A Method to Provide High Volume Transaction Outputs Accessibility to Vision Impaired Using Layout Analysis

Azaedeh Nazemi [1], Iain Murray & David A. McMeekin [2]
Departent of Electrical and Computer Engineering ,Curtin University ,Perth ,WA,Australia
[2] Department of Spatial Sciences, ,Curtin University ,Perth ,WA,Australia
Azadeh.nazemi@postgrad.curtin.edu.au ,I.murray@curtin.edu.au , D.McMeekin@curtin.edu.a

ABSTRACT

The Documents in the financial services, insurance, utilities, and government sectors typically require a high volume of PDF documents to be generated which are stored for presentment or archived for legal purposes. As high volume transactional output (HVTO) demands put increasing pressure on online presentment capabilities, accessibility has become a growing concern. In particular, access to these files proposes significant challenges when these documents are presented to visually impaired people using assistive technologies (i.e. screen readers). Since it is rare that all recipients are prepared to accept electronic delivery of their documents, a large portion of the documents is still printed as PDFs. In an online billing system, bills are sent to customers' email accounts as attached PDF files or HTML links. These bills in the most cases are neither accessible through assistive technologies nor useable by vision-impaired customers. This paper provides a method for HVTO documents automatic transformation to an accessible and navigable Mark-up format such as XML or Digital Accessible Information System (DAISY).

Keywords: Vision-Impaired, Layout Analysis, High Volume Transactional Output (HVTO), Accessibility, Optical Character Recognition (OCR)

1. INTRODUCTIN

PDF documents have several features that make it popular for document viewing such as:

- PDF is page independent, which means that there is no need to process pages 1 to 999 in order to process page 1000. Each page stands on its own. This is valuable

when it comes to printing performance. If necessary, multiple processors can be employed to process pages in parallel[1]

- PDF viewers are platform free (Windows, Mac, Linux, even portable devices)
- PDF supports compression of fonts and pages inside it to make the file smaller.

Addressing accessibility in an HVTO (high-volume transaction output) environment such as financial services can be difficult, but is certainly achievable. The industry has already made great strides to address the accessibility of web sites and content portals (such as an online banking interface). To date, financial institutions and other HVTO statement generators deliver alternative format statements to their visually impaired clients using internal consultants or document accessibility services (DAS) [2]. These statements typically come in the form of Braille, large-print documents, or audio CD. But using current outsourcing options to address accessibility issues to vision impaired clients neither cost-effective nor the preferred method because specialized statements generally delay delivery information. This delay influences an organisation's ability to deliver equitable access to all customers and may be seen as discriminatory towards visually impaired customers. Although current options may meet existing standards, their cost and complexity must be considered.

Many PDF creation software vendors allow fonts to be pruned to prevent generating large PDF files when embedding fonts. Author restrictions, pruned fonts, account numbers, overdue notices, charts, multi-columns, graphs, logos and table interfere with a screen reader's ability to properly convey information in an appropriate order.

This research aims to provide cost-effective and efficient HVTO accessible in descriptive alternative audio formats for vision-impaired customers, which considers usability as an important key role in document accessibility. Figure 1 illustrates the image of a sample bill

