Available online at www.sciencedirect.com
SCIENCE DIRECT®

Pattern Recognition 39 (2006) 57–73

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Document zone content classification and its performance evaluation

Yalin Wang[a,*], Ihsin T. Phillips[b], Robert M. Haralick[c]

[a] *Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA*
[b] *Department of Computer Science, Queens College, CUNY, Flushing, NY 11367, USA*
[c] *The Graduate School, CUNY, New York, NY 10016, USA*

Received 24 November 2004; received in revised form 14 June 2005; accepted 14 June 2005

Abstract

This paper describes an algorithm for the determination of zone content type of a given zone within a document image. We take a statistical based approach and represent each zone with 25 dimensional feature vectors. An optimized decision tree classifier is used to classify each zone into one of nine zone content classes. A performance evaluation protocol is proposed. The training and testing data sets include a total of 24,177 zones from the University of Washington English Document Image database III. The algorithm accuracy is 98.45% with a mean false alarm rate of 0.50%.

Keywords: Pattern recognition; Document image analysis; Document layout analysis; Zone content classification; Background analysis; Decision tree classifier; Viterbi algorithm

1. Introduction

A document is varied in content. It can contain text, math, figure zones, etc. Each of these zones has its own characteristic features. For example, a math zone may contain symbols like $=$, $+$, $\sum$, $\int$, $\ldots$, which a text zone may not contain. Every page contains numerous zones. Each zone is specified by a rectangular box which encloses the zone and its zone type. This paper describes an algorithm for determination of the zone type given a document image and the coordinates of the leftmost-top and rightmost-bottom points of the zones on the document image.

A complete document image understanding system can transform paper documents into a hierarchical representation of their structure and content [1]. The transformed document representation enables document interchange, editing, browsing, indexing, filing, and retrieval. The zone classification technique plays the key role in the success of such a document understanding system. Not only is it useful for successive applications such as OCR [2], table understanding [3], etc., but it can be used to assist and validate document segmentation.

In the design of a zone classifier, a set of measurements are made on the zone. Each measurement is a feature. Some features are calculated along the horizontal, vertical, right-diagonal and left-diagonal directions of the zone. We design a set of signature-like background analysis structure to capture the statistical properties on the background. A feature vector of the zone is a tuple whose components are the measured values from the zone. Two sets of feature vectors are studied in our research. We employ a statistically based decision tree classifier [4] for the classification. Several methods are used in the decision tree classifier optimization to prevent the data over-fitting problem [5,6]. To enrich our model, we incorporate contextual constraints in the classification for some zones. For some types of zones, we use a hidden Markov model (HMM) and

[*] Corresponding author. UCLA Mathematics Department, Box 951555, Los Angeles, CA 90095, USA. Tel.: +1 310 206 7334; fax: +1 310 206 6673.

E-mail addresses: ylwang@math.ucla.edu, ylwang@u.washington.edu (Y. Wang), yun@image.cs.qc.edu (I.T. Phillips), haralick@gc.cuny.edu (R.M. Haralick).

0031-3203/$09.50 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.06.009

classify with a Viterbi algorithm [7] to get optimal classification results.

In Section 7, we describe a performance evaluation protocol for zone content classification experiments. The performance of the classifier is calculated from a contingency table. An entry, $N_{i,j}$, in the contingency table indicates the number of zones in the document image data base that are identified in the ground truth as class i and assigned by the decision rule to class j.

In our zone content classification experiment, the zones are the zone groundtruth entities from University of Washington English Document Image Database III (UWCDROM III) [8]. It includes 1600 scientific and technical document images with a total of 24,177 zones. The zone classes we consider are text with font size $\leqslant 18$ pt, text with font size $\geqslant 19$ pt, math, table, halftone, map/drawing, ruling, logo, and others. We report the experimental results using different feature sets, with and without classifier optimization. With the classifier optimization and the reduced feature set, our algorithm accuracy rate is 98.45% and the mean false alarm rate is 0.50%.

There is a constant interest in the document image analysis field on document zone content classification problem. However, most of common approaches focus on specific type zone extraction and recognition. For example, Xiao and Yan [9] worked on text region extraction problem, Zanibbi et al. [10] on Mathematics Expression recognition, Hu et al. [3] on table extraction problem, Chen et al. [11] and Pham [12] on logo detection, Li et al. [13] on image (halftone) extraction problem, Futrelle et al. [14] on diagram (drawing) extract and classification problem, etc. To the best of our knowledge, our group is the first group who systematically study the zone content classification and conduct experiments on a large data set, UW Document Image Database III [8]. Our research achieves very good results and should be useful to many researches in document image analysis research field.

The rest of this paper is divided into 8 parts. In Section 2, a literature review is given. In Section 3, a formal zone content classification problem statement is presented. In Section 4, we give our background analysis structure analysis, and a detailed description of two sets of features we studied in the experiments. In Section 5, the decision tree classifier and our methods to eliminate data over-fitting are described. Section 6 describes how we incorporated content constraint to improve classification results using the HMM model. The performance evaluation protocol and experimental results are reported in Section 7. Our conclusion and statement of future work are discussed in Section 8.

2. Literature review

The problems of segmenting document pages into homogeneous regions and assigning functional labels to each region are of importance in automatic document understanding systems [1,15]. The construction of a document hierarchy, given an input document, consists of two major steps. The first step is the correct detection and partitioning of entities within the hierarchy, and the second step is the correct classification of those detected entities [16]. The zone content classification technique plays a key role in the second step, logical page structure analysis.

By the condition of whether or not the method considers segmentation, the algorithms can be classified into two major categories: (1) independent of segmentation; (2) including segmentation. The methods belong to the first category concentrate on extracted regions. The methods in the second category usually employ a bottom-up [16] document segmentation method first and then classify the segmented regions in the second phase. Our algorithm falls into the first category.

By the strategies or methods in which algorithms are used in zone content classification, the algorithms can be classified into two major categories: (1) rule-based/grammar-driven and (2) statistical-based (either parametric and non-parametric). The rule-based/grammar-driven algorithms use a set of ad hoc rules or pre-defined syntax rules of the grammars to derive decisions. The ad hoc rules or the syntax of the grammar are fixed and empirically determined. In the statistical-based algorithms, the required free parameters that are used in the decision process are obtained by an off-line training processes.

In the literature, some papers present algorithms for document page classification [17–19]. The goal of their work is to classify the whole page into different classes, such as “title page”, “index”, “regular page”, etc. [19]. Although they use a similar technique to our approach, our work focuses on the functional labeling of segmented regions. So the discussion of page classification is out of the scope of this paper.

It is also interesting to note that many logical page structure analyses are task oriented. For example, table understanding algorithms are aimed at extracting table regions from non-table regions and the systems to facilitate OCR are only designed to separate text from non-text. They are solving only a part of the zone content classification problem. The methods they employ are similar to ours and their results may be compared with ours. The following are a handful of selected algorithms within the above defined categories.

Sivaramakrishnan et al. [20] extracted features for each zone such as run length mean and variance, spatial mean and variance, fraction of the total number of black pixels in the zone, and the zone width ratio for each zone. The decision tree classifier was used to assign a zone class to one of nine classes on the basis of its feature vector. The experiments were conducted on 979 scientific document pages with a total of 13,726 zones drawn from UW-I database [21]. The accuracy rate was 96.67% and mean false alarm rate was 1.15%.

Krishnamoorthy et al. [22] combined the nested X–Y tree decomposition of rectangles into rectangles with recursive horizontal and vertical application of a publication-specific “block grammar” to determine the major logical components of technical articles. This model-driven approach isolates specific document components for selective OCR. In their experiments, more than 20 types of document entities can be identified in sample pages from the *IBM Journal of Research and Development* and *IEEE Transactions on Pattern Analysis and Machines Intelligence*. The drawback of this method is the skew sensitivity and the effort required to construct the grammars in lex/yacc notation.

Antonacopoulos and Ritchings [23] presented a background analysis (white tiles) based algorithm. It can represent and classify complex-shaped printed region. Based on the description of white space inside regions, it classified a given region into text, graphics, line art regions.

Fan and Wang [24] presented a document block classification algorithm using density feature and connectivity histogram. The attribute of each segmented block is divided into three classes: text, graphics, and image. First, they utilized the density feature to determine whether the block is a text or non-text block. If the block is classified as a non-text block, the connectivity histogram is employed to further classify it into graphics or image block. They conducted their experiments on 30 English and Chinese documents. Their average classification accuracy was 94%.

Jain and Yu [25] developed a bottom-up method to partition a page into columns of *text*, *drawings*, *images*, *table regions*, and *rulers*. Following the application of a hierarchical Hough transform to connected components, the estimated skew is accommodated by introducing generalized text lines. Foreground pixels are grouped into rectangular blocks with adjacent same-length horizontal runs preserved as nodes in a block adjacent graph (BAG). The BAG nodes are successively grouped into connected components, text lines, and region blocks. The segmented regions were classified into text, image, table, and drawing classes using empirical rules and thresholds. Selected results from performance tests on 150 varied page images were illustrated.

Le et al. [26] proposed an automated labeling of zones from scanned images with labels such as titles, authors, affiliations, and abstracts. The labeling is based on features calculated from optical character recognition (OCR) output, neural network models, machine learning methods, and a set of rules that is derived from an analysis of the page layout for each journal and from generic typesetting knowledge for English text.

Lee et al. [27] presented a knowledge-based method for sophisticated geometric structure analysis of technical journal pages. The knowledge base encodes geometric characteristics that are not only common in technical journals but also publication-specific in the form of rules. It takes the hybrid of top-down and bottom-up techniques and consists of two phases: region segmentation and identification. They used sets of rules to do region identification. The resulting regions classes include *text line*, *equation*, *image*, *drawing*, *table*, and *ruler*. Their experimental results with 372 images scanned from the *IEEE Transactions on Pattern Analysis and Machine Intelligence* showed that the method has performed geometric structure analysis successfully on more than 99% of the test images.

Jain et al. [28] presented a hierarchical approach for extracting homogeneous regions from on-line handwritten documents. Their algorithm identifies and processes ruled and unruled tables, text and drawings. The on-line document is first segmented into regions with only text strokes and regions with both text and non-text strokes. The text region is further classified as unruled table or plain text. Stroke clustering is used to segment the non-text regions. Each non-text segment is then classified as drawing, ruled table and underlined keyword using stroke properties. Their experimental data was collected from 123 different people without any restriction on the style or content of data. About 99.9% of the text strokes were correctly classified. A classification accuracy of 85.0% was achieved on a data set containing 105 unruled tables and 35 text regions.

In their newspaper segmentation work [29], Mitchell et al. used a bottom-up approach to segment the image into patterns. Then each pattern is classified into one of seven types, namely, text, title, inverse text, photo, graphic/drawing, vertical line, and horizontal line. The patterns were classified using a series of rules based on the pattern size, shape, black pixel numbers and run-length characteristics. They submitted their results to the *First International Newspaper Segmentation Contest* [30].

