Auto-Zoning Newspaper Articles for the Purpose of Corpus Development for Training OCR

By

Alan B. Cannaday II, FamilySearch, Research Team;

Background

Obituaries can provide a lot of information regarding the deceased and their relatives and acquaintances. FamilySearch has recently received an influx of obituaries through donation or contracts for ingestion into the digital storage and/or to make the information searchable through the FamilySearch website. Currently information extraction is being completed using volunteer help by missionaries and through FamilySearch Indexing. This is a time consuming process and thus, only limited information is gleaned from these images. By automating the information extraction process using computers, obituaries have the potential of being processed faster and extracting a larger amount of the total information available. Automatic information extraction, or "robo-keying", is a project currently being researched by the FamilySearch, Research Team.

One of the key steps to robo-keying, is the use of Optical Character Recognition (OCR) or the automatic transcription of documents. Most OCR systems focus less on the overall flow of the text in the document and focus more on the transcription of individual words and letters. This creates a problems in transcription continuity when transcribing newspapers as they can span over multiple columns. A preprocessing step called zoning can be done to create column and article separation and reduce unwanted clutter such as borders, commercial ads, and images in with digital copies of documents.

The zoning process also is currently being done by volunteers in order to present single obituaries to patrons for indexing. Zoning by hand is tedious and time consuming. The effectiveness and speed of zoning could be majorly augmented using automation. The ultimate goal of auto-zoning would be for the results to be used as completely as those being hand zoned and require the utilizing of more advanced techniques after OCR, such as natural language processing. The level of accuracy for this is not necessary to create a corpus for OCR training and would require more research. This paper describes the efforts, by the research team so far, to develop an auto-zoning technique to create zones of accuracy fair enough to help train OCR systems such as Tesseract and Ocular. Though, we recognize this may not be a definitive corpus, for as research progresses, additional needs may develop.

Binarization

Binary Image Input

Images come in the formats of binary, grayscale, and color images. At the current point in this project, we have only received binary and grayscale images. There are numerous binarization processes that have developed in the last decade, especially with the introduction of the Document Image Binarization Contest (DIBCO) [1] introduced as part of ICDAR in 2009. Some of the more successful historical document binarization approaches involves Adaptive NiBlack Thresholding. An in house variation to Adaptive NiBlack Thresholding (described in forthcoming paper) was used in the thresholding of documented that have been used in this process thus far. It needs to be mentioned that the zoning process described herewith processes a binary images and all images not binary would need binarization as a preprocessing step in order for this auto-zoning process to be utilized.

Zoning Criteria

The ideal result for auto-zoning from a newspaper page, beyond just OCR training, would be a set of zones containing a set of characters that represent the whole or part of an article. As this would be the ideal results, a more reasonable completion criteria for auto-zoning was used for the purpose of OCR training as an acceptable zone. This criteria not only met the needs for training an OCR system, but also to reduced confusion for contractor of whom the zones where primarily transcribed. The following where the criteria:

1. Zones may contain multiple articles or parts of articles, but may not represent text from multiple columns.
2. Zones must contain no partial characters. We decides that, for the purpose of training, it is okay for words or sentences to be incomplete, but the characters needed completeness. When the final images where produced few pixels a buffer edge was used beyond the final determined edge. A buffer can extend a zone to collect a few pixel of neighboring characters, however anything more than a few pixels in not acceptable. The inclusion of lines and local boarders is acceptable.
3. Zones must contain text with continuous flow. Continuous flow problems are those that, as of now, cannot be checked in the automation process or are caused by error in the process. Continuous flow means that there is an obvious flow to the way the article is transcribed. This eliminates zones with embedded images that include captions, lettering facing multiple directions, and any instance of embedded multiple columns or font size which flow could be interpreted multiple ways.
4. Zones are preferred to contain multiple lines of text. This means that single character, word, or line zones where less preferred. Though this may not effect OCR training in the long run and we did send some of the single character, word, or line zones in, it saved time in hand vetting the images for transcription and most articles are not single word or single text line. The final images we sent for transcription had a roughly 3:1 ratio of multiline image to images containing single character, word, or line zones.

