Graph-based Layout Analysis for PDF Documents

Canhui Xu[abc*] , Zhi Tang[ab] , Xin Tao[a], Yun Li[ab] and Cao Shi[a]
[a] Institute of Computer Science and Technology, Peking University, Beijing 100871;
[b] State Key Laboratory of Digital Publishing Technology (Peking University Founder Group);
[c] Postdoctoral Workstation of the Zhongguancun Haidian Science Park

ABSTRACT

To increase the flexibility and enrich the reading experience of e-book on small portable screens, a graph based method is proposed to perform layout analysis on Portable Document Format (PDF) documents. Digital born document has its inherent advantages like representing texts and fractional images in explicit form, which can be straightforwardly exploited. To integrate traditional image-based document analysis and the inherent meta-data provided by PDF parser, the page primitives including text, image and path elements are processed to produce text and non text layer for respective analysis. Graph-based method is developed in superpixel representation level, and page text elements corresponding to vertices are used to construct an undirected graph. Euclidean distance between adjacent vertices is applied in a top-down manner to cut the graph tree formed by Kruskal's algorithm. And edge orientation is then used in a bottom-up manner to extract text lines from each sub tree. On the other hand, non-textual objects are segmented by connected component analysis. For each segmented text and non-text composite, a 13-dimensional feature vector is extracted for labelling purpose. The experimental results on selected pages from PDF books are presented.

Keywords: Graph-based segmentation, PDF document analysis, image based document analysis

1. INTRODUCTION

Fixed layout documents and reflowable documents are two categories for digital documents. It is known that the necessity of object structure extraction in PDF documents and the usage of XML format are profitable for indexing, searching and other applications[1]. Successful conversion of fixed layout documents such as PDF to reflowable format highly depends on the accuracy of physical and logical structure extraction. The premise in the first place is reliable layout analysis. Specifically, to increase the flexibility of e-book reading on small portable screens, various researches on layout analysis of PDF format documents were launched[2,3].

It is insufficient to consider layout analysis as a solved problem for conversion of PDF fixed layout documents to reflowable documents, although layout analysis is a well researched domain in past decades for document image analysis. Layout analysis is generally divided into three categories: top-down (knowledge based), bottom-up (data-driven) and hybrid methods [4]. Image document layout analysis segments document image into homogenous geometric regions by using features like proximity, texture or whitespace. Many researches applied the connect components (CCs) of page images. The "docstrum" method [5] exploited the k nearest-neighbor pairs between connect component centers by using features like distance and angle. Kise [6] pointed out that connected components of black pixels can be utilized as primitives so as to simplify the task of page segmentation by combining connected components appropriately. It performed page segmentation based on area Voronoi diagrams by using distance and area ratio of connected components for deleting superfluous edges. Simon [7] proposed a new bottom-up method based on Kruskal's algorithm to classify the text and graph. Xiao [8] utilized a Delaunay triangulation on the point set from the bounded connected components, and describes the page structure by dividing the Delaunay triangles into text area and fragment regions. Ferilli [9] used the distance between connect component borders for bottom-up grouping method. Recently, Koo [10] developed a new approach to assign state estimation on CCs in a document image to perform text block identification and text line extraction. It claims that the limitation of this method suffers from non-textual objects.

The algorithms usually deal with scanned images which are of poor quality. The structure extraction result is subjected to pre-processing steps like noise removal and de-skewing. Comparatively, digital born document based analysis can utilize the inherent meta-data information like text, graphic elements or path operations, and it handle high quality images noise free and un-skewed. Recently, there were attempts in extracting data from PDF documents and

[*] Canhui Xu: E-mail: ccxu09@yeah.net, Telephone: 86-010-82179508

convert it into XML format. Bloethe [11] pointed out that traditional document analysis algorithms performed on PDF documents will drastically improve PDF's content extraction, which will provide convenience to develop high-level applications. And it proposed an X-ed tool to extract text, images and graphics from a PDF document. Chao [12] proposed a method to identify and extract graphic illustrations for PDF documents, which is based on the proximity of page elements.