Harit et al. [31] present a model based document image segmentation scheme that uses XML_DTDs (eXtensible Mark-up Language-Document Type Definition). Given a document image, the algorithm has the ability to select the appropriate model. A wavelet based tool was designed for distinguishing text from non-text regions and characterization of font sizes. The model based analysis scheme makes use of the wavelet tool for identifying the logical components of a document image. Overall, they obtained about a 90% correct segmentation and identification results for documents of different languages.

Some researchers used HMM for document image segmentation and OCR system. Kopec and Chow [2] proposed *Document Image Decoding* which unifies some aspects of reprocessing, layout analysis and character recognition. DID is a communication theory approach to document image recognition patterned after the use of HMMs in speech recognition. The decoding process, based on dynamic programming, attempts to identify the most likely sequence of transitions from the observed pixels. A model with 1700 nodes and over 600 branches was run on 48 columns from 10 pages from the Yellow Pages in about 36 h. The error rate on the listings was less than 2% for names and less than 0.5% for telephone number.

Bazzi et al. [32] drew on their HMM tools for speech and handwritten-character recognition to develop a multifont reader with language-independent algorithms and shape features, and language-dependent orthographic rules, character models, lexicons, and grammars. Their system was tested on UWCDROM III and the DARPA Arabic OCR Corpus. The results indicated that using a 30, 000-word English lexicon and word-bigram frequencies reduces the error rate by about a factor of three over use of character bigrams and trigrams alone. Combining them gives almost another factor of two. The error rate on English was about three times lower than on Arabic.

Li et al. [13] proposed an algorithm that models images by two-dimensional HMMs. The HMM considers feature vectors statistically dependent through an underlying state process assumed to be a Markov mesh, which has transition probabilities conditioned on the states of neighboring blocks from both horizontal and vertical directions. The dependency in two dimensions is reflected simultaneously. They demonstrated their methods on both aerial and document images.

3. Problem statement

Let $\mathscr{A}$ be a set of zone entities in a given document page. Let $\mathscr{L}$ be a set of content labels, such as text, table, math, etc. The function $f : \mathscr{A} \rightarrow \mathscr{L}$ associates each element of $\mathscr{A}$ with a label. The function $V : \mathscr{A} \rightarrow \Lambda$ specifies measurements made on each element of $\mathscr{A}$, where Λ is the measurement space.

The zone content classification problem can be formulated as follows: *Given a zone set $\mathscr{A}$ and a content label set $\mathscr{L}$, find a classification function $f : \mathscr{A} \rightarrow \mathscr{L}$, that has the maximum probability*:

$$P(f(\mathscr{A})|V(\mathscr{A})). \tag{1}$$

Let $\mathscr{S}_1$ and $\mathscr{S}_2$ be two exclusive and exhaustive subsets of $\mathscr{A}$. To consider context information in our classification, we consider two different assumptions made on the joint prior probability functions [33]. One set, $\mathscr{S}_1$, is with an independent assumption and the other set, $\mathscr{S}_2$, is with Markov dependence assumption. We assume the labels of a zone in set $\mathscr{S}_1$ contributes no information relative to the label of another zone in set $\mathscr{S}_2$ and measurements of a zone in set $\mathscr{S}_1$ contribute no information relative to the label of any zone in set $\mathscr{S}_2$. We have

$$\begin{aligned} P(f(\mathscr{A})|V(\mathscr{A})) &= \prod_{i=1,2} P(f(\mathscr{S}_i)|V(\mathscr{A})) \\ &= \prod_{i=1,2} P(f(\mathscr{S}_i)|V(\mathscr{S}_i)). \end{aligned} \tag{2}$$

We have two similar conditions as follows:

1. Conditioned on all the measurements, the label of a zone contributes no information relative to the label of another zone;
2. Measurements of a zone contribute no information relative to the label of another zone.

We assume conditions 1, 2 are true for set $\mathscr{S}_1$ and only condition 2 holds for set $\mathscr{S}_2$. We also assume the elements of set $\mathscr{S}_2$ are $\{Z_1, Z_2, \ldots, Z_s\}$, where s is the zone number in $\mathscr{S}_2$. We can have $P(f(\mathscr{S}_1)|V(\mathscr{S}_1)) = \prod_{\alpha \in \mathscr{S}_1} P(f(\alpha)|V(\alpha))$ and

$$\begin{aligned} &P(f(\mathscr{S}_2)|V(\mathscr{S}_2)) \\ &\quad = P(f(Z_s)|V(Z_s), f(Z_{s-1}), \ldots, f(Z_1))P(f(Z_{s-1}) \\ &\qquad \times |V(Z_{s-1}), f(Z_{s-2}), \ldots, f(Z_1)) \\ &\qquad \ldots f((Z_1)|V(Z_1)). \end{aligned}$$

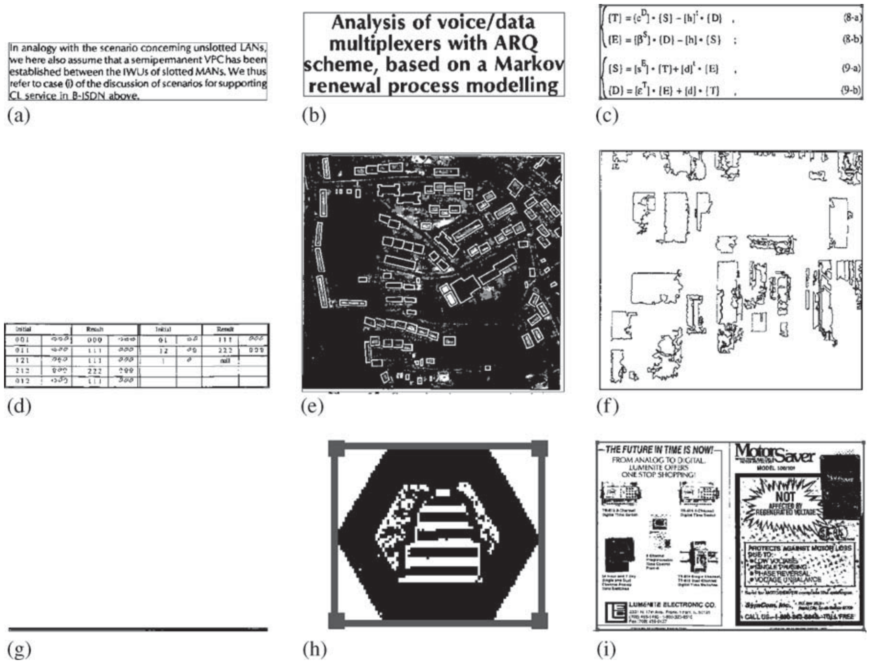

In analogy with the scenario conceming unslotted LANs, we here also assume that a semipermanent VPC has been established between the IWUs of slotted MANs. We thus refer to case (i) of the discussion of scenarios for supporting CL service in B-ISDN above.
(a)
Analysis of voice/data multiplexers with ARQ scheme, based on a Markov renewal process modelling
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

Fig. 1. Illustrates examples of nine zone content classes. (a) Text 1 class; (b) Text 2 class; (c) Math class; (d) Table class; (e) Halftone class; (f) Map/drawing class; (g) Ruling class; (h) Logo class; (i) Others class.

The problem in Eq. (2) can be solved by maximizing each individual probability for $\mathscr{S}_1$ and $\mathscr{S}_2$.

In our zone content classification experiment, the elements in set $\mathscr{A}$ are the zone groundtruth entities from UWCDROM-III document image database [8]. The elements of set $\mathscr{L}$ are text zone with font size $\leqslant 18$ pt, text with font size $\geqslant 19$ pt, math, table, halftone, map/drawing, ruling, logo, and others. The examples of each class are shown in Fig. 1. $V(\tau)$ is a feature vector generated for τ, where $\tau \in \mathscr{A}$. Elements of set $\mathscr{S}_1$ are zones in the live-matter part and elements of set $\mathscr{S}_2$ are zones in the header and footer parts in a given page. We used a decision tree classifier to compute each individual probability in set $\mathscr{S}_1$ and use a HMM to model the dependency in set $\mathscr{S}_2$.

4. Features for zone content classification

In this section, first we give our new background analysis structure definitions in Section 4.1. Second, we give our two sets of features we studied in the experiments in Sections 4.2 and 4.3.

4.1. Background analysis structure definitions

Although some background analysis techniques can be found in the literature [34,35], none of them, to the best of our knowledge, have extensively studied the statistical characteristics of their background structure. Instead, they mainly use heuristic rules on their background structure. We define a new background analysis structure. Our background analysis is based on some basic units: horizontal and vertical blank blocks. These signature-like features are designed to give us information on the distributions of the big foreground chunks in a given document entity. We use several definitions to describe the structure.

Assume black pixels are foreground and white pixels are background.

Definition 1. Let $\mathscr{Z}$ represent a *zone* with R rows and C columns. Let (x_1, y_1) be the coordinate of its lefttop vertex, $\mathscr{Z} = \{(r, c) \in Z \times Z | x_1 \leqslant r < x_1 + R, y_1 \leqslant c < y_1 + C\}$.

Definition 2. Let p be a *horizontal white run*, $p = ((r_1, c_1), \ldots, (r_n, c_n))$, where $(r_i, c_i) \in \mathscr{Z}$, $r_i = r_{i-1}$, $c_i = c_{i-1} + 1$, for $i = 2, \ldots, n$, and pixel (r_1, c_1), (r_n, c_n) must have a black pixel or the zone border on its left and right side, respectively. For each run, we call the location of the starting pixel of the run and its horizontal length as $\text{Row}(p)$, $\text{Column}(p)$, and $\text{Length}(p)$, respectively.

Definition 3. Let $\mathscr{H}R$ be a *horizontal blank block*, $\mathscr{H}R = (p_1, \ldots, p_n)$, where $\text{Row}(p_i) = \text{Row}(p_{i-1}) + 1$, $\text{Column}(p_i) = \text{Column}(p_{i-1})$, $\text{Length}(p_i) = \text{Length}(p_{i-1})$, for $i = 2, \ldots, n$. Clearly, the same idea can be applied to define *vertical white run* and *vertical white blank block*, $\mathscr{V}R$.

Definition 4. A horizontal blank block $\mathscr{H}R = b_r \times b_c$, with lefttop vertex coordinate (x_{b1}, y_{b1}), is a *large horizontal blank block* if and only if it satisfies the following conditions:

- its row numbers and column numbers are large enough compared with the current zone. Specifically, $b_c/C > \theta_1$, where θ_1 is 0.1. This number was statistically determined by our experiment on another table data set;
- It does not touch left or right side of the zone bounding box, i.e. $x_{b1} \neq x_1$ and $x_{b1} + b_c \neq x_1 + C$;

where C is the column number in the zone.

