Internship Report

Submitted to:
Image Understanding and Pattern Recognition
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany

Submitted by:
Trinabh Gupta
2nd Year, B.Tech Computer Science
Indian Institute of Technology , Delhi

Supervisors:
Faisal Shafait
Ilya Mezhirov

Reviewer:
Prof. Dr. Thomas Breuel

Start Date for Internship:-10th May 2007
End Date for Internship:- 26th July 2007

Report Date:-26th July 2007

Abstract

This reports presents the work on:

1. Document Cleanup (Noise Removal) - OCRopus
2. Grouping Text lines into Columns – OCRopus
3. Conversion Programs for converting UW3 and ISRI databases into hOCR format and the evaluation script

1. Document Cleanup – OCRopus

Introduction

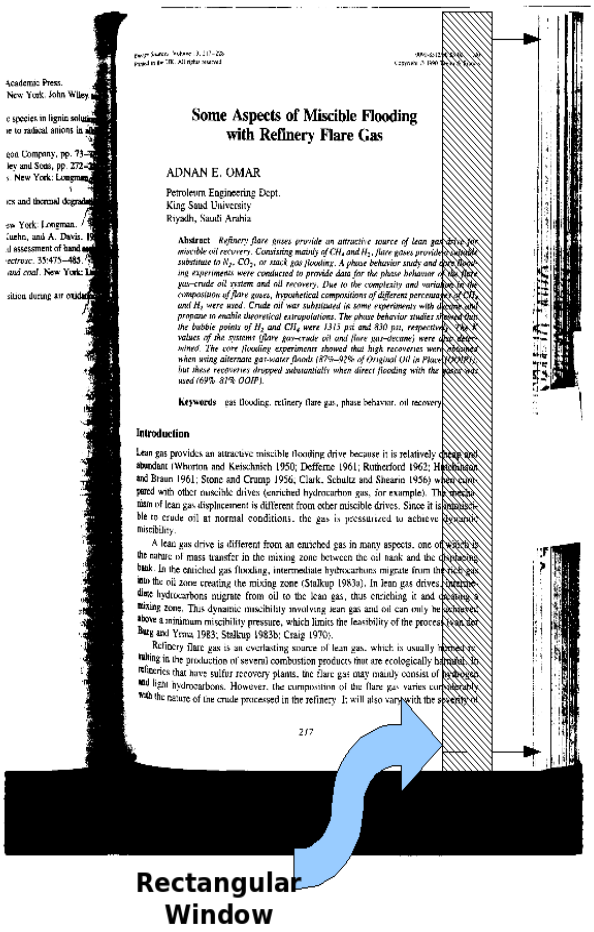
When a page of a book is scanned or photocopied , textual noise (extraneous symbol from the neighboring page) and/or non-textual noise (black borders , speckles etc.) appears along the border of the document. Since the page segmentation algorithms report textual noise regions as text-zones [1] , the OCR accuracy decreases in the presence of textual noise. The most common approach to eliminate marginal noise is to perform document cleanup by filtering out connected components based on their size and aspect ratio [2]. However this method cannot be use to remove textual noise and also this method often removes images from the ground truth zones. Instead of using connected components filtering , researchers have also tried to explicitly detect and remove marginal noise. Unpaper [4] uses a similar approach but it often removes useful data and is also unable to remove textual noise. I have used a combination of all the approaches and made a system that removes the textual noise , margin noise and also does not remove the useful data.