Based on fixed layout documents represented by PDF documents, this paper focus on the extraction of physical layout structure for further logical structure identification for the conversion to reflowable documents. It involves digital document of various layouts, like multi-column documents, horizontal and vertical text line mixed documents, irregular graphic embedded documents or other complex layouts. This work aims to prepare the segmented geometrical block inputs for logical structure recovery procedure. The basic measurement unit of body text is text line, which will be passed to logical structure recognition of logical labels like paragraph, title or possible subtitle. Text and non text are handled separately for layout analysis. Well researched image-based documents analysis is extended for non-text analysis in PDF documents. Synthesized images are used to facilitate graphic composite segmentation. A graph-based algorithm is proposed and its application on PDF layout analysis is presented. The preprocessing step and the graph based method are presented in Section 2 and 3. Its application on PDF sources is presented at section 4. The conclusion and future work is given in Section 5.

2. CONTENT ANALYSIS AND PREPROCESSING

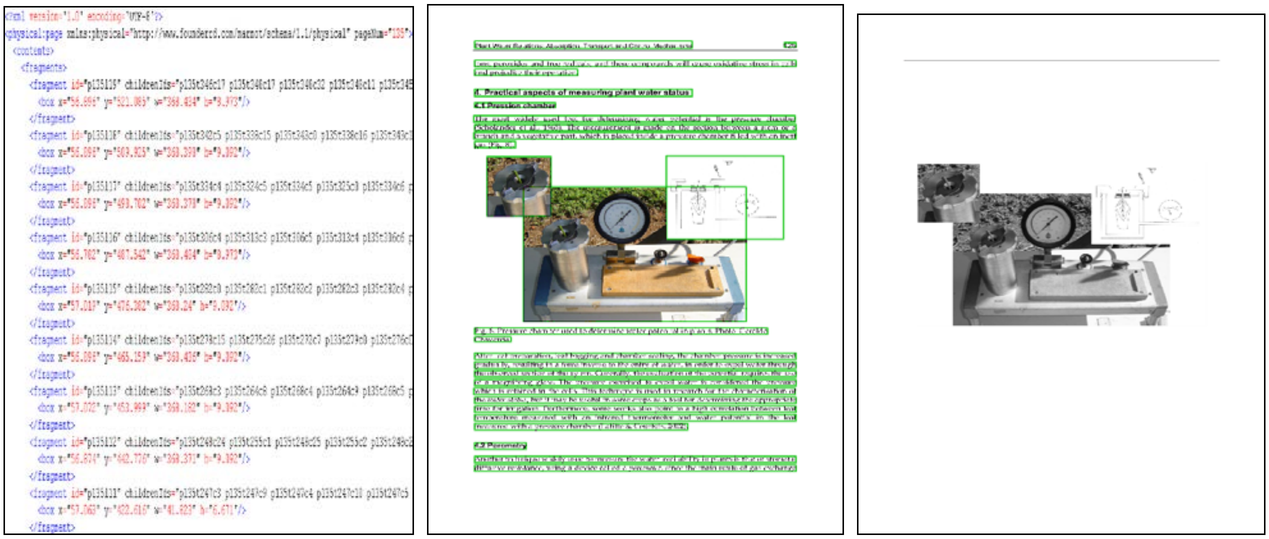
PDF document content stream includes low level page elements, such as text, image and path elements etc. Especially, path consists of path operations and operands, which are used for constituting vector graphic objects, drawings and paintings by lines, curves and rectangles. These low level elements, also called as primitives are the basic building units to be clustered into physical layout blocks, which are inputs of final logical label identification process. In preprocessing, all the text elements and its attributes including font family, font size, color, and bounding box can be extracted from the PDF parser engine provided by Founder Corporation [13]. The centroids of each bounding box are marked out and used for text elements clustering. As is also pointed out in reference [14], the height of the bounding boxes for the text elements fluctuates within a single text line due to the variety of element size, like letter and punctuation. However, the centroids for elements within a line fluctuate little along the text line direction. Our goal is to group the text elements physically according to desired scope like text line level, which are used for further logical labeling process. On the other hand, PDF content stream doesn't provide figure illustration as a whole object. Path elements are often just a fraction of the illustrations, e.g. a line in a drawing. There are two possible schemes to accomplish graphic identification for PDF document pages. Generally, the geometric features of element bounding boxes directly extracted from PDF path and image stream are used for grouping elements into desired physical segments. Another alternative is to use the well researched image based segmentation methods. In our work, paths and image elements are peeled off as non-text elements, which are selected to produce the synthetic non-text image for subsequent traditional document image segmentation.