Definition 5. A vertical blank block $\mathscr{V}R = b_r \times b_c$, with lefttop vertex coordinate (x_{b1}, y_{b1}), is a *large vertical blank block* if and only if it satisfies the following conditions:

- Its row number and column number are large enough compared with the current zone. Specifically, $b_r \geqslant mh$ and $b_c/mw > \theta_2$, where mh and mw are the median height and median width of text glyphs in the zone. θ_2 is empirically determined as 1.4;
- It does not touch left or right side of the zone bounding box, i.e. $x_{b1} \neq x_1$ and $x_{b1} + b_c \neq x_1 + C$;

where C is the column number in the zone.

4.2. Zone content classification features

Every zone in the document is considered to be rectangular. Properties of each zone are used by the classifier for the process of classification. For a given zone, we designed several groups of features such as run length features, spatial features, autocorrelation features, background features, text glyph feature etc. with a total of 69 features.

For each zone, run length and spatial features are computed for each line along four different canonical directions: horizontal, vertical, left-diagonal, and right-diagonal. These four directions are shown in Fig. 2. We use subscript h, v, l and r to represent four directions. When discriminating foreground and background features is necessary, we use superscript 0 and 1 to represent foreground and background features, respectively. For example, $rlmean_h^0$ represents background run length mean feature computed in horizontal direction.

In the following, we describe each feature in detail.

4.2.1. Run length features

A *run* is a list of contiguous foreground or background pixels in a given direction. A *run length* is the number of pixels in a given foreground or background run. Our run length features include foreground/background

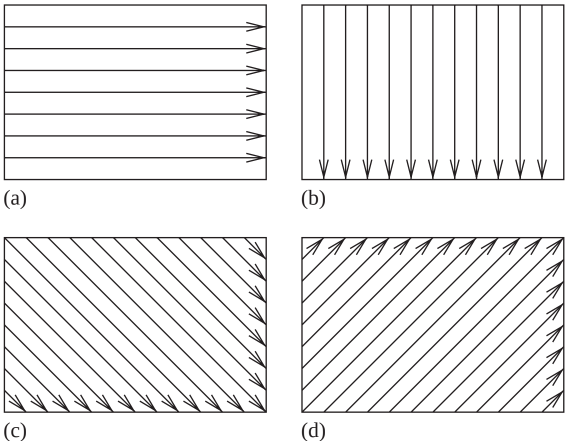


Fig. 2. Illustrates the four directions in which we compute run length and spatial features. (a) Horizontal; (b) vertical; (c) left-diagonal; (d) right-diagonal.

run length mean and variance in each of the four directions.

1. We denote the 8 sets including all the foreground and background run lengths on the four directions in a given zone as $\mathscr{R}L_h^0$, $\mathscr{R}L_v^0$, $\mathscr{R}L_l^0$, $\mathscr{R}L_r^0$, $\mathscr{R}L_h^1$, $\mathscr{R}L_v^1$, $\mathscr{R}L_l^1$ and $\mathscr{R}L_r^1$. The first group of run length features are total foreground and background run length number on the four directions in a given zone. Together we have 8 features.

$$|\mathscr{R}L_h^0| \tag{3}$$

$$|\mathscr{R}L_v^0| \tag{4}$$

$$|\mathscr{R}L_l^0| \tag{5}$$

$$|\mathscr{R}L_r^0| \tag{6}$$

$$|\mathscr{R}L_h^1| \tag{7}$$

$$|\mathscr{R}L_v^1| \tag{8}$$

$$|\mathscr{R}L_l^1| \tag{9}$$

$$|\mathscr{R}L_r^1| \tag{10}$$

2. Foreground and background run length mean features on four directions in a given zone.

$$rlmean_h^0 = \frac{1}{|\mathscr{R}L_h^0|} \sum_{rl \in \mathscr{R}L_h^0} rl, \tag{11}$$

$$rlmean_v^0 = \frac{1}{|\mathscr{R}L_v^0|} \sum_{rl \in \mathscr{R}L_v^0} rl, \tag{12}$$

$$rlmean_l^0 = \frac{1}{|\mathscr{R}L_l^0|} \sum_{rl \in \mathscr{R}L_l^0} rl, \tag{13}$$

$$rlmean_r^0 = \frac{1}{|\mathscr{R}L_r^0|} \sum_{rl \in \mathscr{R}L_r^0} rl, \tag{14}$$

$$rlmean_h^1 = \frac{1}{|\mathscr{R}L_h^1|} \sum_{rl \in \mathscr{R}L_h^1} rl, \tag{15}$$

$$rlmean_v^1 = \frac{1}{|\mathscr{R}L_v^1|} \sum_{rl \in \mathscr{R}L_v^1} rl, \tag{16}$$

$$rlmean_l^1 = \frac{1}{|\mathscr{R}L_l^1|} \sum_{rl \in \mathscr{R}L_l^1} rl, \tag{17}$$

$$rlmean_r^1 = \frac{1}{|\mathscr{R}L_r^1|} \sum_{rl \in \mathscr{R}L_r^1} rl. \tag{18}$$

3. Foreground and background run length variance features on the four directions in a given zone. They can be obtained by calculating the mean of the squares of all the run lengths in the zone and subtracting them by the square of the run length mean. Specifically, they can be computed by the equations below.

$$rlvar_h^0 = \frac{\sum_{rl \in \mathscr{R}L_h^0} rl^2}{|\mathscr{R}L_h^0|} - (rlmean_h^0)^2, \tag{19}$$

$$rlvar_v^0 = \frac{\sum_{rl \in \mathscr{R}L_v^0} rl^2}{|\mathscr{R}L_v^0|} - (rlmean_v^0)^2, \tag{20}$$

$$rlvar_l^0 = \frac{\sum_{rl \in \mathscr{R}L_l^0} rl^2}{|\mathscr{R}L_l^0|} - (rlmean_l^0)^2, \tag{21}$$

$$rlvar_r^0 = \frac{\sum_{rl \in \mathscr{R}L_r^0} rl^2}{|\mathscr{R}L_r^0|} - (rlmean_r^0)^2, \tag{22}$$

$$rlvar_h^1 = \frac{\sum_{rl \in \mathscr{R}L_h^1} rl^2}{|\mathscr{R}L_h^1|} - (rlmean_h^1)^2, \tag{23}$$

$$rlvar_v^1 = \frac{\sum_{rl \in \mathscr{R}L_v^1} rl^2}{|\mathscr{R}L_v^1|} - (rlmean_v^1)^2, \tag{24}$$

$$rlvar_l^1 = \frac{\sum_{rl \in \mathscr{R}L_l^1} rl^2}{|\mathscr{R}L_l^1|} - (rlmean_l^1)^2, \tag{25}$$

$$rlvar_r^1 = \frac{\sum_{rl \in \mathscr{R}L_r^1} rl^2}{|\mathscr{R}L_r^1|} - (rlmean_r^1)^2. \tag{26}$$

4.2.2. Spatial features

Spatial features are designed to capture the foreground pixel distribution information. We denote the foreground pixel set in a given zone as $\mathscr{F}$. Spatial mean, μ, and spatial variance, δ, can be defined as

$$\mu = \frac{1}{|\mathscr{F}|} \sum_{p \in \mathscr{F}} w_p, \quad \delta = \frac{1}{|\mathscr{F}|} \sum_{p \in \mathscr{F}} (w_p - \mu)^2,$$

where w_p is a weight assigned to each 1-pixel. Since we defined four different line directions, we can have four different definitions for w_p. Hence, we have 8 spatial features including spatial mean and variance on four directions.

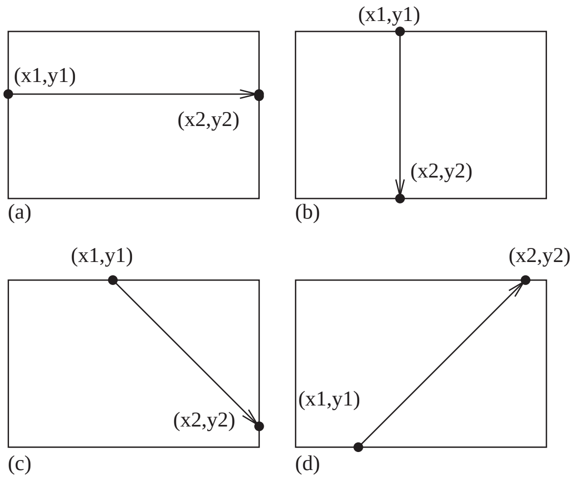


Fig. 3. Illustrates the coordinates of starting and ending points for different passes. (a) Horizontal pass; (b) vertical pass; (c) left-diagonal pass; (d) right-diagonal pass.

As shown in Fig. 2, we have four different directions to compute run length. In each direction, we start the computation from a point on a zone border and continue in a given direction until we hit another zone border again. We call such a computation route a *pass*. For every pass the sum of run lengths in the foreground gives the *pass projection*. Given one direction, each foreground pixel belongs and only belongs to one pass. We let the foreground pixel in the same pass have the same weights so we have four different weight definitions according to each direction. As shown in Fig. 3, we denote the starting and ending pixel coordinates of a pass as (x_1, y_1) and (x_2, y_2), respectively.

The weights for horizontal, vertical, left-diagonal, and right-diagonal directions are denoted as w_h, w_v, w_l, w_r. Their definitions are shown in the below equations.

$$w_h = y_1, \quad w_v = x_1, \quad w_l = x_1 + y_1, \quad w_r = y_2 - x_2.$$

Denote the set of passes in four directions as $\mathscr{L}_h$, $\mathscr{L}_v$, $\mathscr{L}_l$, and $\mathscr{L}_r$. For a pass, say, l_h, we denote its pass projection as $\mathit{proj}_{h,l}$. In our algorithm, we compute spatial means and spatial variances as follows:

$$\mathit{spmean}_h = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_h} w_h \times \mathit{proj}_{h,l}, \tag{27}$$

$$\mathit{spmean}_v = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_v} w_v \times \mathit{proj}_{v,l}, \tag{28}$$

$$\mathit{spmean}_l = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_l} w_l \times \mathit{proj}_{l,l}, \tag{29}$$

$$\mathit{spmean}_r = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_r} w_r \times \mathit{proj}_{r,l}, \tag{30}$$

$$\mathit{spvar}_h = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_h} [\mathit{proj}_{h,l} \times (w_h - \mathit{spmean}_h)^2], \tag{31}$$

$$\mathit{spvar}_v = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_v} [\mathit{proj}_{v,l} \times (w_v - \mathit{spmean}_v)^2], \tag{32}$$

$$\mathit{spvar}_l = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_l} [\mathit{proj}_{l,l} \times (w_l - \mathit{spmean}_l)^2], \tag{33}$$

$$\mathit{pvar}_r = \frac{1}{|\mathscr{F}|} \sum_{l \in \mathscr{L}_r} [\mathit{proj}_{r,l} \times (w_r - \mathit{spmean}_r)^2]. \tag{34}$$

4.2.3. Autocorrelation features

We compute 32 autocorrelation features by means of the fast Fourier transform [36]. The Fourier transform has wide range of applications in image processing problems. Specifically, we define four functions on each given pass, such as pass projection function, number of foreground runs function, run length mean function, and spatial mean function. For the whole zone, we obtain a sequence of these function values. Then we compute the autocorrelation of these sequences and used some statistics as our autocorrelation features.