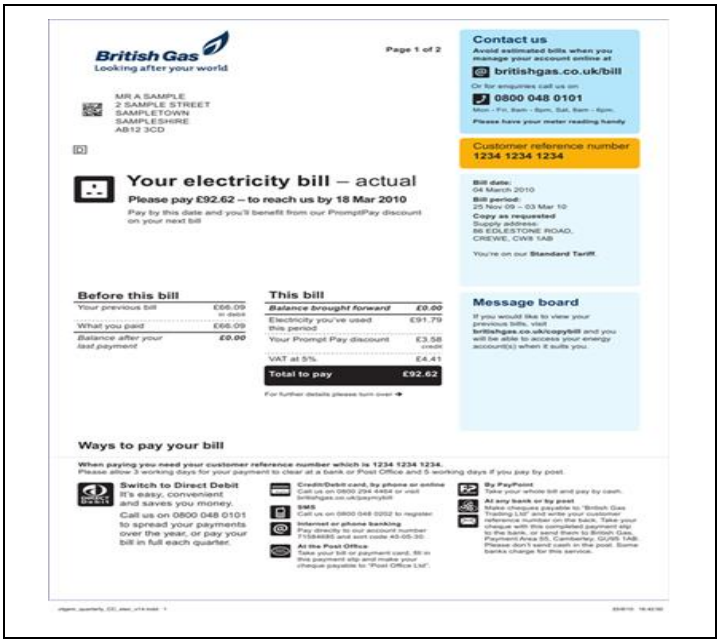


Figure 1The image of a sample bill

2. HVTO CATEGORIES IN TERMS OF ACCESSINBILITY

HVTOs, which are provided to deliver to the customer, have been divided to two categories:

1. Structured but not necessary tagged and consequently they are not navigable. Since a HVTO contains several separated items then the HVTO reading process by a user is totally different from a normal document reading process, which is done sequentially line by line, from top left to bottom right. Thus, navigation ability is a very precious capability during a HVTO reading session. Although this category is not an image only and contains text, in some cases are not accessible through screen readers due to PDF properties such as restrictions adopted during creation. If PDF has structure by converting it to XML, each separable item will be converted to an individual XML element and accessible through screen readers. These elements do not guarantee accurate navigation and usability. **B**y further investigation and modification based on XML parsing these categories will be accessible, usable and navigable.

2. Scanned PDF are inaccessible and definitely needs Optical Character Recognition (OCR) to extract the text from it. However, doing this process before running several pre-processing steps may destroy the reading order and affect the obtained text's usability.

3. HVTO PDF SAMPLES

3.1 Structured PDF

In this section it is supposed which figure 1 is presented as a structured PDF and as a result the following is the output source of conversion it to XML

<text top="429" left="115" width="105" height="20" font="8">on your next bill</text>

<text top="588" left="43" width="146" height="29" font="0">Before this bill</text>

<text top="617" left="43" width="114" height="20" font="8">Your previous bill</text>

<text top="617" left="238" width="50" height="20" font="8">Â£66.09 </text>

<text top="634" left="245" width="39" height="16" font="6">in debit</text>

<text top="657" left="43" width="96" height="20" font="8">what you paid</text>

<text top="657" left="238" width="46" height="20" font="8">Â£66.09</text>

<text top="683" left="43" width="122" height="20" font="17">Balance after your</text>

<text top="701" left="43" width="85" height="20" font="17">last payment</text>

<text top="683" left="246" width="42" height="21" font="13">Â£0.00 </text>

By parsing XML and extract information from it the essential information can be presented as DAISY format.
<tr><th>429 on your next bill 115</th></tr>

<tr><th>588 <Before this bill</ 115</th></tr>

<tr><th>617 Your previous bill 43</th></tr>

<tr><th>617 Â£66.09 43</th></tr>

<tr><th>634 in debit 238</th></tr>

<tr><th>657 what you paid 245</th></tr>

<tr><th>657 Â£66.09 43</th></tr>

<tr><th>683 <Balance after your</ 238</th></tr>

<tr><th>701 <last payment</ 43</th></tr>

<tr><th>683 <<Â£0.00 </</ 43</th></tr>

3.2 Scanned PDF

In this section, it is supposed which figure 1 is presented as a scanned PDF and needs OCR to extract text from it. PDF does not have a spot color space or highlight color space. This