Thus, all the enriched information considered in our system includes peeled text-only PDF content stream as well as the visual appearance of non-text page elements represented by rendered synthetic images. Hence, regarding each PDF document page, there are three types of input files: xml description, synthesized image and labeled ground truth.

Raw xml contains the basic page elements of parsed text, image and graph content streams and their associated attributes. The results of physical layout analysis and logical layout understanding are designed to be outputted as xml files, respectively. Each physical label can record its children with the unique IDs in page elements. For example, a segment is labeled as "fragment" with a physical ID, and it can provide the coordinates of text line bounding box and all the children IDs from raw text elements belonging to this fragment. Logical label produces similar xml description and reflects the hieratical tree structure relation in document analysis and recognition. Figure1 (a) illustrated the physical xml description of physical segments within the page Figure 1 (b).

A .png synthetic image with resolution of 300 dpi can be rendered though selecting desired page elements. As is shown in Figure 1 (c), all the text elements in pages are covered with white pixels and the left non-text page objects only contains graphic parts, lines and other decorations, etc.

Labeled ground-truth serves the purpose of performance evaluation. Our self-developed ground-truthing tool named "Marmot" can accomplish manual labeling and performance evaluation. It is a GUI application based on wxphython, which is applied for labeling experimental data in references[13,15]. The updated version of ground-truthing tool is applied in this paper. A great difference is that the evaluation is based on the variety of PDF content stream sources: text, image and path primitives. Figure 1 (b) is the visualization of manually labeled ground truth in the scale of fragment for physical layout analysis. Upon the fragment level is the block level for physical layout analysis.



```
<?xml version="1.0" encoding="UTF-8"?>
<physical:page xmlns:physical="http://www.founderrd.com/marmot/schema/1.1/physical" pageNum="135">
  <contents>
    <fragments>
      <fragment id="p135119" childrenIds="p135t346c17 p135t346c17 p135t346c32 p135t346c11 p135t345
        <box x="56.896" y="521.083" w="368.434" h="9.373"/>
      </fragment>
      <fragment id="p135118" childrenIds="p135t342c5 p135t338c15 p135t343c0 p135t338c16 p135t343c1
        <box x="56.098" y="503.925" w="368.399" h="9.392"/>
      </fragment>
      <fragment id="p135117" childrenIds="p135t334c4 p135t324c5 p135t334c5 p135t325c3 p135t334c6 p
        <box x="56.098" y="493.702" w="368.378" h="9.392"/>
      </fragment>
      <fragment id="p135116" childrenIds="p135t306c4 p135t313c3 p135t306c5 p135t313c4 p135t316c6 p
        <box x="56.702" y="487.542" w="368.404" h="9.373"/>
      </fragment>
      <fragment id="p135115" childrenIds="p135t282c0 p135t282c1 p135t282c2 p135t282c3 p135t282c4 p
        <box x="57.017" y="476.382" w="368.24" h="9.092"/>
      </fragment>
      <fragment id="p135114" childrenIds="p135t278c15 p135t275c26 p135t272c7 p135t279c0 p135t276c
        <box x="56.098" y="465.159" w="368.435" h="9.392"/>
      </fragment>
      <fragment id="p135113" childrenIds="p135t268c3 p135t264c8 p135t268c4 p135t264c9 p135t268c5 p
        <box x="57.072" y="453.999" w="368.182" h="9.392"/>
      </fragment>
      <fragment id="p135112" childrenIds="p135t248c24 p135t255c1 p135t248c25 p135t255c2 p135t248c2
        <box x="56.674" y="442.776" w="368.371" h="9.392"/>
      </fragment>
      <fragment id="p135111" childrenIds="p135t247c3 p135t247c9 p135t247c4 p135t247c10 p135t247c5
        <box x="57.063" y="422.616" w="41.823" h="6.671"/>
      </fragment>
```

(a) (b) (c)

Figure 1. Examples of the description for documents: (a) a physical xml description of page segments and its contained children elements from raw xml; (b) a document page with labeled ground-truth in the scale of fragment; (c) non-text layer synthesized image.