The noise removal is divided into three parts:-

1. Blackfilter
2. Connected Components Filtering
3. Whitefilter

Black Filter

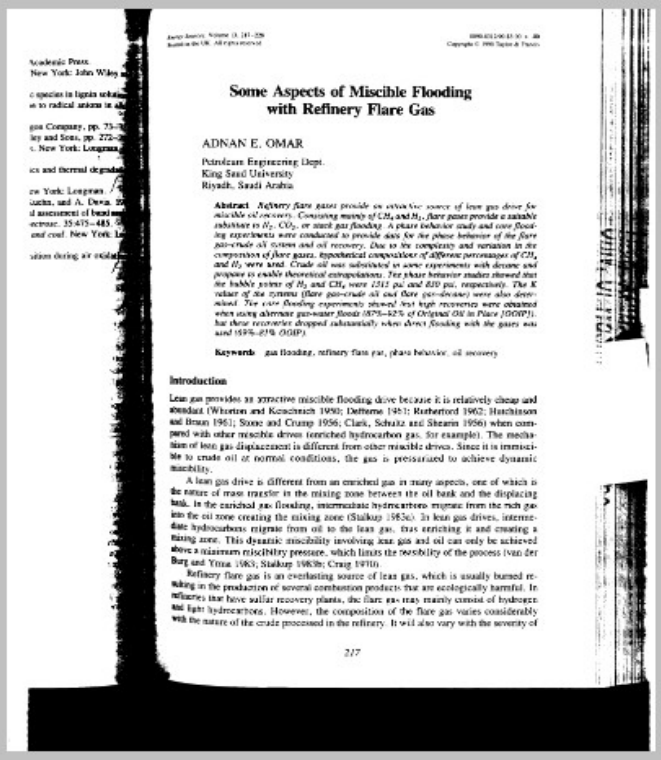
The black filter finds large black areas that come as a result of photocopying or scanning and removes them. It looks for these black areas only at the margins of the image so that it does not affect the text or images in the center of the image. It uses a rectangular window which moves in these parts of the image , calculating the ratio of black pixels under it at any position and comparing it with a threshold .



The rectangular window runs in 1/3rd width or height of the image along the four margins. It starts with the left margin , starting from the *x*-coordinate = 1/3rd of the image width . The width of the rectangular window is specified by a parameter (default set to 5 pixels) . The length or the height of the rectangular window is same as the height of the image. It counts the total number of black pixels under it at any position divided by the total number of pixels under it (equal to width of the rectangular window multiplied by its height) which gives it the ratio of black pixels. If this ratio is greater than the threshold (default set to 0.70) then it removes everything to the left of itself including itself , and also goes directly on to scanning the next margin(right margin in this case). Else , it moves leftward by the parameter called x step (default set to 5 pixels) and continues in the same way until it reaches the left border.

The rectangular window runs similarly on the right edge starting from 2/3rd of the *x*-coordinate and running up to the right border . It then scans the bottom and the top borders , but while scanning the top and the bottom borders the length of the rectangular window is total width of the image minus the points where it met the threshold while scanning the left and right margins . For example : if the width of the image is 3300 pixels and while scanning the left border starting at x=900 it met the threshold at and removed (painted white) the entire left margin from x=0 to x=900 , and while scanning the right border it did not meet the threshold anywhere , so while scanning the top and bottom edges it would scan between x=900 and x=3300 . The length of the rectangular window for scanning top and bottom edges is so chosen as the noise beyond the length of the bar has already been considered . Also if the black pixels ratio goes beyond the threshold while scanning top and bottom edges it removes the entire part up to the top/bottom border , and removes along the entire width of the image .

A visual example of an image after running black filter on it :-



Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

Abstract *Refinery flare gases provide an attractive source of lean gas drive for miscible oil recovery. Consisting mainly of CH_4 and H_2, flare gases provide a suitable substitute to H_2, CO_2, or stack gas flooding. A phase behavior study and core flooding experiments were conducted to provide data for the phase behavior of the flare gas-crude oil system and oil recovery. Due to the complexity and variation in the composition of flare gases, hypothetical compositions of different percentages of CH_4 and H_2 were used. Crude oil was substituted in some experiments with decane and propane to enable theoretical extrapolations. Tee phase behavior studies showed that the bubble points of H_2 and CH_4 were 1315 psi and 810 psi, respectively. The K values of the systems (flare gas-crude oil and flare gas-decane) were also determined. Tee core flooding experiments showed test high recoveries were obtained when using alternate gas-water floods (67%–92% of Original Oil in Place [OOIP]), but these recoveries dropped substantially when direct flooding with the gases was used (59%–81% OOIP).*

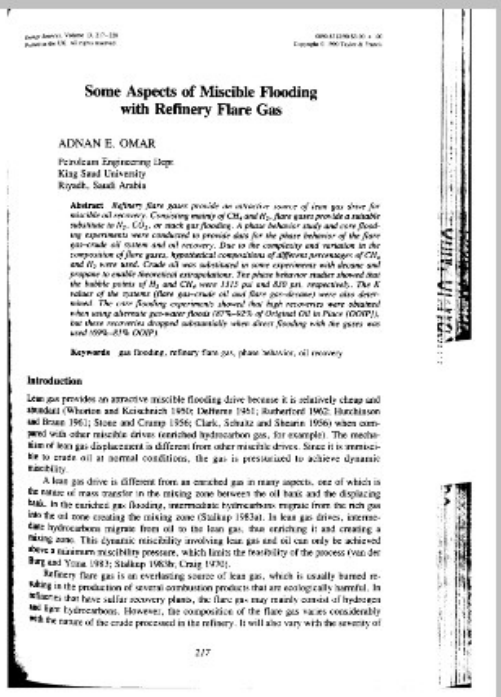
Keywords *gas flooding, refinery flare gas, phase behavior, oil recovery*