The current trend for document segmentation involves the use of some form of texture analysis. Texture analysis seems to be affective, but expensive. Taking into consideration that FamilySearch deals with a large volume of documents, we decided to go with a geometric approach at solving the zoning problem. Though proven, so far, effective to create zones for annotation and OCR training, there is potential for improvements and error reduction. For this reason, all images were vetted prior to being sent out to be transcribed. Also, we have yet to create an appropriate means of scoring automated zoning though it may involve a similar structure to the HNLA2013 [2].

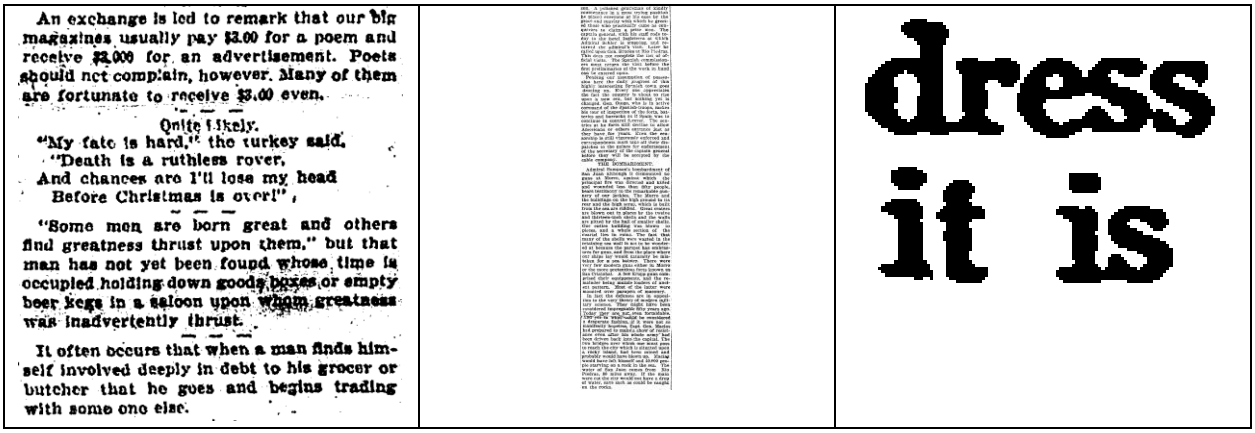
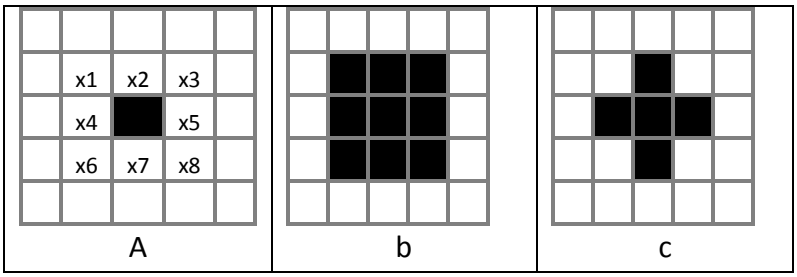


Figure 1: Shows examples of acceptable zoning to be used in OCR training with the conditions explained in the paper. Note the images in the left and center panels contain many line of text with complete lines and the image in the right panel nine letters that are not necessarily part of the same sentence, but all character features are complete and the transcription flow is obvious.

Erosion/Dilation

There are many mentions of erosion and dilation in this paper. The grids in panels a-d of Figure 2 represent a set of pixels size 5x5. The center pixel in panel a is black, while the remaining pixels are white. Each of the 8 pixels surrounding the center pixel could labels as x1-x8 respectively, as shown in panel a. The term erosion come from the appearance of the white pixels being changed or eroded by neighboring black pixels. An 8-point erosion would erode all eight surrounding pixels, x1-x8, of the center black pixel to black as creating panel b. Erosion can also be used in different arrangements: A 4-point erosion would change x2, x4, x5, and x7 surrounding the center black pixel to black as in panel c or other variation as in panel d which method erodes using x2 and x7. Panel f of Figure 2 shows the 8-point erosion or panel e. Dilation can be described as the inverse of erosion where the white dilates into the black.