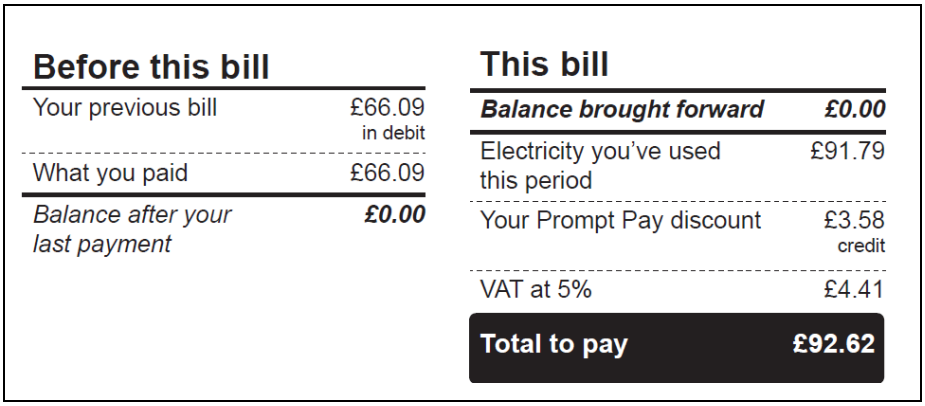
means PDF files need to be either black and white or full color. Accurate pre-processing step includes binarization, image cleaning and skew correction which must be performed before sending scanned PDF to OCR.

Binarization is performed by Implementation of local adaptive thresholding techniques, noise removal performed by using a noise filter reduce impulse or isolated noise in an image.

In addition page frame detection, permitting noise in non-content areas to be cropped away and removed[3]

The result obtained by OCR of binary bill sample image shows it not only needs manual correction for unrecognized non alphanumeric special character in document such as "£" but also requires further investigations in order to reconstruct reading order which destroyed during OCR process.

As it is observed from the text some information are lost in the most important part of this bill due to relocation .This part is shown in figure 2.



Before this bill

Your previous bill	£66.09 in debit
What you paid	£66.09
Balance after your last payment	***£0.00***

This bill

Balance brought forward	***£0.00***
Electricity you've used this period	£91.79
Your Prompt Pay discount	£3.58 credit
VAT at 5%	£4.41
Total to pay	**£92.62**

Figure 1: Lost data segment during OCR bill sample image shown in figure 1

4. PDF LAYOUT ANALYSIS

Performing PDF layout analysis after ordinary pre-processing stage and before main OCR can keep reading order. PDF layout analysis is responsible for identifying text columns, text blocks, text lines, and reading order s. The main target of layout analysis is to take the raw input image and divide it into non-text regions and text lines.

Layout analysis modules must indicate the correct reading order for the collection of text lines. The primary layout analysis is based on whitespace identification and constrained text line finding that both operate on bounding boxes computed for the connected components of the scanned image. The whitespace between the columns is identified as maximum area

whitespace rectangles with a high aspect ratio and large numbers of adjacent, character-sized connected component. The column finder uses a maximal whitespace rectangle algorithm to find vertical whitespace rectangles with a high aspect ratio then selects those rectangles that are adjacent to character-sized components on the left and the right side. These whitespace rectangles represent column boundaries with very high probability. The output of column finding and constrained text line matching is a collection of text line segments.

By checking the bounding boxes of obtained text lines can find relation between them to support appropriate reading order for HVTO. In this research Text-Image Segmentation, Recognition by Adaptive Subdivision of Transformation Space (RAST) -based, Voronoi-based and single column projection for layout analysis are used to classify different regions either text or non-text in the image by block segmentation.

Text-Image Segmentation operates by dividing the input image into candidate regions. Then, features are extracted for each candidate region. Finally, each region is classified using logistic regression into text, grayscale image, line drawing, ruling, and other kinds of regions.

All visual and not textual components must be extracted from binary image. These non-textual components include charts, images of logo, graphs then send for extra processing in chart recognition and chart reader modules optional. Besides tables muse be extracted from PDF to be processed in Table Cell Recognition module for further investigation .This module is responsible to detect and recognize cells. It provides navigation ability through the extracted table information . This navigation can be based on accessing to columns, rows or specific cell information depends on user request.

Extracting these` non-textual components from binary image improves processing speed and OCR accuracy.