Introduction

Lean gas provides an attractive miscible flooding drive because it is relatively cheap and abundant (Wheaton and Kniebusch 1950; Defferre 1951; Rutherford 1962; Hutchinson and Braun 1961; Stone and Crump 1956; Clark, Schulz and Shearin 1956) when compared with other miscible drives (enriched hydrocarbon gas, for example). The mechanism of lean gas displacement is different from other miscible drives. Since it is immiscible to crude oil at normal conditions, the gas is pressurized to achieve dynamic miscibility.

A lean gas drive is different from an enriched gas in many aspects, one of which is the nature of mass transfer in the mixing zone between the oil bank and the displacing bank. In the enriched gas flooding, intermediate hydrocarbons migrate from the rich gas into the oil zone creating the mixing zone (Stalkup 1983a). In lean gas drives, intermediate hydrocarbons migrate from oil to the lean gas, thus enriching it and creating a mixing zone. This dynamic miscibility involving lean gas and oil can only be achieved above a minimum miscibility pressure, which limits the feasibility of the process (van der Burg and Yinza 1983; Stalkup 1983b; Craig 1971).

Refinery flare gas is an everlasting source of lean gas, which is usually burned resulting in the production of several combustion products that are ecologically harmful. In refineries that have sulfur recovery plants, the flare gas may mainly consist of hydrogen and light hydrocarbons. However, the composition of the flare gas varies considerably with the nature of the crude processed in the refinery. It will also vary with the severity of



Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

Abstract *Refinery flare gases provide an attractive source of lean gas drive for miscible oil recovery. Consisting mainly of CH_4 and H_2, flare gases provide a suitable substitute to H_2, CO_2, or stack gas flooding. A phase behavior study and core flooding experiments were conducted to provide data for the phase behavior of the flare gas-crude oil system and oil recovery. Due to the complexity and variation in the composition of flare gases, hypothetical compositions of different percentages of CH_4 and H_2 were used. Crude oil was substituted in some experiments with decane and propane to enable theoretical extrapolations. Tee phase behavior studies showed that the bubble points of H_2 and CH_4 were 1315 psi and 810 psi, respectively. The K values of the systems (flare gas-crude oil and flare gas-decane) were also determined. Tee core flooding experiments showed test high recoveries were obtained when using alternate gas-water floods (67%–92% of Original Oil in Place [OOIP]), but these recoveries dropped substantially when direct flooding with the gases was used (59%–81% OOIP).*

Keywords *gas flooding, refinery flare gas, phase behavior, oil recovery*

Introduction

Lean gas provides an attractive miscible flooding drive because it is relatively cheap and abundant (Wheaton and Kniebusch 1950; Defferre 1951; Rutherford 1962; Hutchinson and Braun 1961; Stone and Crump 1956; Clark, Schulz and Shearin 1956) when compared with other miscible drives (enriched hydrocarbon gas, for example). The mechanism of lean gas displacement is different from other miscible drives. Since it is immiscible to crude oil at normal conditions, the gas is pressurized to achieve dynamic miscibility.

A lean gas drive is different from an enriched gas in many aspects, one of which is the nature of mass transfer in the mixing zone between the oil bank and the displacing bank. In the enriched gas flooding, intermediate hydrocarbons migrate from the rich gas into the oil zone creating the mixing zone (Stalkup 1983a). In lean gas drives, intermediate hydrocarbons migrate from oil to the lean gas, thus enriching it and creating a mixing zone. This dynamic miscibility involving lean gas and oil can only be achieved above a minimum miscibility pressure, which limits the feasibility of the process (van der Burg and Yinza 1983; Stalkup 1983b; Craig 1971).

