variability error is quite small. As expected, the largest variability errors are reported for p_1/r_1. This is due to the human imprecision in specification of the document objects' boundaries.

5.3 GT-UvA - The ground truth editor

The ground truth was manually generated using the **GT-UvA** ground-truth editor software. The **GT-UvA** software is implemented in VisualC++ using MFC and Visual SDK [14] 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 a plain ASCII or in XML format. In Figure 8 one can see the Document Type Definition (DTD) 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 Figure 7.

5.4 Eval - The Evaluation Toolkit

The evaluation measures described in Section 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

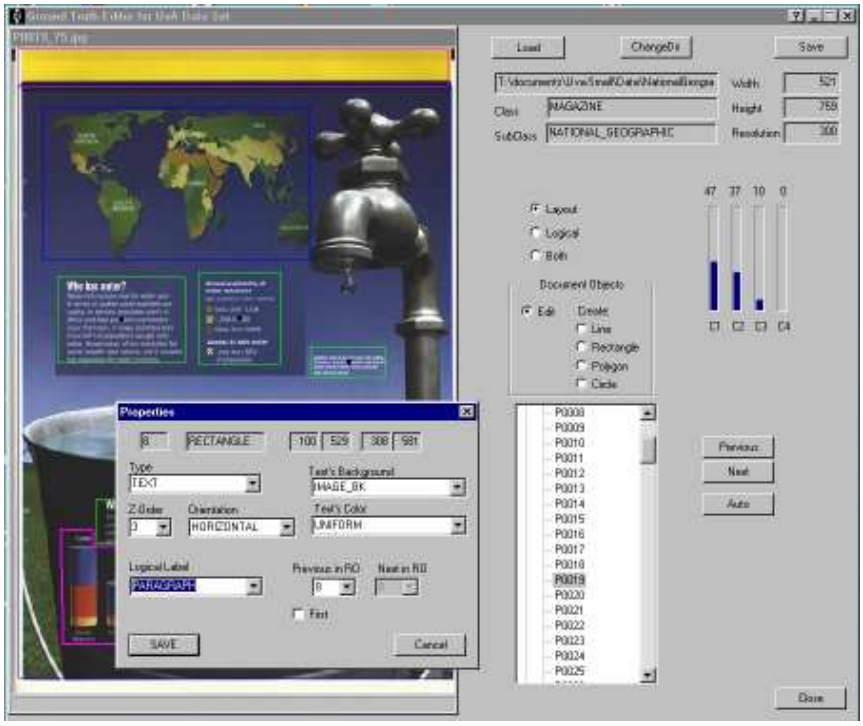


Figure 7: The user interface of the application used for ground truth generation and visualization.

the directory where the dataset is located. For this case, evaluation is performed for each individual page. At the end statistics are generated for the entire dataset.

6 Conclusion

To advance the field in color document analyis a well-defined dataset is essential. We have created the UvA color document dataset consisting of over 1000 document pages, ground truthed at the geometric and logical level.

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 which 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 them are derived from the general evaluation measures precision and recall.

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

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[2] L. Bottou, P. Haffner, P.G. Howard, and Y. LeCun. Djvu: Analyzing and compressing scanned documents for internet distribution. In *ICDAR'99, Bangalore, India*, pages 625–628, 1999.

[3] W.Y. Chen and S.Y. Chen. Adaptive page segmentation for color technical journals' cover images. *Image and Vision Computing*, 16(3):855–877, 1998.

[4] C. Garcia and X. Apostolidis. Text detection and segmentation in complex color images. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 75–78, Istambul, 2000.

[5] H. Hase, T. Shinokawa, M. Yoneda, M. Sakai, and H. Maruyama. Character string extraction from a color document. In *ICDAR'99*, pages 75–78, Bangalore, India, 1999.

[6] X.S. Hua, L. Wenyin, and H.J. Zhang. Automatic performance evaluation for video text detection. In *Proceedings of the 6th ICDAR'01*, pages 545–550, Seattle, USA, 2001.

[7] A.K. Jain and B. Yu. Automatic text location in images and video frames. *Pattern Recognition*, 31(12):2055–2076, 1998.

[8] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy. Automated evaluation of ocr zoning. *IEEE Transactions on PAMI*, 17(1):86–90, 1995.

[9] J. Liang, I.T. Phillips, and R. Haralick. An optimization methodology for document structure extraction on latin character documents. *IEEE Transactions on PAMI*, 23(7):719–734, 2001.

[10] J. Liang, R. Rogers, R. Haralick, and I. Phillips. Uw-isl document image analysis toolbox: An experimental environment. In *Proc. of the 4th International Conference on Document Analysis and Recognition, Ulm, Germany, August 1997.*, pages 984–988, 1997.

[11] S. Mao and T. Kanungo. Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Transaction on PAMI*, 23(3):242–256, 2001.

[12] G. Nagy. Twenty years of document image analysis in PAMI. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

[13] T. Perroud, K. Sobottka, H. Bunke, and L. Hall. Text extraction from color documents - clustering approaches in three and four dimensions. In *Proceedings of the 6th ICDAR'01*, pages 937–941, Seattle, USA, 2001.