RAST is a developed algorithm, consists of three steps: finding the columns, finding the text-lines, then determining the reading order. To find the columns it employs a whitespace rectangle algorithm in that it keeps track of the white spaces rather than the blocks, and combines them as opposed to subdividing the blocks [5]. RAST starts by extracting the connected components then determines the largest possible (maximal) whitespace rectangles (or covers) based on the component bounding boxes. These are then sorted based on how many connected components (e.g., text lines) touch each major side. In this way, column dividers rather than paragraph or section dividers take priority. Once the columns dividers (or gutters) have been found, the connected components are examined and classified as text lines, graphics, and vertical/horizontal rulings based on their shapes and the fact that they do not cross any gutters.

Voronoi algorithm starts by identifying connected components and is able to segment a small collection of complex layouts with the most accuracy the Voronoi algorithm divided the page into regions [6]. As a segmentation algorithm, works fairly well and groups blocks of text

in different colours but did not classify them as text or non-text. Additionally in some cases, it tends to over segment non-text regions. Therefore for HVTO layout analysis both Voronoi and RAST algorithm are used to recognize non-text regions and classify text region[7]

As RAST and Voronoi techniques are not sufficient for accurate PDF layout analysis several techniques are developed in this research to address this issue.

5. DEVELOPED MODULES FOR HVTO LAYOUT ANALYSIS

5.1 Text –Image Segmentation

Text-image segmentation completely separates the image from the text by removing the masked and rectangular regions from an input image. It performs document zone classification using run-lengths and connected components based on features and a logistic regression classifier. Text-Image Segmentation operates by dividing the input image into candidate regions. Then, features are extracted for each candidate region. Finally, each region is classified using logistic regression into text, grayscale image, line drawing, ruling, and other kinds of regions. Since image parts contain fatter lines and larger blobs than the text parts can be extracted by doing :

Dilate the image until all letters are gone, but some parts of the image still remain

convert seg1.png -morphology dilate:3 diamond mpc:-|convert mpc:- txt:-|grep -Ev '#FFFFFF'|sed '1d;s/:.//g;s/,//g'>rgb.txt*

xs=$(cat rgb0.txt|awk '{print $1}'|sort -b -k1n,1|awk 'NR==1')

xe=$(cat rgb0.txt|awk '{print $1}'|sort -b -k1n,1 |awk END'{print}')

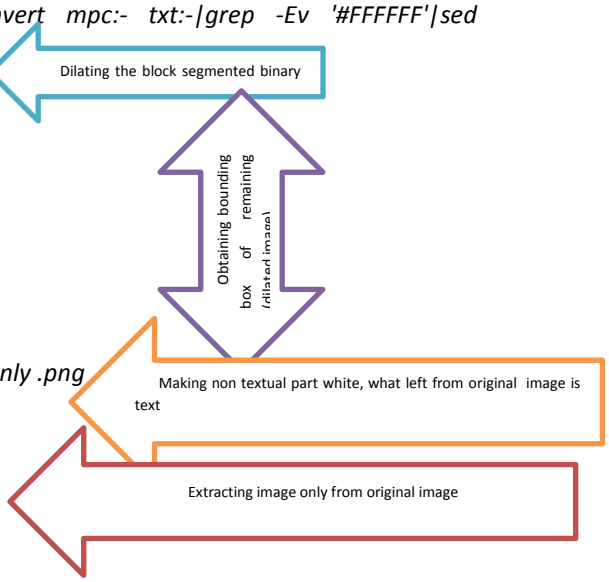
ys=$(cat rgb0.txt| awk '{print $2}'|sort -b -k1n,1|awk 'NR==1')

ye=$(cat rgb0.txt|awk '{print $2}'|sort -b -k1n,1|awk END'{print}')

x=$(($xe-$xs));y=$(($ye-$ys))

convert seg1.png -draw "fill white rectangle $xs,$ys $xe,$ye" textonly .png

convert seg1.png -crop $x"x"$y"+"$xs"+"$ys imageonly .png



Another method to separate text from image is after dilating, perform conditional-erode the dilated image, using the original image as the mask, until the image part is complete again. This means the dilated image has been eroded , but never set a pixel value to below its value in the original source image. The original image is used as a mask to protect parts of the image from changes , this will restore all shapes that still have some seed part left, so only the logo has been left : *convert seg1.png -morphology dilate:3 diamond dilated.png*

convert dilated.png -morphology erode:20 diamond -clip-mask monochrome.png eroded.png

Finally using image contains only image and original image to obtain text only image *: convert eroded.png -negate bin.png -compose plus -composite test.pn*2. Image part including: graph body.[22]

5.2 Chart recognition

Several techniques are used for chart recognition. Chart recognition module is responsible to determine chart type such as pie, bar or line chart. This module is essential to be performed just after text-image se4gmentation and before chart reader Chart recognition module uses image morphology.