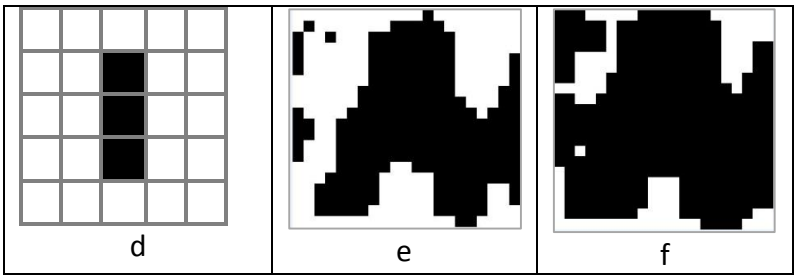


Figure 2: These panels are used to help explain erosion. The grids in panels a through d represent a set of pixels size 5x5. The center pixel in panel a is black or value 0, while the remaining pixels are white or value 1. Panel b shows 8-point erosion, panel c shows 4-point erosion, and panel b through d represent different possibilities of erosion.

Clutter and Noise Removal

From this point forward, let's say a feature is any set of black pixels in an image that neighbor or are neighbors by extension of neighboring black pixels. Features includes characters, line, boarders, clutter image, noise, and others. In this process and explained later, blob detection is a very important part of zone creation. Blob detection is the process of detection segmenting regions within an image were all the features within the blob have at least one similar attribute. Human minds exploit attributes such as relative closeness of characters and words, line spacing, and line delimiters to determine which characters and words belong to which articles on the page. By analyzing these same attributes could also be exploited through image processing to create word and article blobs.

Features can be falsely identified to be in the same blob due to unwanted features. These features can cause false connecting or bridging between blobs and the over-extension of blob boundaries. This can lead to blobs with multiple columns and the inclusion of characters from neighboring columns respectfully. By eliminating unwanted features, blob bridging and boundaries edges are more controlled, thus, producing more desirable zones. These unwanted features include:

1. Clutter; any unwanted features larger than a character.
2. Lines; technically lines, or more specifically delimiter lines, could be considered clutter. However, because of the thinness of most delimiter lines are similar to the thickness of character line width, lines receive a separate treatment than clutter.
3. Noise; any unwanted features smaller than a character.

Clutter Removal

The removal of large black features is valuable to eliminate unwanted borders and large, black sections of images that tend to cause false bridging in later steps. The approach is as follows:

1. Use 8-point dilation a certain number of times (we used 7). This will eliminate a large percentage of the unwanted character features from being indicated as clutter, though some of the large, bold titles and headers will still be indicated as clutter and eliminated. The remaining black pixels indicate locations of remaining clutter.

2. The locations of the remaining black pixels can then be used as seeds, or starting points, for flood filling original binary image. Flood filling by seed means that each feature at each seed location is turned to white. Thus, eliminating clutter features (Figure 3).

Possible Improvements: Using word zone highs to determine a second clutter removal using the mean zone high to gage the number of dilations. Also, bold lettered headers also can get categorized as clutter.