To construct the synthesized image, the bounding boxes of all the page elements embedded by PDF in raw xml files are exported and then converted from the metric of logic units to image pixels. The original page of Figure 1 (b) is from a PDF document crawled from web. Figure 1 (c) contains not only the image content stream source, including 122 image primitives from raw xml file, but also the path content stream source, having 15 path primitives from raw xml file. According human vision, however, there are only two graphic composites within this page: a decorative straight line and an illustrative figure. In most cases, the graphics are made from small image primitives and various path operations. There are also other complex cases for illustration composite extraction, especially when it consists of mixed path, image and text content streams altogether.

3. PROPOSED METHOD

Layout analysis is carried out on the inputs prepared previously. To be more specific, two layers are processed separately, and afterwards the results from both two layers are integrated as final physical segments. On text layer, PDF document primitives are directly analyzed. On non-text layer, traditional image analysis is performed on the synthesized image.

3.1 Neighborhood Graph Construction for Text Primitives

For all extracted text primitives, a hybrid method is proposed for text elements clustering in this section. Among our previous works, page columns based on whitespace analysis approach[16] is detected in the first place, after which the text lines are clustered. Bottom-up method is applied to merge text elements into text line only according to spatial distances, and then text lines grow into text blocks by features similarity including font size, inter text line spacing, left alignment and right alignment[11,17]. For regular typesetting with strict rules, the text blocks within the body text region can be successfully detected, but it suffers at the existence of both horizontal and vertical direction reading order, also in some cases illustrative text segments can be mistakenly identified as body text line.

Graph-based method [18] developed can capture certain perceptually important non-local image characteristics for segmentation purposes. In [10], the connect component (CCs) are used as graph vertices. Unlike its application on image segmentation in pixel level or CC state, in this paper, page elements corresponding to vertices are constructed in the graph. Then all the text elements can be connected by establishing a neighbourhood system. Delaunay tessellation is applied in this regard. It is a convenient and powerful neighbourhood representation of 2D image. An undirected graph can be defined as $G=(V,E)$, vertex set is V and $(v_i,v_j)\in E$ are the edges connecting two vertexes. The dissimilarity between adjacent elements v_i and v_j is measured as weights $w(v_{i,}v_j)$ for each edge $(v_i,v_j)\in E$ constructed. In this application, the elements in V are the centroids of the bounding boxes extracted from PDF parser.

In this graph based method, the edge causing the grouping of two components is exactly the minimum weight edge between the components. That implies the edges causing merges are exactly the edges that would be selected by Kruskal's algorithm for constructing the minimum spanning tree T_{MST} of each component [18]. The computational complexity is reduced by path-compression. The input is a page graph $G=(V,E)$ with n vertices and m edges. And the total weight is minimized among all other possible spanning trees of the same graph. In this paper, the weight for each edge $(v_i,v_j)\in E$ uses Euclidean distance function $f_E(v_i,v_j)$:

$$f_E(v_i,v_j)=\left[(v_i(x)-v_j(x))^2+(v_i(y)-v_j(y))^2\right]^{1/2} \tag{1}$$

The layout analysis tasks of text block and text line extraction evolve into the problem of segmenting a graph tree formed by MST of the superpixel representation of each page element into subgraphs by cutting unqualified edges connecting the CCs. Our MST cutting algorithm is as follows:

1) Sort all the edges $e_k\in E$ in an increasing order, and calculate the mean value and standard variance value;

$$M=\frac{1}{n-1}\sum_{k=1}^{n-1}w(e_k) \tag{2}$$

$$V=\frac{1}{n-1}\sum_{k=1}^{n-1}[w(e_k)-M]^2 \tag{3}$$

2) Set cutting threshold value $\theta=kV$, where k is the adjusting parameter. In the application of block segmentation it is set as 1;

3) Remove the edges e_c satisfying $(w(e_c)-M)>\theta$ from T_{MST}. Then, $\{v_i\}_{i=1}^n$ are partitioned into various connected sub trees B_c.