Refinery flare gas is an everlasting source of lean gas, which is usually burned resulting in the production of several combustion products that are ecologically harmful. In refineries that have sulfur recovery plants, the flare gas may mainly consist of hydrogen and light hydrocarbons. However, the composition of the flare gas varies considerably with the nature of the crude processed in the refinery. It will also vary with the severity of

217

The black filter was evaluated on some of the images of UW3 database compared to the previous connected components filtering . The evaluation was done with respect to the "ideal-images" which were made by removing everything else except what was there in the ground truth zones (the zone boxes) of the images. For evaluation the images starting with "D0" were used and these are exactly hundred in number. Hamming distance was calculated between the ideal image and the image obtained by running black filter on the original image and then similarly the ideal image was compared to the image obtained after doing connected components filtering (as was done in ocr-layout-rast). Two types of hamming distances were calculated – one for the entire image , and the other for the ground truth zones. Ideally what we want is that the ground truth zones do not get affected at all or rather of the total hamming , the ground truth hamming distance is a very small percentage.
For the images starting with "D0" of the UW3 database , evaluation gave the following numbers:-

Average Total Hamming Distance (CC filtering)	2.1117
Average GT Zones Hamming Distance (CC filtering)	1.5921
Average Total Hamming Distance (Black filter)	1.2586
Average GT Zones Hamming Distance (Black filter)	0.2544

First thing to note is that total hamming distance for the image after running black filter on it is significantly less than that obtained after running connected component connected components filtering. Also another significant point is that for CC filtering GT Zones Hamming Distance is 75.2 % of total hamming distance and it is just 20.2 % for the black filter. This is because the black filter runs mostly near the edges and thus affects the ground truth zones much less compared to connected component filtering in which often images are removed from ground truth zones (although we increased the tolerance for images in connected component filtering) .

White Filter

The white filter is very similar to the black filter , the difference being that it removes everything up to the border if it finds a big **white** block. Also it had different threshold and it runs on slightly different areas of the image. White filter similar to the black filter runs on all of the left , right , top and bottom margins , but for the left and right margins it starts from x-coordinate equal to $1/5^{th}$ and $4/5^{th}$ of the image width compared to $1/3^{rd}$ and $2/3^{rd}$ for the black filter . For the top margin it starts from $24/25^{th}$ of the image height and for the bottom margin from $1/50^{th}$ of the image height . This is so in order to prevent the page-footers from being removed as they are very close to the bottom border. The threshold used for the white filter is 0.995 , so that if the number of white pixels are more than 99.5 % of the total pixels under the rectangular window, only then the portion is wiped out. Another significant point to note is that white filter is run on the image returned after running black filter on the original image and doing a connected component filtering.

I then ran white filter on the "D0" images that were previously obtained after running black filter on the original image and calculated hamming distances to the ideal images.

For the "D0" images of the UW3 database after running black and white filters :-

Average Total Hamming distance	0.5965
Average Hamming distance in GT Zones	0.2607

The important thing to note here is that although the total hamming distance is significantly reduced , the average hamming distance for the the GT Zones is nearly the same , which is good as it means that the ground truth zones are not affected and the reduction in total hamming distance is almost entirely due to removal of unwanted noise outside the ground truth zones.

Also I did an evaluation on the "S0" images . I compared the result of running black filter and white filter together with the previous connected components filtering.

Average Total Hamming Distance(CC filtering)	2.1655
Average GT Zone Hamming Distance (CC)	1.9516
Average Total Hamming Distance (BF+WF)	1.1553
Average GT Zone Hamming Distance (BF + WF)	0.9317

Again there is significant improvement in the hamming distances for the S0 images .

Connected Components Filtering

Also later on , my supervisor suggested that we should do a connected component filtering between black filter and white filter as the white filter worked better if we gave it a more cleaner image. The connected components filtering removed only very big components or very small components –

1. Components with area less than *9* pixels or
2. Height less than *3* pixels or
3. Width less than *3* pixels or
4. Components with height greater than *2/3*rd of image height or
5. Width greater than *2/3*rd of image width or the
6. Components which were within *50* pixels from the borders.

After running Connected components filtering on the image obtained after running black filter on it:-

Energy Sources, Volume 18, pp. 217–231
Printed in the USA.