By Image Morphology method the structure of shapes within an image could be cleaned up and studied. It works by comparing each pixel in the image against its neighbors in various ways, so as to either add or remove, brighten or darken that pixel. Applied over a whole image, perhaps repetitively, specific shapes can be found and/or removed and modified. If an pixel is white and completely surrounded by other white pixels, then that pixel is obviously not on the edge of the image. The whole process actually depends on the definition of a 'Structuring Element' or 'Kernel', which defines what pixels are to be classed as 'neighbors' for each specific morphological method. The dilate operation returns the maximum value in the neighborhood. The erode operation returns the minimum value in the neighborhood. Use the composite program to overlap two dilate and erode images. Performing binarization, erode and dilate morphology , compositing erode and dilate images, rotation and edge detection produce circle image from pie chart and nothing from line chart and bar chart. Distinguish between bar and line chart is executed by eliminating horizontal lines from image .In this stage several vertical lines remain from bar chart

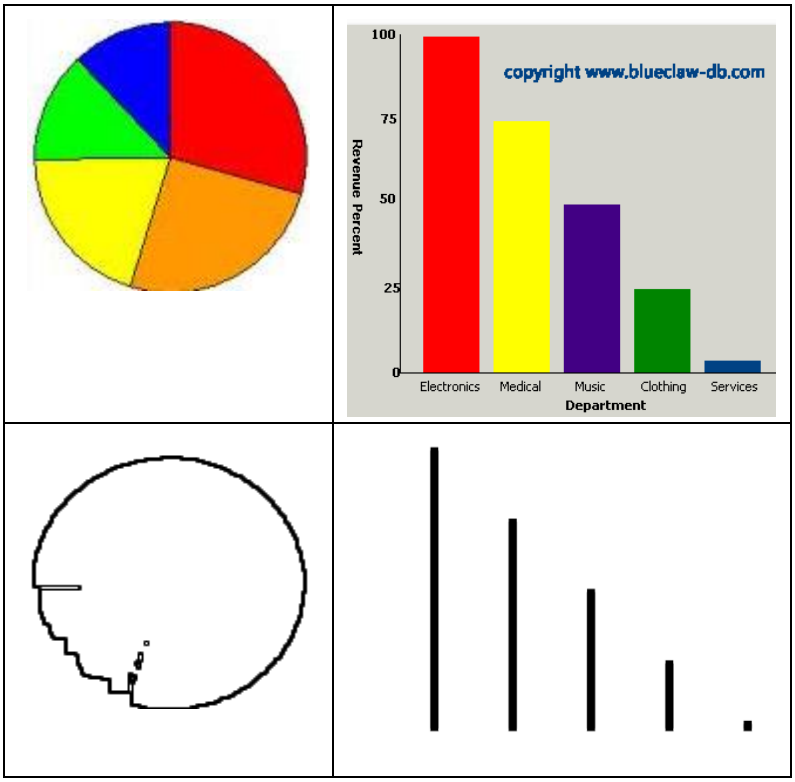


Figure 5-Chart recognition based on morphology

5.3 Table Reader

In creating HVTO, authors use tables to arrange data in rows and columns. The precise conventions and terminology for describing tables varies depending on the context. Further, tables differ significantly in variety, structure, flexibility, notation, representation and use .