In the following, we describe the four functions first and follow a brief introduction of correlation computation using the Fourier Transform. Then we define the total 32 autocorrelation features we used in our experiments.

1. Denote the set of run length in a horizontal pass, l, as $\mathscr{R}L_{h,l}$. Similarly we can define $\mathscr{R}L_{v,l}$, $\mathscr{R}L_{l,l}$, $\mathscr{R}L_{r,l}$ for other three direction passes. Among the four functions, the pass projection function has been defined earlier. They are denoted by

$$\mathit{proj}_{h,l} \tag{35}$$

$$\mathit{proj}_{v,l} \tag{36}$$

$$\mathit{proj}_{l,l} \tag{37}$$

$$\mathit{proj}_{r,l} \tag{38}$$

The function of the number of foreground runs on each pass are straightforward.

$$|\mathscr{R}L_{h,l}| \tag{39}$$

$$|\mathscr{R}L_{v,l}| \tag{40}$$

$$|\mathscr{R}L_{l,l}| \tag{41}$$

$$|\mathscr{R}L_{r,l}| \tag{42}$$

The function of run length spatial mean on each pass can be defined as follows:

$$\mathit{rlmean}_{h,l} = \frac{\mathit{proj}_{h,l}}{|\mathscr{R}L_{h,l}|}, \tag{43}$$

$$\mathit{rlmean}_{v,l} = \frac{\mathit{proj}_{v,l}}{|\mathscr{R}L_{v,l}|}, \tag{44}$$

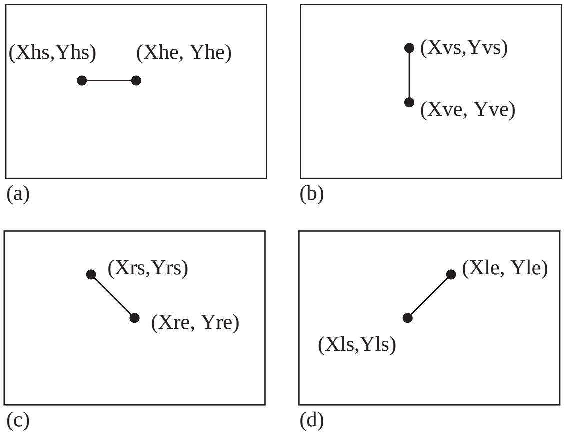


Fig. 4. Illustrates the coordinates of starting and ending points for different run length. (a) Horizontal run length; (b) vertical run length; (c) left-diagonal run length; (d) right-diagonal run length.

$$rlmean_{l,l} = \frac{proj_{l,l}}{|\mathscr{R}L_{l,l}|}, \tag{45}$$

$$rlmean_{r,l} = \frac{proj_{r,l}}{|\mathscr{R}L_{r,l}|}. \tag{46}$$

To define the spatial mean function for a line, we need define pos and $leng$ functions for a given run length. In Fig. 4, we show the starting and ending coordinates of run lengths in different directions. For the run length shown in Fig. 4, the definition of pos and $leng$ functions are

$$\begin{aligned}
&pos_{h,rl} = x_{h,s}, \quad leng_{h,rl} = x_{h,e} - x_{h,s},\\
&pos_{v,rl} = y_{v,s}, \quad leng_{v,rl} = y_{v,e} - y_{v,s},\\
&pos_{l,rl} = x_{l,s}, \quad leng_{l,rl} = x_{l,e} - x_{l,s},\\
&pos_{r,rl} = x_{r,s}, \quad leng_{r,rl} = x_{r,e} - x_{r,s}.
\end{aligned} \tag{47}$$

The spatial mean function for each line can be defined as follows:

$$\begin{aligned}
spmean_{h,l} = \frac{1}{proj_{h,rl}} \Bigg(&\sum_{rl\in\mathscr{R}L_{h,l}} pos_{h,rl} \times leng_{h,rl}\\
&+ \frac{1}{2}\Bigg(\sum_{rl\in\mathscr{R}L_{h,l}} (leng_{h,rl})^2 - proj_{h,rl}\Bigg)\Bigg),
\end{aligned} \tag{48}$$

$$\begin{aligned}
spmean_{v,l} = \frac{1}{proj_{v,rl}} \Bigg(&\sum_{rl\in\mathscr{R}L_{v,l}} pos_{v,rl} \times leng_{v,rl}\\
&+ \frac{1}{2}\Bigg(\sum_{rl\in\mathscr{R}L_{v,l}} (leng_{v,rl})^2 - proj_{v,rl}\Bigg)\Bigg),
\end{aligned} \tag{49}$$

$$\begin{aligned}
spmean_{l,l} = \frac{1}{proj_{l,rl}} \Bigg(&\sum_{rl\in\mathscr{R}L_{l,l}} pos_{l,rl} \times leng_{l,rl}\\
&+ \frac{1}{2}\Bigg(\sum_{rl\in\mathscr{R}L_{l,l}} (leng_{l,rl})^2 - proj_{l,rl}\Bigg)\Bigg),
\end{aligned} \tag{50}$$

$$\begin{aligned}
spmean_{r,l} = \frac{1}{proj_{r,rl}} \Bigg(&\sum_{rl\in\mathscr{R}L_{r,l}} pos_{r,rl} \times leng_{r,rl}\\
&+ \frac{1}{2}\Bigg(\sum_{rl\in\mathscr{R}L_{r,l}} (leng_{r,rl})^2 - proj_{r,rl}\Bigg)\Bigg).
\end{aligned} \tag{51}$$

2. After we compute one function on each pass, we obtain a sequence of values, indexed by the pass number. For example, for the spatial mean function, we can get four sequences, $spmean_{h,k}$, $k = 0, \ldots, |\mathscr{L}_h| - 1$; $spmean_{v,k}$, $k = 0, \ldots, |\mathscr{L}_v| - 1$; $spmean_{r,k}$, $k = 0, \ldots, |\mathscr{L}_r| - 1$; $spmean_{l,k}$, $k = 0, \ldots, |\mathscr{L}_l| - 1$. Similarly we can get all the sequences for the other three functions.

We define the autocorrelation function on any sequence g_j, $j = 0, \ldots, N-1$ as

$$Autcor(g, g)_j = \sum_{k=0}^{N-1} g_{j+k} g_k.$$

We can use this equation to compute the autocorrelation functions of the functions defined earlier.

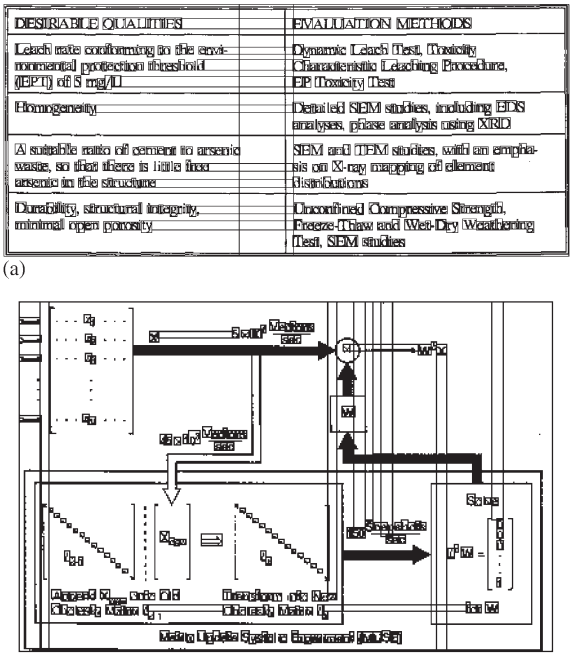
The *discrete correlation theorem* says that this discrete correlation of one real function g is one member of the discrete Fourier transform pair [37]

$$Autcor(g, g)_j \Leftrightarrow G_k G_k^*,$$

where G_k is the discrete Fourier transform of g_j and the asterisk denotes complex conjugation.

Based on this theorem, we can compute autocorrelations using the FFT as follows: FFT the data set, multiply the transform by the complex conjugate of itself, and inverse transform the product [38]. The result, say, r_k, will formally be a complex vector of length N. However, it will have all its imaginary parts zero since the original data set was real. The components of r_k are the values of the correlation at different lags, with positive and negative lags stored in the wrap-around order: The correlation at zero lag is in r_0, the first component; the correlation at lag 1 is in r_1, the second component; the correlation at lag -1 is in r_{N-1}, the last component; etc.

3. Using the FFT, we obtain a sequence of autocorrelation function values. The index for which the autocorrelation function goes to 10% of its maximum value is calculated. Another feature of interest is the slope of the tangent to the autocorrelation function values whose indexes are close 0. We used the general linear least squares method [38] to compute the slope of the points near r_0.



DESIRABLE QUALITIES	EVALUATION METHODS
Leach rate conforming to the environmental protection threshold (EPT) of 5 mg/L	Dynamic Leach Test, Toxicity Characteristic Leaching Procedure, EP Toxicity Test
Homogeneity	Detailed SEM studies, including EDS analyses, phase analysis using XRD
A suitable ratio of cement to arsenic waste, so that there is little free arsenic in the structure	SEM and TEM studies, with an emphasis on X-ray mapping of element distributions
Durability, structural integrity, minimal open porosity	Unconfined Compressive Strength, Freeze-Thaw and Wet-Dry Weathering Test, SEM studies

(a)

(b)

Fig. 5. Illustrates the example bounding boxes of large horizontal blank blocks, large vertical blank blocks, and text glyphs. The so-called text glyphs are labeled by a statistical glyph filter. (a) A table zone example; (b) a map/drawing zone example.

We define 16 functions and take 2 statistics for each function. The autocorrelation group contributes 32 features to the feature set.

4.2.4. Background features

Not only are foreground features important for zone content classification, but background features for each zone are also important. We have considered the background run length in run length feature set discussion. In this section, we introduce two background oriented features.

The first feature is straightforward: a fraction of the number of black pixels to the total number of pixels in the zone, br, $br = \#$ of background pixel/# of total pixel.

The second feature is defined using our background analysis based technique. The basic background analysis structures are defined in Section 4.1.

The background feature is: The total area of large horizontal and large vertical blank blocks, A,

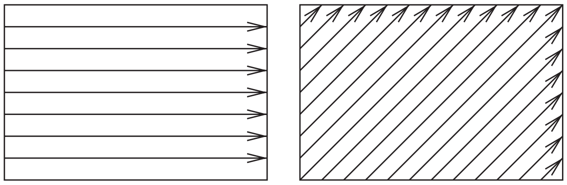
$$A = \sum_{R \in \mathscr{B}} Area(R), \tag{52}$$

where $\mathscr{B} = \{R | R = \mathscr{H}R \text{ or } \mathscr{V}R, \mathscr{H}R \subset \mathscr{Z} \text{ and } \mathscr{V}R \subset \mathscr{Z}\}$.