0090-8312/96/$12.00 + .00

Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

Abstract *Refinery flare gases provide an attractive source of lean gas drive for miscible oil recovery. Containing mainly of CH_4 and H_2, flare gases provide a suitable substitute to N_2, CO_2, in rich gas flooding. A phase behavior study and core flooding experiments were conducted to provide data for the phase behavior of the flare gas-crude oil system and oil recovery. Due to the complexity and variation in the composition of flare gases, hypothetical compositions of different percentages of CH_4 and H_2 were used. Crude oil was substituted in some experiments with decane and propane to enable theoretical extrapolations. The phase behavior studies showed that the bubble points of H_2 and CH_4 were 1315 psi and 850 psi, respectively. The K values of the systems (flare gas-crude oil and flare gas-brine) were also determined. The core flooding experiments showed that high recoveries were obtained when using alternate gas-water floods (87%–92% of Original Oil in Place (OOIP)), but these recoveries dropped substantially when dryn flooding with the gases was used (69%–81% OOIP).*

Keywords gas flooding, refinery flare gas, phase behavior, oil recovery

Introduction

Lean gas provides an attractive miscible flooding drive because it is relatively cheap and abundant (Wheeler and Kolschaich 1950; Deffenbaugh 1961; Rutherford 1962; Hutchinson and Braun 1961; Stone and Crump 1956; Clark, Schultz and Shearin 1956) when compared with other miscible drives (enriched hydrocarbon gas, for example). The mechanism of lean gas displacement is different from other miscible drives. Since it is immiscible to crude oil at normal conditions, the gas is pressurized to achieve dynamic miscibility.

A lean gas drive is different from an enriched gas in many aspects, one of which is the nature of mass transfer in the mixing zone between the oil bank and the displacing bank. In the enriched gas flooding, intermediate hydrocarbons migrate from the rich gas into the oil zone creating the mixing zone (Stalkup 1983a). In lean gas drives, intermediate hydrocarbons migrate from oil to the lean gas, thus enriching it and creating a mixing zone. This dynamic miscibility involving lean gas and oil can only be achieved above a minimum miscibility pressure, which limits the feasibility of the process (van der Burg and Yrma 1983; Stalkup 1983b; Craig 1970).

Refinery flare gas is an overlooking source of lean gas, which is usually burned resulting in the production of several combustion products that are ecologically harmful. In refineries that have sulfur recovery plants, the flare gas may mainly consist of hydrogen and light hydrocarbons. However, the composition of the flare gas varies considerably with the nature of the crude processed in the refinery. It will also vary with the severity of

Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

Abstract Refinery flare gases provide an attractive source of lean gas drive for miscible oil recovery. Containing mainly of CH_4 and H_2, flare gases provide a suitable substitute to N_2, CO_2, in rich gas flooding. A phase behavior study and core flooding experiments were conducted to provide data for the phase behavior of the flare gas-crude oil system and oil recovery. Due to the complexity and variation in the composition of flare gases, hypothetical compositions of different percentages of CH_4 and H_2 were used. Crude oil was substituted in some experiments with decane and propane to enable theoretical extrapolations. The phase behavior studies showed that the bubble points of H_2 and CH_4 were 1315 psi and 850 psi, respectively. The K values of the systems (flare gas-crude oil and flare gas-brine) were also determined. The core flooding experiments showed that high recoveries were obtained when using alternate gas-water floods (87%–92% of Original Oil in Place (OOIP)), but these recoveries dropped substantially when dryn flooding with the gases was used (69%–81% OOIP).