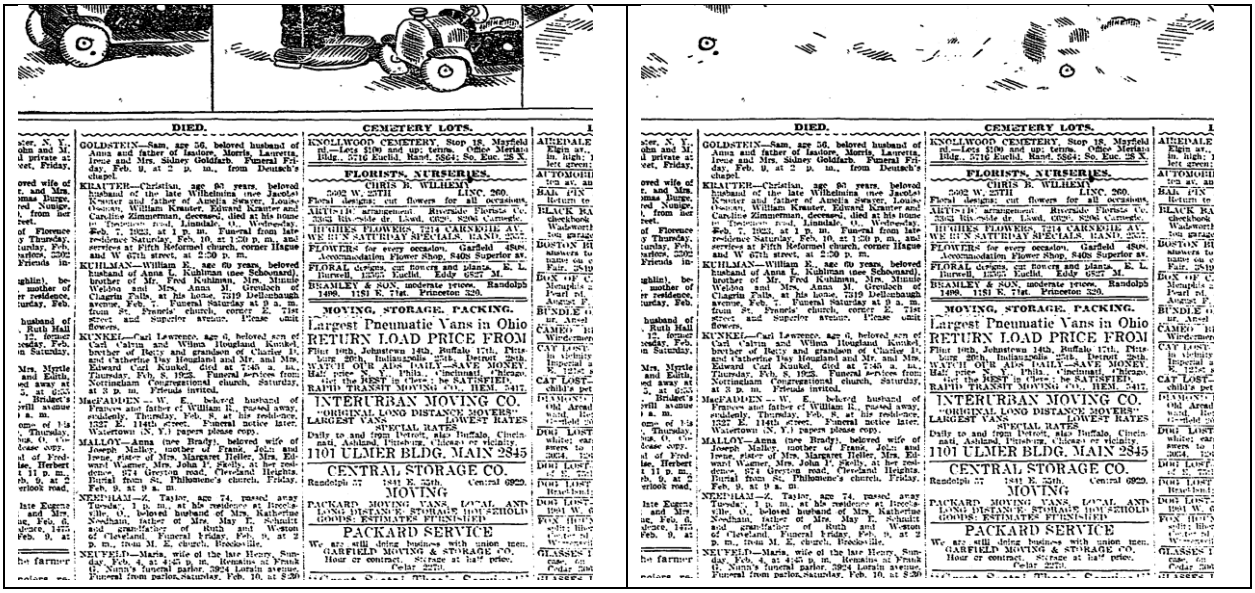


Figure 3: Shows original image in the left and image after clutter removal on the right.

Line Removal

In newspapers, lines are used as delimiters and, just as other clutter, can cause bridging and boundary overextension of zone. By eliminating lines, white space is also created as a delimiter, thus making the image more ideal for automated zoning. Line detection is a largely studied field of image processing and is usually done on non-text based images such as documents. The process generally also requires uniform, grid based, or intersecting lines. Our approach on line detention and elimination relies on the behaviors and characteristics of a line when blurred with neighboring, possibly other pixels that belong to the same line.

The theory is that if there was a line of black ink one a white background and someone was to smear the ink in the same direction as the line, then the line would remain black. However, if someone was to smear the ink in a direction perpendicular to the line then the ink would smear and the shade would lighten. Our approach is as follows with visual representation of the steps in Figure 4 :

1. Let X_1 represent a binary image where white represents a high value of 1, and black represents the low value of 0. Lines generally are black in images.

2. By uniformly smoothing X_1 in a vertical direction using a filter of high L and width 1. This will produce an image where all values are between 0 and 1. Vertical lines tend to maintain a low value where are texts tends to blur into mid-range values and background maintains a higher value (top-left panel). We used $L = 21$.

   Let filter F() be the uniform filter of size Lx1,

   Then let $F(X_1) = X_2$.

3. The image X_2 is then smoothed in the horizontal direction to produce X_3. Because of the thinness of most delimiter lines, lines form X_2 tend to become gain high value in X_3. However, text features and background, having previously been blurred in X_2, maintain similar values this new image (top-center panel).

   $$F(X_2) = X_3.$$

4. By taking the different $X_2 - X_3$ (top-right panel), the text and background are minimized leaving higher values in the locations of possible lines. Possible lines can then be found using image thresholding value, t. We used $t = 0.75$.

   $$(X_2 - X_3) < t = X_4$$

   Note: An extra dilation can be performed after the thresholding to reduce possible false line. In this case

   $$dilate((X_2 - X_3) < t) = X_4$$

5. Using the remaining black pixels in X_4 as seeds, flood fills are executed on a threshold version of X_3 (bottom-left panel). We used $t = 0.025$.