Fig. 5 shows two examples with large horizontal and vertical blank blocks.

4.2.5. Text Glyph feature

Most of zones have some text glyphs. The information that how many text glyphs a given zone has is also an useful feature. The number of text glyphs in this zone, W, normalized by the zone area is the text glyph feature.



(a) (b)

Fig. 6. Illustrates the two directions in which we compute run length and spatial features. (a) Horizontal; (b) diagonal.

$$\frac{W}{Area(\mathscr{Z})}. \tag{53}$$

The so-called text glyphs are not from any OCR output. They are outputs of a statistical glyph filter. The inputs of this filter are the glyphs after finding connected component operation. The statistical glyph filter classifies each connected component into one of two classes: text glyph and non-text glyph. The filter uses a statistical method to classify glyphs and was extensively trained on UWCDROM-III document image database. Fig. 5 illustrates two zone examples showing the overlayed bounding boxes text glyphs.

4.2.6. Area feature

The area covered by the zone, $Area(\mathscr{Z})$, is calculated by multiplying the length of the zone with its width. The length and width of a zone can be obtained from the coordinates of the zone, $Area(\mathscr{Z}) = R \times C$.

4.2.7. Column width ratio feature

It is a common observation that math zones and figure zones have a smaller width compared to text zones. For every zone, the quotient of the zone width and the width of its column is calculated.

$$\frac{C}{Width_{column}}, \tag{54}$$

where $Width_{column}$ is the width of the text column in which the zone is.

4.3. Zone content feature reduction

In the early stage of our research, we used as many features as possible. After we obtained more understanding about the zone content classification problem, we studied reducing the number of features while still retaining the original accuracy. We were able to reduce the feature vector size from 69 to 25 and get comparable classification accuracy.

For each zone, run length and spatial features are computed for each line along two different canonical directions: horizontal, diagonal. These two directions are shown in Fig. 6. We list below the reduced set of 25 features.

Their full definitions can be found in Section 4.2.

- Run length features. They are Eqs. (7), (10), (11), (14), (15), (18), (19), (22), (23) and (26);
- Spatial features. They are Eqs. (27), (30), (31), and (34).
- Autocorrelation features. The function definitions are Eqs. (35), (38), (39), (42), (43), (46), (48), and (51). After we compute one function on each run segment, we get a sequence of values, indexed by the run segment number. Using the Fast Fourier Transform [38], we can get the autocorrelation functions value for every function. Each feature is the slope of the tangent to the autocorrelation function values whose indexes are close to 0.
- Background feature, Eq. (52).
- Text glyph feature, Eq. (53).
- Column width ratio feature, Eq. (54).

5. Decision tree classifier

5.1. Classification process

A decision tree classifier makes the assignment through a hierarchical decision procedure. The classification process can be described by means of a tree, in which at least one terminal node is associated with each class and non-terminal nodes represent various collections of mixed classes [4].

For the construction of a decision tree, we need a training set of feature vectors with true class labels. Let $U = \{u_k : k = 1, \ldots, N\}$ be the unit-training set to be used to design a binary tree classifier. Each unit u_k has an associated measurement X_k with known true class. At any non-terminal node, let Ω^n be the set of M^n classes still possible for a unit at node n. Let $U^n = \{u_k^n : k = 1, \ldots, N^n\}$ be the subset of N^n training units associated with node n. If the number of units for class c in node n is denoted by N_c^n, we must have $N^n = \sum_{c=1}^{M^n} N_c^n$.

Now we describe how the decision rule works at node n. Consider unit u_k^n which has measurement vector x_k^n. If the discriminant function $f(x_k^n)$ is less than or equal to a threshold, then u_k^n is assigned to class Ω^n_{left}, otherwise it is assigned to class Ω^n_{right}. An assignment to Ω^n_{left} means that a unit descends to the left child node and an assignment to Ω^n_{right} can be understood in a similar way. Given a discriminant function f, the units in U^n are sorted in such a way that $f(x_k^n) \leqslant f(x_{k+1}^n)$ for $k = 1, \ldots, N^n - 1$. Let w_k^n be the true classes associated with the measurement vectors x_k^n. Then a set of candidate thresholds T^n for the decision rules is defined by

$$T^n = \left\{ \frac{f(x_{k+1}^n) - f(x_k^n)}{2} \middle| w_{k+1}^n \neq w_k^n \right\}. \tag{55}$$

For each threshold value, unit u_k^n is classified by using the decision rule specified above. We count the number of samples n_{Lc}^t assigned to Ω^n_{left} whose true class is c and we count the number of samples n_{Rc}^t assigned to Ω^n_{right} whose true class is c, that is,

$$n_{Lc}^t = \#\{u_k^n \mid f(x_k^n) \leqslant t \text{ and } w_k^n = c\},$$

$$n_{Rc}^t = \#\{u_k^n \mid f(x_k^n) > t \text{ and } w_k^n = c\}.$$

Let n_L^t be the total number of samples assigned to Ω^n_{left} and n_R^t be the total number of samples assigned to Ω^n_{right}, that is,

$$n_L^t = \sum_{c=1}^{M^n} n_{Lc}^t \quad \text{and} \quad n_R^t = \sum_{c=1}^{M^n} n_{Rc}^t.$$

We define the purity PR_n^t of the assignment made by node n to be

$$PR_n^t = \sum_{c=1}^{M^n} \left(n_{Lc}^t \log \frac{n_{Lc}^t}{n_L^t} + n_{Rc}^t \log \frac{n_{Rc}^t}{n_R^t} \right). \tag{56}$$

The discriminant threshold t is chosen such that it maximizes the purity value PR_n^t. The purity is such that it gives a maximum value when the training samples are completely separable. The expansion of the tree is stopped if the decision rules used at the nodes cannot be applied to smaller training sets or if the number of training feature vectors at the node is smaller than a predetermined value.

In the simplest form of a linear decision rule (f is linear), one of the components of the measurement vector is taken and a set of candidate thresholds, T, are calculated for that component. The one that gives the maximum purity is chosen. This process is repeated for all the components in the measurement vector. From the thresholds computed for all the components in the measurement vector, the one that yields maximum purity is chosen.

When a feature vector is input to a decision tree, a decision is made at every non-terminal node as to what path the feature vector will take. This process is continued until the feature vector reaches a terminal node of the tree, where a class is assigned to it.

5.2. Eliminating data over-fitting in decision tree classifier

In building a decision tree classifier, there is a risk of memorizing the training data, in the sense that nodes near the bottom of the tree represent the noise in the sample. As mentioned in [5,6], some methods were employed to make better class probability, such as building multiple trees and use the benefits of averaging, approximate significance tests, etc. In practice, we use one data set to build the tree and use another independent data set to prune the tree to reduce the possibilities of data over-fitting in the trained decision tree. We used two simple methods to prune the decision tree.

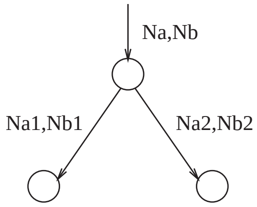


Fig. 7. Illustrates a decision tree node with its child nodes.

In Fig. 7, there is a node with its two child nodes. N_a, N_b are the number of class A vectors and class B vectors which arrive at this node. Similarly, N_{a1} and N_{b1}, N_{a2} and N_{b2} are the number of class A vectors and class B vectors which arrive at its left child node and its right node, respectively. We can compute two different error probabilities associated with this node.

- *Inherent error rate*. It is the error probability of setting this node as a leaf node,

$$\frac{\min(N_a, N_b)}{N_a + N_b};$$

- *Effective error rate*. It is the error probability of expanding this node with two children nodes,

$$\frac{\min(N_{a1}, N_{b1}) + \min(N_{a2}, N_{b2})}{N_a + N_b}.$$

The probability ratio is

$$e_1 = \frac{\min(N_{a1}, N_{b1}) + \min(N_{a2}, N_{b2})}{\min(N_a, N_b)}.$$

The condition is that if $e_1 \geqslant \theta$, where θ is a given threshold, we will make this node as a leaf node, otherwise, we keep its two child nodes.

Another constraint condition is that we will stop expanding this node if the probability of an arbitrary separation of vector numbers is less than a threshold. The probability can be computed as

$$e_2 = \frac{C_{N_a}^{N_{a1}} C_{N_b}^{N_{b1}}}{C_{N_a+N_b}^{N_{a1}+N_{b1}}} = \frac{C_{N_a}^{N_{a2}} C_{N_b}^{N_{b2}}}{C_{N_a+N_b}^{N_{a2}+N_{b2}}},$$

where $C_{N_a}^{N_{a1}}$ is the combination number of the elements of N_a taken N_{a1} at a time. If $e_2 < \delta$, where δ is a given threshold, we will stop expanding this node and make it as a leaf node. The class label associated with the leaf node is the class associated with more training vectors at the node than any other class.

6. Finding the optimal sequence in $\mathscr{S}_2$ by HMM

To further improve the zone classification result, we make use of context constraint in some zone set, $\mathscr{S}_2$. We model context constraint as a Markov Chain. Let s be element number in $\mathscr{S}_2$. Let $Z = (Z_1, Z_2, \ldots, Z_s)$, where $Z_t \in \mathscr{S}_2, t = 1, \ldots, s$, be a zone sequence. We use a Markov chain model to represent the context constraint of the sequence of zone type, $f = f(Z_1), f(Z_2), \ldots, f(Z_s)$. The probability $P(f|Z)$ requires the zone type of the current block, as well as the zone types of all the predecessor blocks.

$$P(f|Z) = \Pi_{Z_i \in Z} P(f(Z_i)|f(Z_{i-1}), f(Z_{i-2}), \ldots, f(Z_1)), \tag{57}$$

where $P(f(Z_i)|f(Z_{i-1}), f(Z_{i-2}), \ldots, f(Z_1))$ is the probability of $f(Z_i)$, under the condition that the zone types of the previous zone are $f(Z_{i-1}), f(Z_{i-2}), \ldots, f(Z_1)$. For a first-order Markov chain, the current state is only dependent on the previous state, i.e.

$$P(f(Z_i)|f(Z_{i-1}), f(Z_{i-2}), \ldots, f(Z_1)) = P(f(Z_i)|f(Z_{i-1})). \tag{58}$$

Given this conditional independence assumption, the probability in Eq. (57) is simplified as

$$P(f|Z) = \Pi_{Z_i \in Z} P(f(Z_i)|f(Z_{i-1})). \tag{59}$$

Therefore, the zone class probability with the context constraint

$$\Pi_{Z_i \in Z} P(V(Z_i)|f(Z_i)) P(f(Z)) \tag{60}$$

is a simple HMM, shown in Fig. 8. Rabiner [7] has a good tutorial on the HMM and the reader may be referred to it. In our problem, the hidden state are the zone content class $f(Z)$ which are not observable. $P(f(Z_i)|P(Z_{i-1})), Z_i \in Z$ are called *transition probabilities*, and $P(f(Z_1))$ are the *initial probabilities*. A sequence of observations $V(Z)$ is measured at each element in Z where $P(V(Z_i)|f(Z_i))$ are observation probabilities.