Euclidean distance is applied in this cutting process to segment the text elements into different blocks for their visual proximity. Based on the results of top down block segmentation, a bottom-up text line extracting is carried out. In each block B_c, the orientation of each edge within this sub tree is calculate as:

$$f_A(v_i,v_j)=\left[\tan^{-1}\frac{\Delta y_{i,j}}{\Delta x_{i,j}}\right]_{180^\circ} \tag{4}$$

where $\Delta x_{i,j}=\left|v_j(x)-v_i(x)\right|$, $\Delta y_{i,j}=\left|v_j(y)-v_i(y)\right|$, $[\cdot]_{180^\circ}$ indicates $0\le f_A(v_i,v_j)\le 180^\circ$. In Docstrum[5], nearest neighbour angle histogram and distance histogram of K-nearest units are used to estimate the text line orientation, interline spacing and within-line spacing. The drawback is that the estimation didn't consider the varying spatial cases and the same parameters are applied over the whole document image. From any incoming unknown PDF documents, the page segmentation parameters can change greatly due to the variety of typesetting rules. Interline spacing and text line orientations are two properties, which should be estimated dynamically through the statistics distribution of the connected components within an unconstrained page document. Our method follows similar procedure to identify text lines within each block B_c. According to the statistics of angle distribution calculated from formula (4), the sub tree of block B_c are cut further when there were multiple text lines within one block, and the edges with angle connecting successive text lines are removed from the sub tree. Thus, interline spacing can be calculated. As for the cases when inter word spacing is greater than interline spacing, it results in the existence of fragmentation of a text line, which are merged according to the width of this block B_c and the similar height or width of vertex centroids.

3.2 Non-textual objects Segmentation

Non-textual objects are usually rejected or wide out for document image layout analyses. In Ref [10], it is assumed that non-textual objects are spatially distant from text blocks. Thereafter, non-textual objects are rejected using the properties of clusters within the constructed Delaunay tessellation neighborhood system, and only clusters in text region are considered for layout segmentation. In our work, non-textual objects are not isolated from the layout analysis, since it plays a significant role in the application of reflowable reconstruction of PDF document structure.

There exist two possible schemes to segment identification for PDF document pages. Generally, the geometric features of element bounding boxes directly extracted from PDF path and image stream are used for grouping elements into desired physical segments, which is used in reference [12]. Although the bounding box is guaranteed to encompass the elements for graphics, it is not necessarily the smallest box enclosing the element, like a bounding box contains the white background which is invisible to viewers. Problems like these will cause serious inaccuracy for graphic segmentation. Furthermore, when the path and images elements for constructing a holistic graphic composite are great in numbers, the grouping processing can be computationally slower.

Another alternative is to use the well researched image based segmentation methods. As a separate layer, non-text objects are processed using traditional image analysis method in this work. Connected component analysis is considered from visual perspective. Spatial closeness of graphic objects is described by local texture features. To detect graphic composite holistically, merging process is necessary. According to the inter text line spacing, thresholds are set for connected components grouping. As for graphics embedded or surrounded by text elements, further post processing of integration is handled.

3.3 SVM classification

After performing layout analysis upon both text and non-text layer, we have the resulting bounding boxes of text line composite objects and graphic composite objects.

Feature engineering becomes crucial for classification and labeling. Three types of features are used to enrich the discriminating capacity, including geometric layout, character and image features, which are listed in Table 1. A 13 dimensional feature vector is extracted for each composite object whether it is from analysis results of text or non text layer. The only difference is that character features of graphic composite objects are set to zeroes, which indicates whether the source of feature extraction comes from different layer can contribute for classification. For both text and non text segments, all the segmented sub images are stored, and image features describing texture spectrum are extracted.

Previous works[15,19] attempt to use Support Vector Machine (SVM) to classify formula and ordinary text for PDF documents. Both isolated and embedded mathematical expressions in PDF documents can be detected by two-class SVM classifier with accuracy over 90%. In this work, a larger variety of class labels are considered. The segments extracted from preceding analysis within document page are divided into physical class labels such as body text, graphic text, page number text, footer text, header text, marginal text, title text, figure, and figure fragment.

To explore the dissimilarity capacity of the provided features, multi-class SVM classifier is used in our labeling task. Radial Basis Function is employed as the kernel function. The classifier is first trained on the labeled data to obtain model parameters, and then the model parameters are utilized for predicting the physical label of the layout segments.