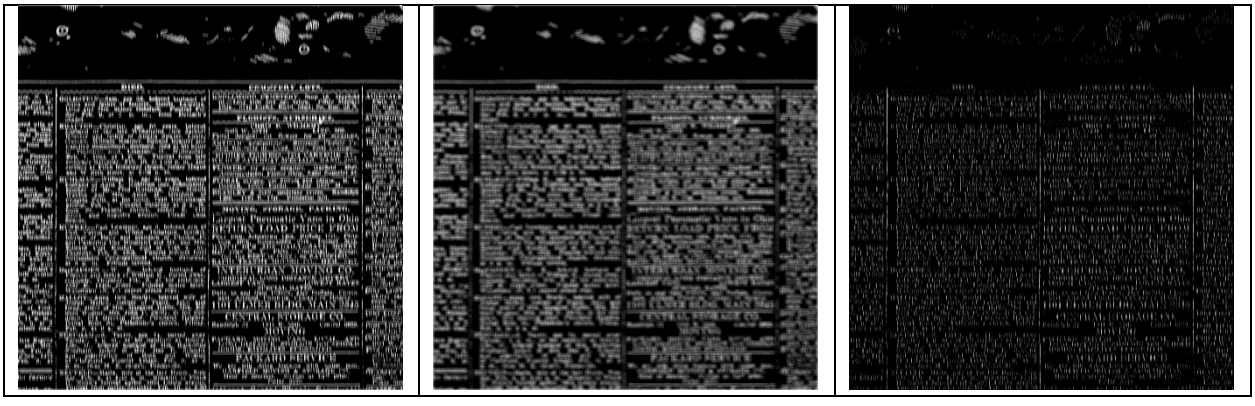
   $$floodfill_{4point(X_3<t)} \forall (X_4 == 0) = X_5$$

6. Steps 1-5 is then repeated for the horizontal direction to produce Y_5
7. Thus, by combining the inverse of X_5 and Y_5 (bottom-center panel) we get a full representation of the lines found in the image.

   $$X_5 + Y_5 = XY_5$$

8. The final step is to combine X_1 and XY_5 producing the results from clutter removal and line removal (bottom-right panel).

Possible Improvements: Line width could somehow be analyzed through-out the image to determine appropriate lengths for the blur filter.



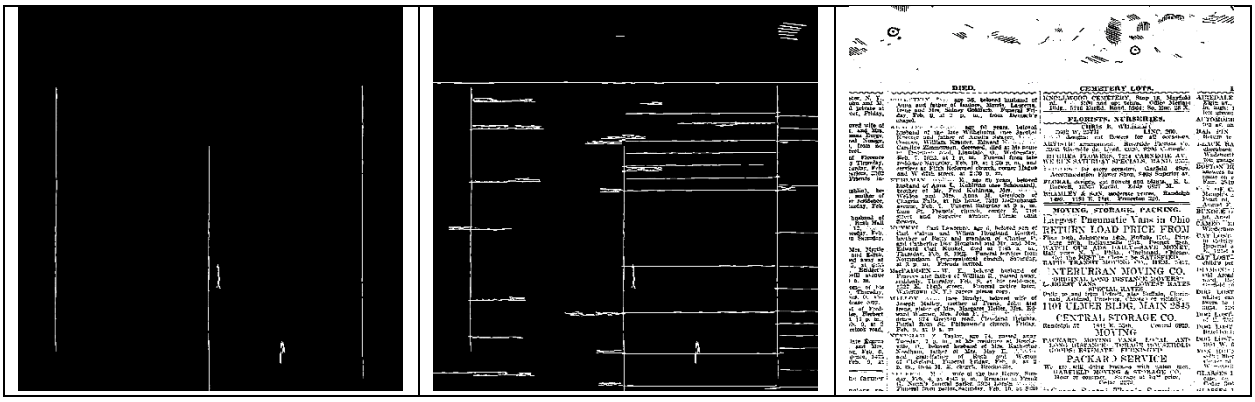


Figure 4: Panels show the steps of line removal. Note that the smoothed images are inversed to help demonstrate effects from steps. Top left show results from clutter removal after vertical smoothing (inversed) while top center show results from vertical smoothing removal after horizontal smoothing (inversed). Top right shows the difference of vertical smoothing and horizontal smoothing. Bottom left shows threshold results from vertical line identification. Bottom middle shows results for vertical and horizontal line identification. The final panel shows results from line removal.