We use Viterbi algorithm [7] to find the most likely state sequence $f(Z^*)$, which can be used as the optimal solution to Eq. (60). The recursion formula for computing the highest probability along a single path, which ends at Z_i with the zone content class $f(Z_i) = j$, $j = 1, \ldots, 9$, where $j = 1, \ldots, 9$ stands for the 9 possible zone content class, is the following,

$$\delta_i(j) = \begin{cases} P(f(Z_1) = j)P(V(Z_1)|P(B_1 = j)), & i = 1, \\ \max_t \delta_{i-1} P(f(Z_i) = j|f(Z_{i-1} = t))P(V(Z_i)|f(Z_i)), & i > 1, \end{cases} \tag{61}$$

where t and j are the possible zone content types.

The algorithm is an application of dynamic programming for finding a maximum probability path in a directed graph. We use array $\Psi_i(j)$ to keep track of the argument which

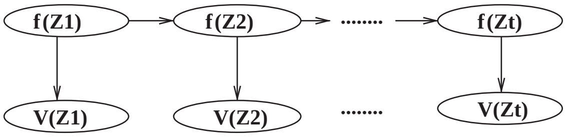


Fig. 8. Illustrates defined hidden Markov model. $f(Z_i)$, $i = 1, \ldots, t$, represents the label on each zone and $V(Z_i)$, $i = 1, \ldots, t$ represent the measurement on each zone.

maximize the probability in Eq. (61):

$$\Psi_i(j) = \arg\max_t [\delta_{i-1} P(f(Z_i) = j | f(B_{i-1} = t)]. \qquad (62)$$

At the end of the sequence, a backtracking step is performed to retrieve the most probable path of zone content classification.

To apply the Viterbi algorithm [7], three types of parameters have to be estimated: state transition probability matrix, state observation probability matrix, and initial state distribution vector. The state observation probability matrix and initial state distribution vector are estimated from the groundtruth data in the training data set. The state observation probability matrix are just the probability that each given zone belongs to each class. This probability is readily estimated from the training data set by decision tree structure. Suppose example x falls to leaf l in the tree structure T. We have C mutually exclusive and exhaustive classes, $d_1, \ldots, d_C$. A vector ϕ_l is associated with the leaf node l. Its elements are the proportion of the number of class d_i training samples over the number of total training samples falling to leaf l. We can compute the probability that x belongs to each class by

$$P(c = d_j | x, T) = \phi_{l,j}, \quad j = 1, \ldots, C.$$

The reason for us to use HMM on some set of zones and not on the whole zone set is twofold. First, although extracting the reading order from a given page is still an open problem, getting the header and footer parts are much easier. Second, some zone sequences are similar, e.g. table, map/drawing and halftone zones all tend to be followed by text zones. In a small set of zones, not restricted to the header and footer parts, the sequence gives us more reliable information about the zone class.

7. Experiments and results

7.1. Performance evaluation protocol

A hold-out method is used for the error estimation in our experiment. We divided the data set into 9 parts. We trained the decision tree on the first 4 parts, pruned the tree using another 4 parts and then tested on the last 1 part. To train the Markov model, we trained on the first 8 parts and tested it on the last 1 part. We estimate the state transition probability and initial state distribution vector are estimated from the first 8 parts and the state observation probability matrix are estimated from decision tree classifier. We continue this procedure, each time omitting one part from the training data and then testing on the omitted part. Then the combined 9 part results are put together to estimate the total error rate [4].

Table 1
Possible true- and detected-state combination for two classes

True class	Assigned class	
	a	b
a	P_{aa}	P_{ab}
b	P_{ba}	P_{bb}

The output of the decision tree is compared with the zone labels from the ground truth in order to evaluate the performance of the algorithm. A contingency table is computed to indicate the number of zones of a particular class label that are identified as members of one of nine classes. The rows of the contingency table represent the true classes and the columns represent the assigned classes. The cell at row r and column c is the number of zones whose true class is r while its assigned class is c.

We also compute four rates here: *Correct Recognition Rate* (*CR*), *Mis-recognition Rate* (*MR*), *False Alarm Rate* (*FR*), *Accuracy Rate* (*AR*). Suppose we only have two classes: a and b. The possible true- and detected-state combination is shown in Table 1. We compute four rates for class a as follows:

$$CR = \frac{P_{aa}}{P_{aa} + P_{ab}}, \quad MR = \frac{P_{ab}}{P_{aa} + P_{ab}},$$

$$FR = \frac{P_{ba}}{P_{ba} + P_{bb}}, \quad AR = \frac{P_{aa} + P_{bb}}{P_{aa} + P_{ab} + P_{bb} + P_{ba}}.$$

Using this performance evaluation protocol, we did extensive experiments with different conditions. In our experiment, the training and testing data set was drawn from the scientific document pages in the University of Washington document image database III [8]. It has 1600 scientific and technical document pages with a total of 24,177 zones. The class labels for each of the zones are obtained from the database. The header and footer parts are obtained from the database while they are readily automatically segmented. These zones belonged to nine different classes. Two text classes of font size 4–18 pt and font size 19–32 pt), math, table, halftone, map/drawing, ruling, logo and others. Below are the results of these experiments.

7.2. Experimental results with 69 features without optimized decision tree and HMM

If we assume conditional independence between the zone classifications, the probability in Eq. (1) may be

Table 2
Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class in UWCDROM-III

	$T1$	$T2$	M	T	H	M/D	R	L	O	CR (%)	MR (%)
$T1$	21,416	18	55	7	3	6	1	0	3	99.57	0.43
$T2$	16	99	3	0	5	1	2	0	1	77.95	22.05
M	174	0	564	3	0	15	2	0	0	74.41	25.59
T	17	0	2	151	0	37	2	0	0	72.25	27.75
H	5	0	0	0	289	32	0	1	11	85.50	14.50
M/D	30	0	19	17	19	632	0	0	3	87.78	12.22
R	9	0	0	0	0	2	421	0	1	97.23	2.77
L	5	1	2	1	2	1	0	1	0	7.69	92.31
O	11	0	0	0	32	21	0	0	6	8.57	91.43
FR (%)	10.01	0.08	0.35	0.12	0.26	0.49	0.03	0.00	0.08		

The algorithm used 69 features and assumed conditional independence between the zone classifications. The decision tree classifier was not optimized. In the table, T_1, T_2, M, T, H, MD, R, L, O represent text with font size $\leqslant$ 18 pt, text with font size $\geqslant$ 19 pt, math, table, halftone, map/drawing, ruling, logo, others, respectively.

Table 3
Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class in UWCDROM-III

	$T1$	$T2$	M	T	H	M/D	R	L	O	CR (%)	MR (%)
$T1$	21,446	13	34	10	2	3	2	1	1	99.69	0.31
$T2$	16	106	1	0	2	1	1	0	1	82.81	17.19
M	49	2	683	3	0	17	1	2	1	90.11	9.89
T	9	0	4	159	1	38	1	1	2	73.95	26.05
H	1	2	0	1	369	12	0	1	2	95.10	4.90
M/D	8	0	20	30	18	630	1	1	2	88.73	11.27
R	6	0	2	0	1	3	419	0	0	97.22	2.78
L	5	5	0	1	0	1	1	0	0	0.00	100.00
O	3	2	1	2	4	3	0	0	7	31.82	68.18
FR (%)	3.64	0.10	0.26	0.20	0.12	0.33	0.03	0.02	0.04		

The algorithm used 69 features and modeled context constraints with HMM in some zone set. It used an optimized decision tree classifier. In the table, T_1, T_2, M, T, H, MD, R, L, O represent text with font size $\leqslant$18 pt, text with font size $\geqslant$19 pt, math, table, halftone, map/drawing, ruling, logo, others, respectively.

decomposed as

$$P(f(\mathscr{A})|V(\mathscr{A})) = \prod_{\tau\in\mathscr{A}} P(f(\tau)|V(\tau)). \tag{63}$$

The problem can be solved by maximizing each individual probability $P(f(\tau)|V(\tau))$ in Eq. (63), where $\tau \in \mathscr{A}$.

Based on this idea, we compute a feature vector with 69 features for each zone. Then we applied decision tree classifier in the data. Using our performance evaluation protocol, the experimental result are shown in Table 2. For a total of 24,177 zones, the accuracy rate was 97.53% and the mean false alarm rate was 1.26%.

7.3. Experimental results with 69 features with optimized decision tree and HMM

In this experiment, we used the problem statement as Eq. (2). With 69 features and optimized decision tree and HMM in set $\mathscr{S}_2$. For a total of 24,177 zones, the accuracy rate was 98.52% and mean false alarm rate was 0.53%, as shown in Table 3.

7.4. Experimental results with feature reduction

In this experiment, the feature set only had 25 features with optimized decision tree and HMM in set $\mathscr{S}_2$. For a total of 24,177 zones, the accuracy rate was 98.45% and mean false alarm rate was 0.50%, as shown in Table 4.

7.5. HMM result analysis

Because of the biased data, we applied HMM on header and footer regions instead of the whole page. Of a total of 24,177 zones, there are 21,512 text 1 class zones. The zone numbers of table class, halftone class and map/drawing class are 215, 388 and 710, respectively. Most table, halftone and map/drawing zones are followed by a text zones. When we apply HMM on the whole page, it tends to recognize table zones as map/drawing zones since in the HMM training data set the number of map/drawing zones is far larger than that of table zones. In the header/footer regions, there are only text 1 class zones, text 2 class zones, rule class zones and others class zones. HMM solution gives us about 0.48%

Table 4
Contingency table showing the number of zones of a particular class that are assigned as members of each possible zone class in UWCDROM-III

	$T1$	$T2$	M	T	H	M/D	R	L	O	CR (%)	MR (%)
$T1$	21,426	23	40	7	1	7	1	3	3	99.60	0.40
$T2$	19	104	1	0	1	2	0	0	1	81.25	18.75
M	47	1	686	2	0	18	1	1	2	90.50	9.50
T	6	0	4	162	0	35	0	1	2	77.14	22.86
H	1	0	1	1	345	27	0	0	0	92.00	8.00
M/D	2	3	20	20	28	648	1	1	5	89.01	10.99
R	3	0	2	0	0	2	424	0	1	98.15	1.85
L	7	3	1	0	0	0	0	2	0	15.38	84.62
O	4	0	2	0	2	7	1	0	6	27.27	72.73
FR (%)	3.34	0.12	0.30	0.13	0.13	0.42	0.02	0.02	0.06		

The algorithm used 25 features and modeled context constraints with HMM in some zone set. It used an optimized decision tree classifier. In the table, T_1, T_2, M, T, H, MD, R, L, O represent text with font size $\leqslant 18$ pt, text with font size $\geqslant 19$ pt, math, table, halftone, map/drawing, ruling, logo, others, respectively.