Keywords gas flooding, refinery flare gas, phase behavior, oil recovery

Introduction

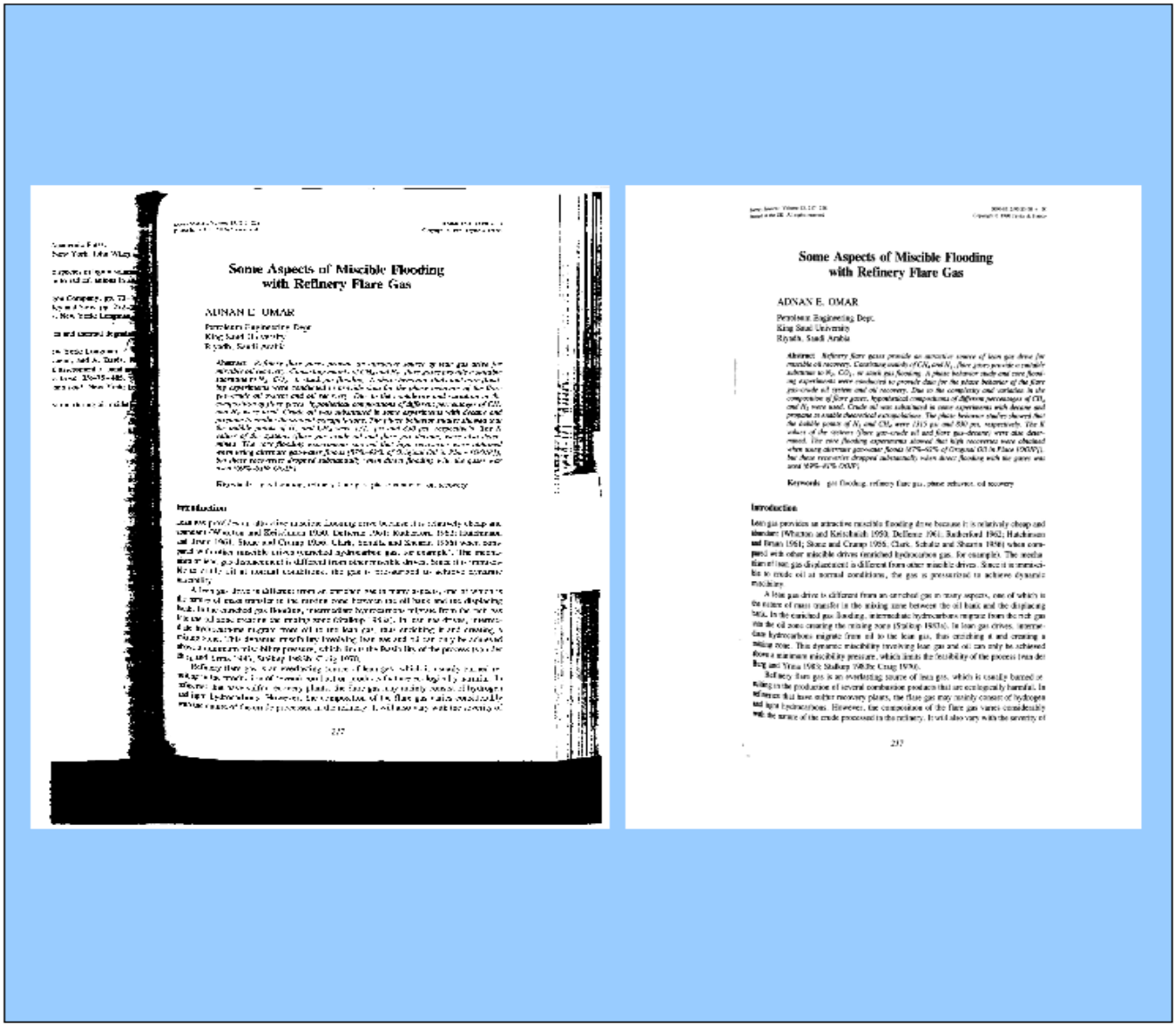
Lean gas provides an attractive miscible flooding drive because it is relatively cheap and abundant (Wheeler and Kolschaich 1950; Deffenbaugh 1961; Rutherford 1962; Hutchinson and Braun 1961; Stone and Crump 1956; Clark, Schultz and Shearin 1956) when compared with other miscible drives (enriched hydrocarbon gas, for example). The mechanism of lean gas displacement is different from other miscible drives. Since it is immiscible to crude oil at normal conditions, the gas is pressurized to achieve dynamic miscibility.

A lean gas drive is different from an enriched gas in many aspects, one of which is the nature of mass transfer in the mixing zone between the oil bank and the displacing bank. In the enriched gas flooding, intermediate hydrocarbons migrate from the rich gas into the oil zone creating the mixing zone (Stalkup 1983a). In lean gas drives, intermediate hydrocarbons migrate from oil to the lean gas, thus enriching it and creating a mixing zone. This dynamic miscibility involving lean gas and oil can only be achieved above a minimum miscibility pressure, which limits the feasibility of the process (van der Burg and Yrma 1983; Stalkup 1983b; Craig 1970).

Refinery flare gas is an overlooking source of lean gas, which is usually burned resulting in the production of several combustion products that are ecologically harmful. In refineries that have sulfur recovery plants, the flare gas may mainly consist of hydrogen and light hydrocarbons. However, the composition of the flare gas varies considerably with the nature of the crude processed in the refinery. It will also vary with the severity of

217

The final result after running black filter , connected components filtering and white filter :-



Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

Some Aspects of Miscible Flooding with Refinery Flare Gas

ADNAN E. OMAR

Petroleum Engineering Dept.
King Saud University
Riyadh, Saudi Arabia

The entire code for document cleanup (noise removal) , implementing the interface "ICleanupBinary" was put into a separate directory "ocr-doc-clean" and integrated with the development version of ocropus [5] .