In such a case, which HVTO contains table, RAST and Voronoi methods indicate table boundary. Table can be extracted completely as an individual segment. This segment sends to **.** Table Cell Recognition Module that is responsible to specify essential information in table such as, number of column and rows, columns' title, cells' position. This information provide opportunity to generate a descriptive alternative for the table. Alternative description must be detailed and completed that can be replaced with table concepts and explains a table in cooperate with navigation ability. Navigation ability during table reading session helps users to communicate actively with cells value and follow them in column or row order. Table cells recognition module divides table to three categories as following

- Table contains horizontal and vertical identifiers lines, which indicates with yellow colour in RAST result, cells positions can be obtained by finding these lines intersection points

Number of columns=Number of vertical lines -1

Number of rows= Number of horizontal lines-1

*Number of cells= Number of columns * Number of rows*

$$cell_{ij} = intersection\ point\ of\ (horizontal - line)_i and\ (vertiacl - line)_j$$

- Table does not contain horizontal and vertical identifiers lines but white spaces between columns are recognizable by RAST, thus column finder by RAST separates them by yellow vertical line

Number of columns=Number of vertical lines -1

- Table of information neither contains identifier lines nor white spaces between columns are recognizable by layout analysis techniques. In such a case Table Cells Recognition Module include two sub modules:

1. Rows finding: One-Column-Projection is performed over table segment obtained by RAST or Voronoi and divides it to lines. Number of rows in table is equal the number of lines in One-Column-Projection result. Lines bounding boxes include lower-left and upper-right corner coordinates points which indeed specify horizontal identifier lines.

2. Column finding : this sub-module is responsible to:

- Identify connected components(cc) in each line
- Identify white space between two non-connected adjacent components in each line
- Collect white spaces bounding boxes for all lines and sort them based on aspect ratio
- Calculate median value for white space aspect ratio
- Consider median value as a threshold value to specify column separator

Then to access cells position:

h=$(identify -format "%h" hvto.png)

w=$(identify -format "%w" hvto.png)

for i=0 to w

for j=0 to h

if x_{i+1}-x_i=1

$C_{x_i y_j}$, $C_{x_{i+1} y_j}$ *are connected components*

Else $\Delta_i = x_{i+1}$-x_i

median of $\Delta = \frac{\Delta_{\frac{n}{2}} + \Delta_{\frac{n}{2}+1}}{2}$

for i=0 to w

if x_{i+1}-x_i> *median of* Δ

$C_{x_i y_j}$, $C_{x_{i+1} y_j}$ *may be most left points of cells*

$$p_k = x_i$$

else

$C_{x_i y_j}$, $C_{x_{i+1} y_j}$ *are not cell*

for i=0 to k

no[p_i]++

if no[p_i]>1 p_i *is a cell identifier vertical line*

Number of lines=Number of segments in single-column-projection result

Title row of table=first segments of single-column-projection result

Vertical cells identifier=most common wide white space areas

Horizontal cells identifier = lower-left and upper-right corner coordinates points of lines bounding boxes

Cells bounding boxes=intersection points of Vertical cells identifier and Horizontal cells identifier.

As a communication tool, a table allows a form of generalization of information from an unlimited number of different contexts. It provides a familiar way to convey information that might otherwise not be obvious or readily understood. A table consists of an ordered arrangement of rows and columns. The tables are inaccessible as a scanned PDF component such as all other components in scanned PDF .Additionally there is no guarantee for tables to representing correct ordinary structured PDF due to lack of tags.

Table reader module extracts all table cells sort them based on column or row. The method to access each data cell individually is based on finding all columns and rows intersection points To find these points Table Reader uses several image processing techniques. Since table structure often contains vertical lines as column separator and horizontal lines as row separator using morphology erode and dilate technique first removes vertical lines and provides Rows position this processes repeated by removing horizontal lines and obtain columns position

Now by using crop technique and all intersection points table is segmented to cells .All cell are tagged based on positions and sent to OCR .Presenting cell segments to vision impaired users is the main issue regarding table accessibility. Listening to table straight through, without chance to see it visually can be quite confusing. Even by seeing table contents, it can still be confusing if the table is not marked up properly. It means table content linearization is not

enough for tables efficient usability To provide data table accessibility and usability, this access must use both passive and active methods. It means Table Reader must have navigation ability and user can interact with data.