Table 5
Illustrates the performance evaluation results of three experiments

Section	Accuracy rate (%)	False alarm rate (%)
7.2	97.53	1.26
7.3	98.52	0.53
7.4	98.53	0.50

improvement in the 2274 zones of the header and footer regions. Although the improvement is very limited, we claim HMM would give us more improvement if the training data set were more balanced.

7.6. Feature reduction analysis

In our early work, we used a feature vector consisting of 69 features and got very good results. In our recent work, we tried to reduce the unnecessary features from the feature vector while keeping good performance. By analysis and experiments, a total of 44 features were eliminated.

In Table 5, we compared the experimental results from Sections 7.2, 7.3 and 7.4. The identical data set was used in the experiments. Comparing the results, we can have some conclusions.

- The decision tree optimization and context model help us to gain improvement in both accuracy rate and false alarm rate terms.
- The dimension reduction from 69 features to 25 feature does not affect the performance too much. It demonstrates that we have a successful feature reduction.

The advantages for the decision tree optimization and context model are straightforward. In the below paragraphs, we try to give some reflection on our feature reduction work.

As shown in Fig. 2, there were four feature computation directions. Since the images in UWCDROM-III are all deskewed already, there are no strong variations in different directions. We changed the four directions to the current two directions 6. It directly removed 32 features from the feature vector.

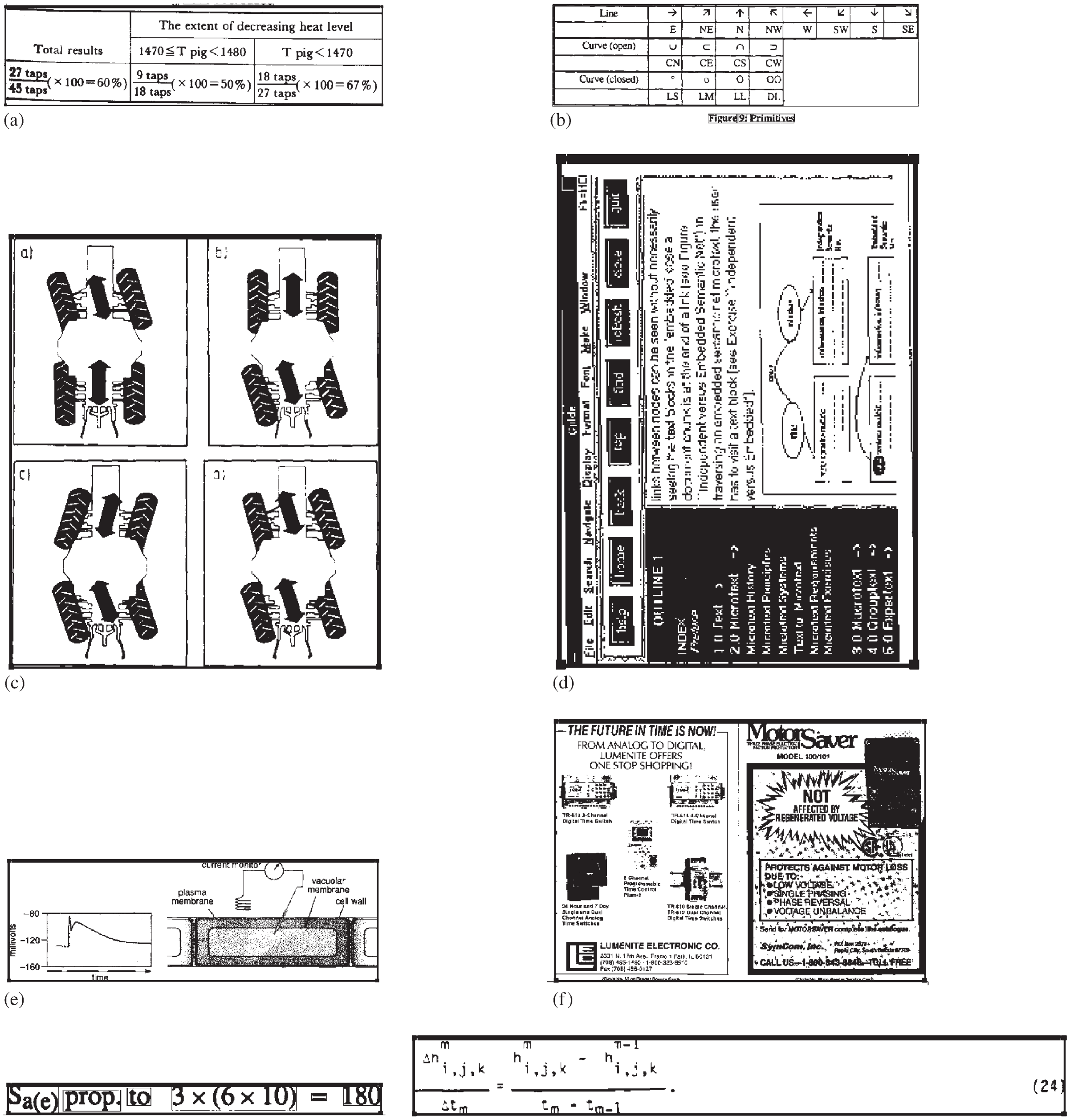
Some features are redundant. For example, there were four background features, background run length number in the two given directions, a fraction of black pixels to the total number of pixels and total area of large horizontal and large vertical blank blocks. Since the feature, total area of large horizontal and large vertical blank blocks, is computed using the other three feature information, we eliminated the other three features. There were two zone area related features, zone bounding box area and a fraction of the number of text glyphs to the zone bounding box area. There are dependent features so we eliminated the first of them.

There were 16 features computed by autocorrelation function. We defined four functions which are computed in two different directions. The features are the slope of the tangent to the autocorrelation function values whose indexes are close 0 and, the index for which the autocorrelation function goes to 10% of its maximum value. By the experiments, we eliminated 8 of them. From the experimental results, we believe our feature deduction was successful.

In our current work, we were mainly motivated to eliminate the dependency between different features. We used analysis and experiments to select 25 features and achieved comparable experimental results. Because our feature vector has a relatively small size, this simple scheme worked fine with us. However, we understand that generally this simple scheme does not work well for feature vector with hundreds of features. In the future work, we believe that it would be very interesting and beneficial to our research if we use some standard statistical techniques, such as AdaBoost [39], to systematically study and select statistically significant features.

7.7. Failed cases study

In Fig. 9, we show some failed cases of our experiment in Section 7.3. Fig. 9(a) is a Table zone misclassified as



Total results	The extent of decreasing heat level	
	1470≦T pig<1480	T pig<1470
$\frac{27\text{ taps}}{45\text{ taps}}(\times 100=60\%)$	$\frac{9\text{ taps}}{18\text{ taps}}(\times 100=50\%)$	$\frac{18\text{ taps}}{27\text{ taps}}(\times 100=67\%)$

(a)

Line	→	↗	↑	↖	←	↙	↓	↘
	E	NE	N	NW	W	SW	S	SE
Curve (open)	∪	⊂	∩	⊃				
	CN	CE	CS	CW				
Curve (closed)	°	o	O	OO				
	LS	LM	LL	DL				

Figure 9: Primitives

(b)

(c)

links between nodes can be seen without necessarily seeing the text blocks in the 'embedded' case a document chunk is at the end of a link (see Figure "Independent versus Embedded Semantic Net") in traversing an embedded semantic net microtext, the user has to visit a text block (see Exercise "Independent versus Embedded").

OFFLINE 1
INDEX
Preface
1.0 Text ->
2.0 Microtext ->
Microtext History
Microtext Principles
Microtext Systems
Text to Microtext
Microtext Representations
Microtext Extensions
3.0 Macrotext ->
4.0 Grouptext ->
6.0 Expertext ->

(d)

(e)

THE FUTURE IN TIME IS NOW!

FROM ANALOG TO DIGITAL, LUMENITE OFFERS ONE STOP SHOPPING!

TR-613 3-Channel Digital Time Switch

TR-614 4-Channel Digital Time Switch

8 Channel Programmable Time Control Plus+8

24 Hour and 7 Day Single and Dual Channel Analog Time Switches

TR-610 Single Channel, TR-612 Dual Channel Digital Time Switches

LUMENITE ELECTRONIC CO.
2331 N. 17th Ave., Franklin Park, IL 60131
(708) 455-1450 · 1-800-323-8510
Fax (708) 455-0127

(Circle No. 18 on Reader Service Card)

MotorSaver
THREE PHASE ELECTRIC MOTOR PROTECTOR
MODEL 100/101

NOT AFFECTED BY REGENERATED VOLTAGE

PROTECTS AGAINST MOTOR LOSS DUE TO:
- LOW VOLTAGE
- SINGLE PHASING
- PHASE REVERSAL
- VOLTAGE UNBALANCE

Send for MOTORSAVER complete line catalogue.

SymCom, Inc., P.O. Box 2570, Rapid City, South Dakota 57709

CALL US—1-800-843-8848—TOLL FREE

(Circle No. 19 on Reader Service Card)

(f)

$S_{a(e)}$ prop. to $3 \times (6 \times 10) = 180$

(g)

$$\frac{\Delta h^{m}_{i,j,k}}{\Delta t_{m}} = \frac{h^{m}_{i,j,k} - h^{m-1}_{i,j,k}}{t_{m} - t_{m-1}}. \qquad (24)$$

(h)

Fig. 9. Illustrates some failed examples. (a) Table zone misclassified as Math zone; (b) Map/drawing zone misclassified as Table zone; (c) Map/drawing zone misclassified as Table zone; (d) Map/drawing zone misclassified as Halftone zone; (e) Halftone zone misclassified as Map/Drawing zone; (f) Others zone misclassified as Halftone zone; (g) Math zone misclassified as Text 1 zone; (f) Math zone misclassified as Table zone.

Math zone due to the presence of many numericals and operators. Fig. 9(b) is a Map/Drawing zone misclassified as Table zone in that the content of the figure is just a table. Fig. 9(c) is another Map/Drawing zone misclassified as Table case. The reason is that the subfigures are put in a two-dimensional grid which looks like a table. Fig. 9(d), (e) show two examples in which a Map/Drawing zone was classified as Halftone zone(d), a Halftone zone was classified as Map/Drawing zone(e). The reason for these two errors is that a Map/Drawing zone sometimes contains halftone-like part and a Halftone zone sometimes contains drawing-like part. Fig. 9(f) is an example in which an others zone was misclassified as Halftone zone. It is actually an advertisement part in the technical paper and it contains halftone-like parts. Fig. 9(g) shows a most frequent error of our current system. Our system classified a Math zone into Text 1 zone class. Sometimes our system lacks ability to detect such a single line math equation zone which may include some description words. Fig. 9(h) shows an error example in which a Math zone was misclassified as a table zone because of its sparse nature.