Noise Removal

The last step in removing unwanted features is noise removal. This is the removal of unwanted features smaller than a character size. We remove noise using two techniques; median filtering and "closing." The median filter eliminates "salt and pepper" noise, spurs, and other edge related noise. By dilating a number of times features are eliminated just as we did before in clutter removal, accept this time we only dilated once. Then, with erosion, remaining features are returned roughly to their original shape. The returned shape doesn't have to be perfect, just close enough. This effectively closes off small features like lingering line segments. The dilation process is also only done once.

Possible Improvements: An initial assessment could be done to determine the quantity and type of noise to do more targeted noise removal.

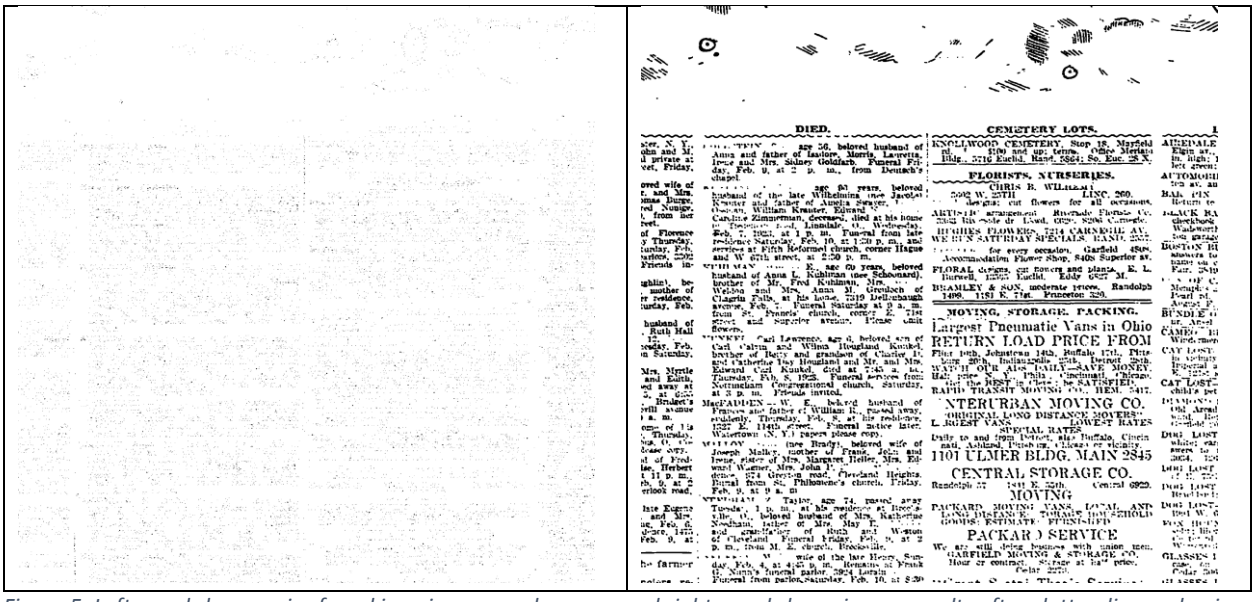


Figure 5: Left panel shows noise found in noise removal process and right panel shows image results after clutter, line and noise removal.

Article Zoning

Again, the ideal result for automated zoning from a newspaper page, beyond just OCR training, would a set of zones containing a set of characters that represent the whole or part of an article. However, the process described follows the criteria described in the **Zoning Criteria** section as needed for training OCR systems. As mentioned before, in moving forward, a way of scoring zoning will need to be developed to show accuracy and improvement.

The outline for the article zoning process is as follows:

1. Resize the image to a quarter of the size.
2. Word zone creation, used for article boundary correction.
3. Article zone creation.
4. Zone creation using word and article blobs.

Resizing

The process of resizing an image to extract characteristic from the image is known as scale-space theory [3] or hierarchical image segmentation [4]. Resizing may seem like an odd thing to do right off, but resizing preserves a scaled approximation of the features and the location and proximity to one another. By resizing the image, usable features can be analyzed in a way not applicable to the full resolution image. With Octave, resizing is done using a bilinear process.

Note: To return to a binary image the resulting image had a threshold performed on at with $t = 0.25$.