Table 1. Feature selection for the segments.

Feature type	Feature name	Definition	Text line	Graphics
Geometric layout features	Height	The height of a composite object's bounding box	√	√
	Area	The area of a composite object's bounding box	√	√
	W/H ratio	The ratio of width and height	√	√
Character features	V-FontSize	The variance of the font size	√	
	CharNo	The number of character within an object	√	
Image features	BlkPix	The number of black pixel over object area	√	√
	StdPix	The intensity std of the subimage	√	√
	Entropy	The entropy of the subimage	√	√
	Contrast	The contrast of the subimage	√	√

	Correlation	The correlation of the subimage	√	√
	Energy	The energy of the subimage	√	√
	Homogeneity	The homogeneity of the subimage	√	√

4. RESUTLS

Delaunay triangulation is generated by conventional incremental method to construct element neighborhood relation. By imposing the bounding boxes of text elements on the page image, the super-pixel representation can reduce the processing time considerably. The features from all the vertices of triangles are extracted, including Euclidean distance and orientation angle, which are utilized in text layer analysis proposed in section 3.1. According to the method proposed, the minimal spanning tree (MST) of page graph is generated with Kruskal’s algorithm. Generally, it is assumed that inter-line space is larger than the inter-word space in body text area and the inter-character space is smaller than inter-words space for Latin-derived languages. For sign languages like Chinese, character length is fixed, and inter-character space is equal to inter-words space which is smaller than interlines space. As is shown in Figure 2 (a), Euclidean distance based MST is built up on a two-columned Chinese page from the e-book “Taiwan Historical Architecture”. This page has mixed horizontal and vertical text-line orientations. The body text is in horizontal orientation. And the decoration in the left margin is in vertical text line. There exist some scattered illustrative text fragments surrounding the bottom left graphic object. As can be seen, the MST on all the text elements has extracted the right text orientation.

Figure 2 (b) has shown the text block partition results of Figure 2 (a) by breaking MST edges with the proposed cutting method. Furthermore, Figure 2 (b) gives the text line extraction results based on the analysis upon all the text blocks. Taking the non-text layer results, graphic object such as pictorial image and drawing can be integrated into the text line extraction results. Figure 2 (c) presents the final layout analyzing results of this PDF page. Visually, there exist texts are overlapped with bounding boxes of graphic objects, which is of no problem for the reason that the element IDs are recorded for each composite.

Figure 2 (d) to (f) have shown the layout analyzing result of an English PDF page from the e-book “Advances in Selected Plant Physiology Aspects”. The inter-word space of this page violates the assumption that inter-line space is greater than the inter-word space in body text due to the alignment setting and the variety of English word length. The MST edges sometimes don’t pass through the whole text line like its performance on Chinese texts. This causes the whole text line is divided into several fragments within the MST. Figure 2 (d) gives the partitioned MST result after breaking the MST edges. Figure 2 (e) shows the block segmentation based on partitioned MST. And the merged text line extraction and the incorporated graphic segmentation results are illustrated in Figure 2 (f).