By checking the experimental results, we also realized some possible limitations of UWCDROM-III database. For instance, Fig. 9(g) can hardly be classified as text or math zone because it is a union of both classes. It may be a good idea to allow a single zone, e.g. the one shown in Fig. 9(g), to have multiple classifications with appropriate weightings in the future document database collection and groundtruthing work.

8. Conclusion and future work

Given the segmented document zones, correctly determining the zone content type is very important for the subsequent processes within any document image understanding system. This paper describes an algorithm for the determination of zone type of a given zone within an input document image. In our zone classification algorithm, zones are represented as feature vectors. Each feature vector consists of a set of measurements of pre-defined properties. We considered two different feature vectors. One was with 69 features and the other had only 25 features. A probabilistic model, decision tree, is used to classify each zone on the basis of its feature vector [4]. Two methods are used to optimize the decision tree classifier to eliminate the data over-fitting problem. To enrich our probabilistic model, we incorporate context constraints for certain zones within their neighboring zones. We also model zone class context constraints as a HMM and used Viterbi algorithm [7] to obtain optimal classification results.

In this paper, the features we designed included run length features, spatial features, autocorrelation features, background features, etc. We classify each zone into one of the nine classes, 2 text classes (of font size 4–18 pt and font size 19–32 pt), math, table, halftone, map/drawing, ruling, logo, and others. We designed two different feature vectors. One is with a total of 69 features and another is with a total of 25 features.

In the experiments, the data set consisted of 1600 UWCDROM-III images with a total of 24,177 zones. We described a performance evaluation protocol for the zone content classification experiments. The cross-validation method was used in the performance evaluation of the three algorithms. The experiments showed that the classification optimization and context model help us gain an improvement in terms of accurate rate and false alarm rate. With the feature reduction from 69 to 25, the accuracy rate and the false alarm rate are similar for the current and the last algorithms. However, since the features used in the current algorithm was reduced from 69 features to 25, the classification speed of the current algorithm was reduced proportionally. Specifically, without considering the feature extraction time, the training, pruning and testing time was reduced more than a half after the reduction of the feature number.

We also showed some failed cases. Many errors are due to the difficult discrimination between single line math and text 1 class. Our future work include the development of math zone identification technique, modeling zone content dependency feature in a more general zone set.

The motivation of our zone content classification research lies in two parts. One part is that we want to find an efficient method to classify different zone types. We have achieved a relatively high accuracy on a standard document database, UWCDROM-III. The future work should go to more difficult document analysis, such as multiple colors of popular magazines. The other part is that we want to take a systematic and complete study of statistical properties of document zones. We firmly believe that the features we learned and the experiments we conducted would be useful for future more challenging document analysis tasks. Certainly, most of modern magazines and publications, such as Time and Wired, are produced electronically and available already labeled on the web. The document analysis researchers should apply the knowledge learned in the traditional research topics to these emerging research topics such as web document analysis [40], hand written document understanding [28] and ancient document archive indexing [41], etc. For practical propose, the expected recognition ratios for these research should be no less than 90%.

References

[1] R. Haralick, Document image understanding: geometric and logical layout, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 1994, pp. 385–390.

[2] G.E. Kopec, P.A. Chou, Document image decoding using markov source models, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 602–617.

[3] J. Hu, R. Kashi, D. Lopresti, G. Wilfong, Evaluating the performance of table processing algorithms, Int. J. Document Anal. Recognition 4 (3) (2002) 140–153.

[4] R. Haralick, L. Shapiro, Computer and Robot Vision, vol. 1, Addison-Wesley, Reading, MA, 1997.

[5] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmont, 1984.

[6] W. Buntine, Learning classification trees, Statist. Comput. Journal (1992) 63–76.

[7] L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proc. IEEE 77 (1989) 257–285.

[8] I. Phillips, Users' reference manual. CD-ROM, UW-III Document Image Database-III, 1995.

[9] Y. Xiao, H. Yan, Text region extraction in a document image based on the delaunay tesselation, Pattern Recognition 36 (3) (2003) 799 –809.

[10] R. Zanibbi, D. Blostein, J.R. Cordy, Recognizing mathematical expressions using tree transformation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1455–1467.

[11] J. Chen, M.K. Leung, Y. Gao, Noisy logo recognition using line segment hausdorff distance, Pattern Recognition 36 (4) (2003) 943 –955.

[12] T.D. Pham, Unconstrained logo detection in document images, Pattern Recognition 36 (12) (2003) 3023–3025.

[13] J. Li, A. Najmi, R.M. Gray, Image classification by a two dimensional hidden markov model, IEEE Trans. Signal Process. 48 (2) (2000) 517–533.

[14] R.P. Futrelle, M. Shao, C. Cieslki, A.E. Grimes, Extraction, layout analysis and classification of diagrams in pdf documents, in: 7th International Conference on Document Analysis and Recognition (ICDAR'03), 2003, pp. 1007–1014.

[15] G. Nagy, Twenty years of document image analysis in pami, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 38–62.

[16] J. Liang, I.T. Phillips, R.M. Haralick, A methodology for document image structures extraction, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 719–734.

[17] F. Esposito, D. Malerba, G. Semeraro, E. Annese, G. Scafuro, An experimental page layout recognition system for office document automatic classification: an integrated approach for inductive generalization, in: Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, 1990, pp. 557–562.

[18] F. Esposito, D. Malerba, G. Semeraro, Classification in noisy environments using a distance measure between structural symbolic descriptions, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 390 –402.

[19] F. Cesarini, M. Lastri, S. Marinai, G. Soda, Encoding of modified *x–y* trees for document classification, in: Sixth International Conference on Document Analysis and Recognition ICDAR01, Seattle, WA, September 2001, pp. 1131–1135.

[20] R. Sivaramakrishnan, I. Phillips, J. Ha, S. Subramanium, R. Haralick, Zone classification in a document using the method of feature vector generation, in: Proceedings of the Third International Conference on Document and Analysis and Recognition (ICDAR'95), Montreal, Canada, August 1995, pp. 541–544.

[21] I. Phillips, S. Chen, R. Haralick, Cd-rom document database standard, in: Second International Conference on Document Analysis and Recognition (ICDAR'93), Tsukuba Science City, Japan, October 1993, pp. 478–483.

[22] M. Krishnamoorthy, G. Nagy, S. Seth, M. Viswanathan, Syntactic segmentation and labeling of digitized pages, IEEE Trans. Pattern Anal. Mach. Intell. 15 (7) (1993) 737–747.

[23] A. Antonacopoulos, R.T. Ritchings, Representation and classification of complex-shaped printed regions using white tiles, in: Third International Conference on Document Analysis and Recognition (ICDAR'95), Montréal, Canada, August 1995, pp. 1132–1135.

[24] K.C. Fan, L.S. Wang, Classification of document blocks using density features and connectivity histogram, Pattern Recognition Lett. 16 (1995) 955–962.

[25] A.K. Jain, B. Yu, Document representation and its application to page decomposition, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 294–307.

[26] D.X. Le, J. Kim, G. Pearson, G.R. Thom, Automated labeling of zones from scanned documents, Proceedings SDIUT99, 1999, pp. 219–226.

[27] K. Lee, Y. Choy, S. Cho, Geometric structure analysis of document images: a knowledge-based approach, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1224–1240.

[28] A.K. Jain, A.M. Namboodiri, J. Subrahmonia, Structure in on-line documents, in: Sixth International Conference on Document Analysis and Recognition (ICDAR01), Seattle, WA, September 2001, pp. 844 –848.

[29] P.E. Mitchell, H. Yan, Newspaper document analysis featuring connected line segmentation, in: Sixth International Conference on Document Analysis and Recognition (ICDAR01), Seattle, WA, September 2001, pp. 1181–1185.

[30] B. Gatos, S.L. Mantzaris, A. Antonacopoulos, First international newspaper segmentation contest, in: Sixth International Conference on Document Analysis and Recognition (ICDAR01), Seattle, WA, September 2001, pp. 1190–1194.

[31] G. Harit, S. Chaudhury, P. Gupta, N. Vohra, S.D. Joshi, A model guided document image analysis scheme, in: Sixth International Conference on Document Analysis and Recognition (ICDAR01), Seattle, WA, September 2001, pp. 1137–1141.

[32] I. Bazzi, R. Schwartz, J. Makhoul, An omnifont open-vocabulary ocr system for english and arabic, IEEE Trans. Pattern Anal. Mach. Intell. 21 (6) (1999) 495–504.

[33] R. Haralick, Decision making in context, IEEE Trans. Pattern Anal. Mach. Intell. 5 (4) (1983) 417–428.

[34] H.S. Baird, Background structure in document images, Document Image Analysis, 1994, pp. 17–34.

[35] A. Antonacopoulos, Page segmentation using the description of the background, Computer Vision and Image Understanding, June 1998, pp. 350–369.

[36] R.C. Gonzalez, G.R.E. Woods, Digital Image Processing, Addison-Wesley, Reading, MA, USA, 1992.

[37] A.V. Oppenheim, R.W. Schafer, Discrete-Time Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[38] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes in C, Cambridge University Press, Cambridge, 1988.

[39] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2) (2004) 137–154.

[40] A. Antonacopoulos, J. Hu (eds.), Web Document Analysis: Challenges and Opportunities, World Scientific Publishing Company, November 2003.

[41] B. Couasnon, I. Leplumey, A generic recognition system for making archives documents accessible to public, in: Proceedings of the Third International Conference on Document and Analysis and Recognition (ICDAR'03), Edinburgh, August 2003, pp. 228–232.

About the Author—YALIN WANG received his PhD degree from the University of Washington in 2002, in electrical engineering. His research interests include pattern recognition, document image analysis, medical imaging, computer graphics. He was the best student paper award winner in "Fifth IAPR DAS" and the best paper award winner in CCCT'03.

About the Author—IHSIN T. PHILLIPS received her PhD degree from the University of Maryland, College Park, in 1984 in computer science. Her research areas include image processing, pattern recognition, document image understanding, document image database design, and performance evaluation of document image analysis, and recognition systems. She is currently the chairperson of the IAPR technical committee on performance evaluation.

About the Author—ROBERT M. HARALICK is a distinguished professor in the Department of Computer Science, Graduate Center, City University of New York. He is responsible for developing the gray scale cooccurrence texture analysis technique and the facet model technique for image processing. He is a fellow of IEEE for his contributions in computer vision and image processing and a fellow of IAPR for his contributions in pattern recognition, image processing, and for service to IAPR.