As a final evaluation , OCRopus[5] was run on the entire UW3 database with and without the document cleanup and OCRopus[5] output was compared with the ground truth using hOCR-tools.

Total Characters in UW3 ground truth are :- 5117393

	Without Document Cleanup	**With Document Cleanup**
Total Segmentation errors	628952 (12.29%)	457936 (8.948 %)
Total Ocr errors	279150 (5.45%)	278908 (5.45 %)

So the document cleanup helps significantly in reducing the segmentation errors (around 3.3 %) which arise mostly due to unwanted text.
Note: The conversion program used (for converting UW3 ground truth to hOCR [6] format) for this evaluation did not still consider the latex commands in the ground truth , so the errors may be approximate but relatively it does not make any difference (as the program used was same for both evaluations).

2. Grouping text lines into paragraphs and paragraphs into columns- OCRopus

A basic system was set up that grouped the text lines detected into columns .The text lines are found using RAST [2] and sorted in reading order (part of ocr-layout-rast) [3] .

The grouping is done in three parts . Firstly the lines are separated on the basis of *y*-coordinate. Since the text lines are sorted in reading order , if the *y*-coordinate of next line is greater than the *y* coordinate of the current line , then definitely we have reached the end of a column .

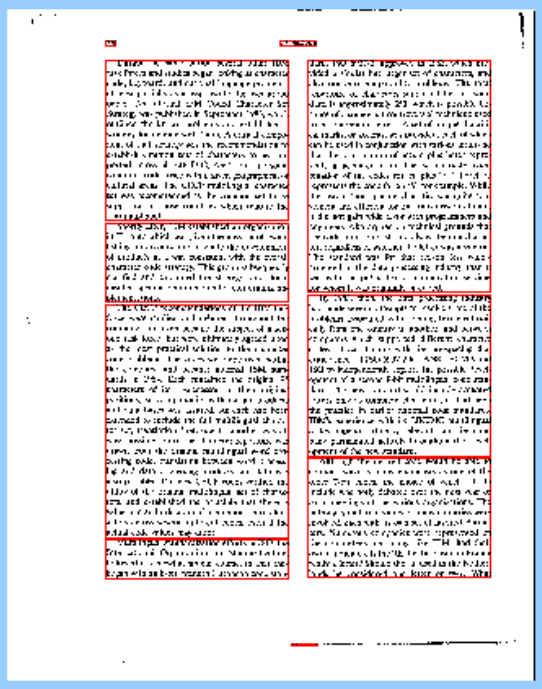
In the second part of grouping for the partially grouped text lines relative alignment is found . Considering each probable column that was made in the first part each line's relative alignment is found with respect to its previous line. The line maybe left aligned , right aligned , center aligned or justified. Alignment is done by considering the *x*-coordinates. A line is left aligned compared to its previous if *X0* of current line lies within a certain range of pixels of *X0* of previous line. Also for a line to be left aligned it must not be right aligned and center aligned. Similarly for a line to be right aligned compared to its previous its *X1* must lie within a certain range of *X1* of previous line . It must not be left aligned or center aligned for it to be right aligned. For the line to be center aligned its *(X0+X1)/2* must lie within a certain range of previous line's center . Also it must not be left aligned or right aligned. If the line is left , right as well as center aligned then it is called justified. Else the line is considered to be not aligned to the previous line.

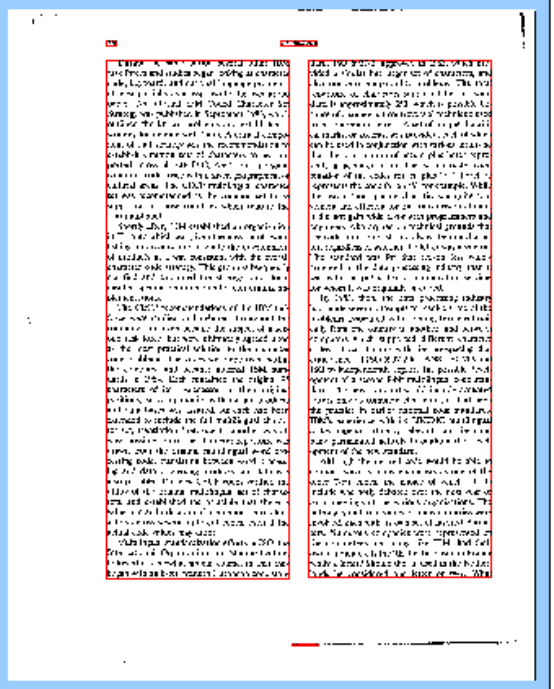
	X0	**X1**	**Xc = X0+X1 / 2**
Left	Yes	No	No
Right	No	Yes	No
Center	No	No	Yes
Justified	Yes	Yes	Yes