(PK, FK)	Column Name	Description	Data Type	Size	Required	Default value
PK	CustomerID	Unique five-character code based on customer name.	Text	5	Yes	
	CompanyName	Customer Company	Text	40	Yes	
	ContactName	Name of the person who should be contacted	Text	30		
	ContactTitle	Title of the contact person	Text	30		
	Address	Street or post-office box	Text	50	Yes	
	City	City name	Text	20	Yes	
	State	State Code	Text	2	Yes	
	Zip	Five digit zip code	Text	5	Yes	
	Phone	10 digit phone number	Text	15	Yes	
	Fax	Fax number	Text	15		
	Email	Customer email address	Text	20		

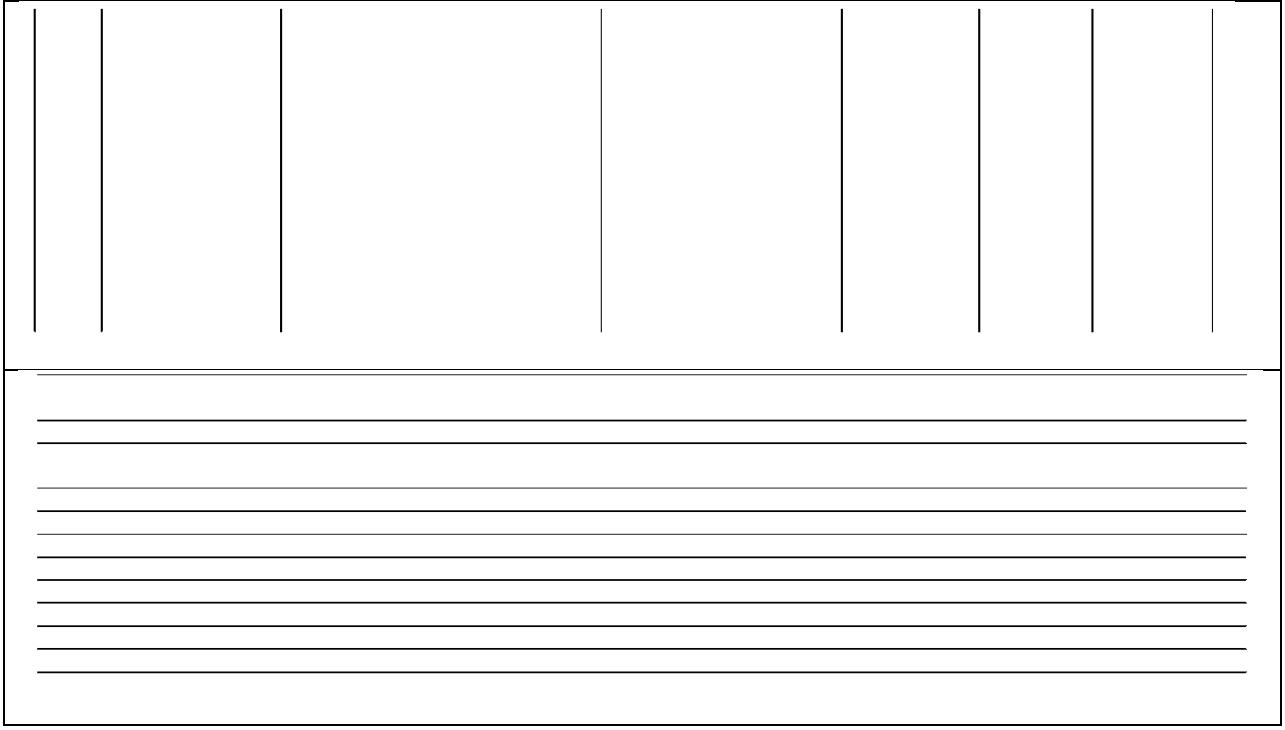


Figure 4. A sample table and its vertical and horizontal separator lines obtained by morphology technique

6. CONCLUSION

.