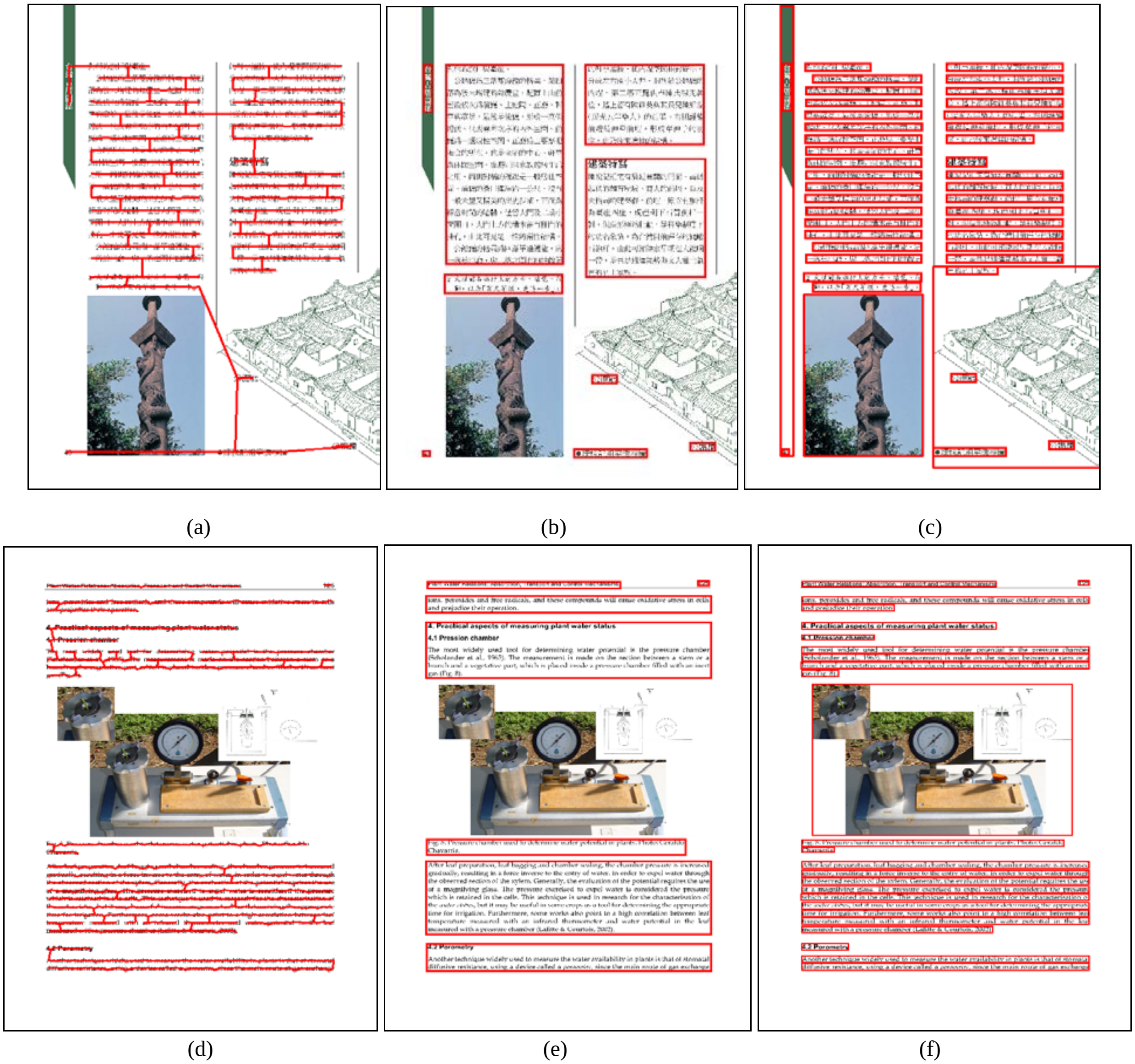
To evaluate our proposed algorithm, a dataset containing variety in language, page layouts has been built. It is consisting of 30 pages, with Chinese and English pages at the proportion of 1:1. The evaluated document pages in Chinese were from 8 e-books provided by Founder Apabi digital library, approximately two selected pages per book. The English pages were crawled from web.

Table 2. Quantitative Evaluation

Column	No. of fragments	Precision	Recall	F1
One column	153	82.6%	75.0%	78.6%
Two columns	370	80.7%	75.4%	77.9%

By extracting the features shown in Table 1, SVM classification is carried out on the dataset. The performance evaluation metrics are from three aspects, including precision, recall and F1. The testing on the 10 pages documents, the labeling performances are given in Table 2. The average processing time for each PDF page ranges from 3 to 5 seconds on a personal computer (3GHz CPU, 3G RAM).

This experiment explored only the dissimilarity of local features of each fragment. The F1 results are similar for fragment labeling of both one column and two columns pages, which gave us a probable potential for performance improvement by taking consideration of the fragment context information and 2D layout features, etc.


(a) (b) (c)

(d) (e) (f)

Figure 2. Two examples of PDF layout analysis: (a) to (c) segmenting results of two columned Chinese page from the book "Taiwan Historical Architecture"; (d) to (f) segmenting results of one-columned English page from the book "Advances in Selected Plant Physiology Aspects".