[14] Microsoft Research. The Microsoft Vision SDK Library. World Wide Web - URL: http://www.research.microsoft.com/projects/VisSDK.

[15] D.S. Ryu, S.M. Kang, and S.W. Lee. Parameter-independent geometric document layout analysis. In *Proceedings of the 2000 International Conference on Pattern Recognition ICPR'00.*, pages 397–400, Barcelona, Spain, 2000.

[16] J. Sauvola, S. Haapakoski, H. Kauniskangas, T. Seppanen, M. Pietiklainen, and D. Doermann. A distributed management system for testing document image analysis algorithms. In *Proceedings of the 4th ICDAR'97*, pages 989–995, Ulm, Germany, 1997.

[17] J. Sauvola and H. Kauniskangas. MediaTeam Document Database II. CD-ROM collection of document images, University of Oulu, Finland. http://www.mediateam.oulu.fi/MTDB/index.html.

[18] K. Sobottka, H. Bunke, and H. Kronenberg. Identification of text on colored book and journal covers. In *ICDAR'99*, pages 57–60.

[19] L. Todoran, M. Aiello, C. Monz, and M. Worring. Logical structure detection for hetrogeneous document classes. In *Proc. SPIE Vol. 3407, Document Recognition and Retrieval VIII, Paul B. Kantor, Daniel P. Lopresti, Jiangying Zhou Eds.*, pages 99–111, San Jose, California, 2001.

[20] S. Tsujimoto and H. Asada. Major Components of a Complete Text Reading System. *Proceedings of the IEEE*, 80(7):1133–1149, 1992.

[21] V.Wu, R. Manmatha, and E.M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Transactions on PAMI*, 21(11):1224–1229, 1999.

[22] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[23] T. Watanabe and T. Sobue. Layout analysis of complex documents. In *Proceedings of the 2000 International Conference on Pattern Recognition ICPR'00.*, pages 447–450, Barcelona, Spain, 2000.

[24] B. Yanikoglu and L. Vincent. Pink panther: A complete environment for ground-truthing and benchmarking document page segmentation. *Pattern Recognition Letters*, 31(9):1191–1204, 1997.

[25] Y. Zhong, K. Karu, and A. K. Jain. Locating Text in Complex Color Images. *Proceedings of the Third International Conference on Document Analysis and Recognition*, 1:146–149, August 1995.

[26] Y. Zhong, K. Karu, and A. K. Jain. Locating Text in Complex Color Images. *Pattern Recognition*, 28(10):1523–1535, August 1995.

```
<!-- 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)>
```

Figure 8: The DTD used for the UvA data set.

Acknowledgements

Leon Todoran is supported by Senter den Haag and Océ Technologies BV, Venlo (IOP project IBV 96008).

ISIS reports

This report is in the series of ISIS technical reports. The series editor is Rein van den Boomgaard (rein@science.uva.nl). Within this series the following titles are available:

References

[1] F.J. Seinstra, D. Koelma, J.M. Geusebroek, F.C. Verster, and A.W.M Smeulders. Efficient applications in user transparent parallel image processing. Technical Report 2001-16, Intelligent Sensory Information Systems Group, University of Amsterdam, October 2001.

[2] E.A. Engbers and A.W.M Smeulders. Requirements for generic grouping in vision and an algorithm. Technical Report 2001-17, Intelligent Sensory Information Systems Group, University of Amsterdam, November 2001.

[3] A. Jonk, C. de Boer, R. van den Boomgaard, and A.W.M Smeulders. A case study in performance analysis of recognition of graphical signs - detecting arrows. Technical Report 2001-18, Intelligent Sensory Information Systems Group, University of Amsterdam, November 2001.

[4] T.V. Pham and A.W.M. Smeulders. Statistical strategy for object class recognition using part detectors. Technical Report 2001-19, Intelligent Sensory Information Systems Group, University of Amsterdam, December 2001.

[5] C.G.M. Snoek and M. Worring. Multimodal video indexing: A review of the state-of-the-art. Technical Report 2001-20, Intelligent Sensory Information Systems Group, University of Amsterdam, December 2001.

[6] A. Jonk, R. van de Boomgaard, and A.W.M Smeulders. Grouping lines. finding curvilinear structures in images. Technical Report 2001-21, Intelligent Sensory Information Systems Group, University of Amsterdam, December 2001.

[7] A. Jonk, R. van de Boomgaard, and A.W.M Smeulders. A line tracker. Technical Report 2001-22, Intelligent Sensory Information Systems Group, University of Amsterdam, December 2001.

[8] L. Todoran, M. Worring, and A.W.M. Smeulders. The uva color document dataset. Technical Report 2002-01, Intelligent Sensory Information Systems Group, University of Amsterdam, February 2002.

You may order copies of the ISIS technical reports from the corresponding author or the series editor. Most of the reports can also be found on the web pages of the ISIS group (http://www.science.uva.nl/research/isis).

Intelligent Sensory Information Systems
University of Amsterdam
The Netherlands