The research described in this paper has undertaken to develop an application prototype to make HVTO accessible to vision impaired users. The flowchart shown in Figure 5 clearly labels the steps that are included in this process. In developing this prototype, several Open Source software packages have been used as a baseline. These packages include: espeak, ImageMagick, OCRopus. These have then been used with existing algorithms as well as further

developed with the custom algorithms in an attempt to be able to produce accessible HVTO to visually impaired people.

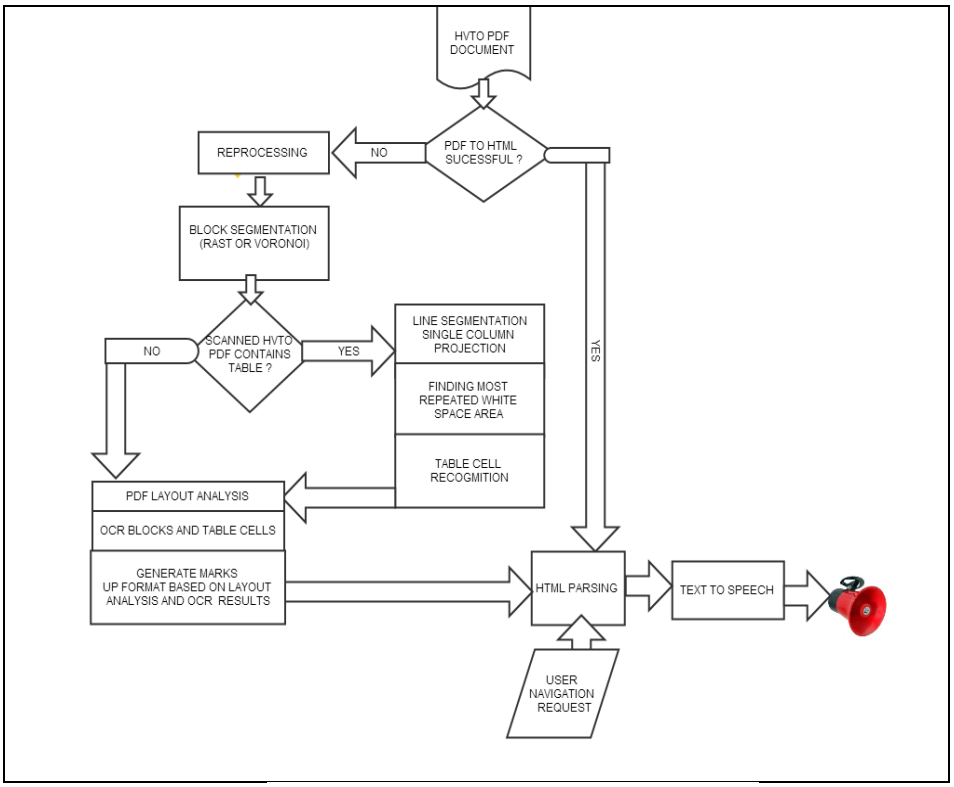


Figure 5. HVTO accessibility application flowchart

.

REFERENCES

[1] Is PDF an appropriate choice for high volume transactional production printing? Crawford Technologies Inc. www.crawfordtech.com

[2]P, Ganza, (2010), Accessibility and the High-volume Transaction Output Enterprisenn

[3] T.M Breuel,"The OCRopus Open Source OCR System" doi=10.1.1.99.8505C

[4]A Nazemi, I Murray, D McMeekin(2014) Layout Analysis for Scanned PDF and Transformation to the Structured PDF Suitable for Vocalization and Navigation Computer and Information Science 7 (1), 162

[5]K, Kise. ,A, Sato,. and M, NB Iwata "Segmentation of Page Images Using the Area

Voronoi Diagram." Computer Vision and Image Understanding, vol. 70(3),

June 1998, pp. 370-382.

[6]A,Winder. (2010),"*Extending the page segmentation algorithms of the OCRopus documentation layout analysis system"*, Boise State University Graduate College, *Theses and Dissertations.* Paper 122.

[7] T.M ,Breuel,. "Two Geometric Algorithms for Layout Analysis." Document

Analysis Systems, August 2002, pp. 188-199.