Now after relative alignment is done , the lines are further divided into groups at the points where alignment changes. So now we have groups of lines all of which have same relative alignment , essentially each group is very closely ordered . A point to be noted is that the groups are still arranged in reading order . Now if a group contains exactly one line and it is left aligned to the last line of the previous group , then it is merged into the previous group because it is essentially the last line of the paragraph . Also if a group contains exactly two lines , the second one being right aligned to the first one , then these two lines are merged into the next group , because essentially these two lines are start of the paragraph. The groups of lines we have now are paragraphs.

In the third part of grouping the bounding boxes of the paragraphs are considered . In this part we group the paragraphs into columns. Firstly the boxes are separated on the basis of *y*-coordinate , if the *y* coordinate (Y1) of the paragraph is greater than *Y1* of previous paragraph then we essentially have a column break. Then for each group we calculate the overlap between every two consecutive paragraphs. Overlap is calculated as twice of intersection divided by sum of widths of both paragraphs, where intersection is min(first.X1 ,second.X1)-max(first.X0,second.X0) . If the overlap is greater than a threshold(default set to 0.6) we group these paragraphs together and consider them a column.

Here is how the program worked on one of the images in UW3 database. The first one shows how it detected the paragraphs and the second one shows the result of grouping these paragraphs into columns.





3. Conversion Programs

Conversion programs for UW3 database

For evaluating OCRopus [5] on UW3 database, UW3 ground truth had to be converted into hOCR [6] format – the format in which OCRopus [5] gives it output, so that we can use the hOCR-tools for evaluation . The UW3 database is given in DAFS-B format . The conversion program written in **python** converts the DAFS-B format into hOCR [6] format .

Usage: dafs-to-hocr arg1 arg2

arg1: ground truth text file
arg2: output html file

In the DAFS format the page is the biggest entity, which is divided into entities like zones which are further divided into lines and lines into words , giving it a tree like structure. The page has its properties (attributes like Dominant Font Size , Language etc) which is all incorporated in the meta data in the output file. Each zone also has its similar properties which are also incorporated into the output file in the form of html tags .The latex commands given in ground truth are also converted into html entities . All the information about the ground truth is incorporated into the output hOCR [6] file except the word boxes as these are not given by OCRopus [5] and are not required for evaluation. Also the evaluation program counts the total number of characters in the ground truth .

Conversion program for ISRI Database

The basic difference between ISRI and UW3 databases is that the ISRI database does not contain the line boxes or the word boxes. It just has the zone boxes (like text zone , header , footer) and the data present in these zones. Also the boxes are given in one file and the data in each box is given in a separate file for each zone.

The conversion program is coded in **python** .

Usage: isri-to-hocr arg1 arg2 arg3

arg1 : text file containing zone boxes
arg2 : directory containing data files for the zones
arg3 : output html file

Evaluation Script for Ocropus (on UW3 database)

I also wrote a bash script for evaluating OCRopus[5] on UW3 database, using the DAFS to hOCR [6] conversion program and hOCR-tools. It gives the Segmentation errors and Ocr errors for each image in UW3 database and also calculates the errors in percentage using the total number of characters.

Usage: evaluate arg1 arg2 arg3

arg1: UW3 Image Directory
arg2: Directory in which to put the ocropus results
arg3: Ocropus Directory

This script can be used to evaluate changes made into OCRopus [5] in terms of the segmentation errors and ocr errors on UW3 database.

References:

1. Shafait, F., Keysers, D., Breuel, T.M.: Pixel-accurate representation and evaluation of page segmentation in document images. In: 18th Int. Conf. on Pattern Recognition, Hong Kong, China (Aug. 2006) 872–875
2. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Document Analysis Systems, Princeton, NY (Aug. 2002) 188–199
3. Breuel, T.M.: A practical, globally optimal algorithm for geometric matching under uncertainty. Electr. Notes Theor. Comput. Sci. 46 (2001) 1–15
4. URL- http://unpaper.berlios.de/#overview
5. URL- http://www.ocropus.org/
6. Breuel, T.M. : The hOCR Microformat for OCR Workflow and Results , ICDAR