5. CONCLUSION AND FUTURE WORK

In this work, a graph-based layout analysis method is proposed for PDF document analysis. It involves steps such as preprocessing, text primitives clustering, non-text objects segmentation, feature extraction and labeling. The experimental results on PDF pages have shown satisfactory preliminary results. The contributions of this paper are as follows

- A two layer analysis is introduced for physical structure layout analysis.
- Graph based text elements analysis method is proposed by grouping the page elements according to edge weights like the proximity and orientation statistics for text line identification.
- Features, including geometric layout, character and image features, are extracted for the segmented text and non-text fragments.

The future work will focus on 2D feature engineering of segmented fragments for improving the performance of logical labeling and layout understanding.

Acknowledgements. This work was supported by the National Basic Research Program of China (973 Program) (No.2012CB724108)

REFERENCES

[1] Doucet, A., Kazai, G., and Meunier, J.-L., "ICDAR 2011 book structure extraction competition", in 2011 International Conference on Document Analysis and Recognition, 1501-1505 (2011).

[2] Simone, M. and Emanuele, M., "Table of contents recognition for converting PDF documents in e-book formats", in 10th ACM symposium on Document, 73-76 (2010).

[3] Marinai, S., Marino, E. and Soda, G.,, "Conversion of PDF books in ePub format", in 2011 International Conference on Document Analysis and Recognition, 478-482 (2011).

[4] Tang, Y.Y., Yan, C.D. and Suen, C.Y., "Document processing for automatic knowledge acquisition", IEEE Transactions on Knowledge and Data Engineering, 6(1), 3-21 (1994).

[5] O'Gorman, L., "The document spectrum for page layout analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11), 1162-1173 (1993).

[6] Kise, K., Sato, A. and Iwata, M., "Segmentation of page images using the area voronoi diagram", Computer Vision and Image Understanding, 70, 370-382 (1998).

[7] Simon, A., Pret, J. and A. Peter, J., "A fast algorithm for bottom-up document layout analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(3), 273-277 (1997).

[8] Xiao, Y. and Yan, H., "Text region extraction in a document image based on the Delaunay tessellation", Pattern Recognition, 36(2003), 799-809 (2003).

[9] Ferilli, S., Biba, M. Esposito, F., "A distance-based technique for non-Manhattan layout analysis", in 10th International Conference on Document Analysis and Recognition, 231-235, 2009.

[10] Koo, H. and Cho, N., "State estimation in a document image and its application in text block identification and text line extraction", in ECCV, 421-434 (2010).

[11] Bloechle, J-L., Lalanne, D., Ingold, R., "Ocd: an optimized and canonical document format," 2009 International Conference on Document Analysis and Recognition, 236-240 (2009).

[12] Chao, H., "Graphics extraction in PDF document," in Document Recognition and Retrieval X, Santa Clara, CA, USA, pp, 2003.

[13] Fang, J., Tao, X., Tang, Z., Qiu, R., Liu, Y., "Dataset, ground-truth and performance metrics for table detection evaluation," 2012 10th IAPR International Workshop on Document Analysis Systems, 445-449 (2012).

[14] Fan, J., "Text segmentation of consumer magazines in PDF format," in International Conference on Document Analysis and Recognition, 794-798 (2011).

[15] Lin, X., Gao, L., Tang, Z., Lin, X. and Hu, X., "Performance evaluation of mathematical formula identification.", in 10th International Workshop on Document Analysis Systems, 287-291 (2012).

[16] Breuel, T., M., "Two geometric algorithms for layout analysis," Document Analysis Systems (DAS'02), 188-199 (2002).

[17] Fang, J., Tang, Z. and Gao, L., "Reflowing-driven paragraph recognition for electronic books in PDF", in SPIE-IS&T International Conference of Document Recognition and Retrieval XVIII, 78740U 78741-78749 (2011).

[18] Felzenszwalb, P. F. and Huttenlocher, D. P., "Efficient Graph-Based Image Segmentation", International Journal of Computer Vision, 59(2), 167-181 (2004).

[19] Lin, X., Gao, L., Tang, Z., Lin, X. and Hu, X., "Mathematical formula identification in PDF documents," 2011 International Conference on Document Analysis and Recognition, 1419-1423 (2011).