Word Zone Creation

The reason word zones are important is because they are created using early information about the image and help later with the correction of article zone boundaries which will be explained later.

With the newly resized image, word blobs are created as follows:

1. Using a 4-point erosion characters begin to blob together. In reference to the images we used, this was accomplished in a single step for most images.
2. The boundaries for the word zones are then found using the maximum and minimum values for the rows and columns occupied by the blob.
3. Some error is corrected by finding the unions of overlapping zones. The manner we used to classify if two zones should be merged is not a robust solution and could be improved. However, the rule we used was, if the boundaries of two zones overlapped by 1/12 of the smaller zone size then these zones where merged and new zone boundaries where produced from the over maximum and minimums of the boundaries of the two zones.

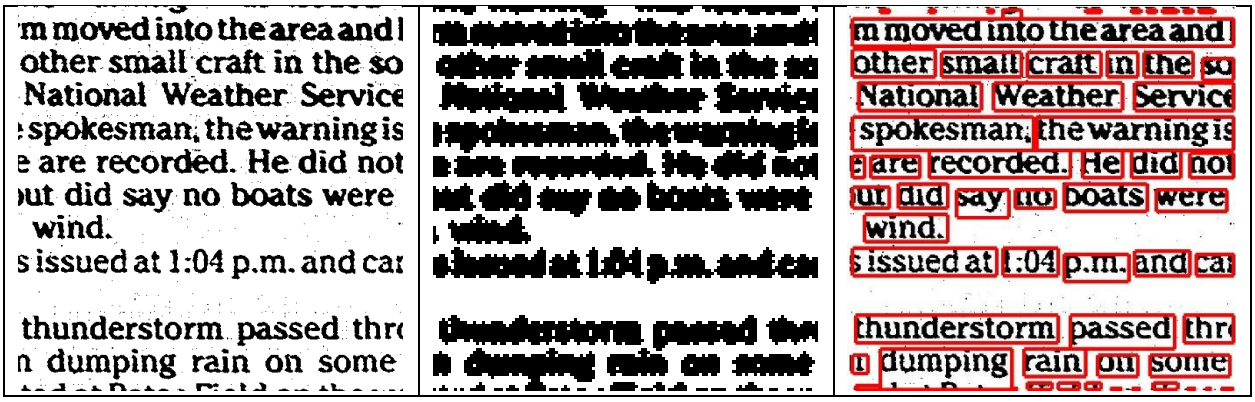


Figure 6: Left image shows original snippet from newspaper. Center image shows results from creating the word blobs. Right images results from creating word zones from word blobs superimposed over original snippet.

Article Zone Creation

The creation of article zones was a little more difficult to complete and took a lot of experimentation to get it to the point that it's at. This is due to trying to prevent bridging between blobs. The process is as follows:

1. A series of vertical 2-point erosion steps. This is the erosion of pixels x2 and x7 as shown in panel d of Figure 2. The construct of most articles takes shape of a brick lattice formed by the word blobs as a result of step 1 in section Word Zone Creation. This lattice has the properties of homogeneous vertical spacing and heterogeneous horizontal spacing (see right panel in Figure 6). The rows of word lines generally overlap from one row to another. By using 2-point vertical erosion, the overlapping word blobs will connect from text line to text line. In our code, 2 point-erosion is done twice (Figure 7, left panel).
2. An inverse "hole fill". A "hole fill" is when all set of connected black pixels of a binary image that are completely surrounded by white pixels are made flood filled white. The reverse is when a set of white pixels complexly surrounds by black pixels are flood filled black. This was done as a pre-step to the closing and helps cut down on overlap correction later on (Figure 7, left-center panel).
3. Closing. As explained in Word Zone Creation, this is done using a series of dilation steps and then erosion steps. This process serves as a way of eliminating unwanted bridging between article blobs and some unwanted edge noise that may cause unwanted article zones merges. Our code used 5 dilation steps and 5 erosion steps (Figure 7, right two panels).
4. The boundaries for the preliminary article zones are then found. As described in section step 2 of Word Zone Creation
5. Zone overlap correction. As described in section step 3 of Word Zone Creation

Figure 7: From left to right: Results from 2-point erosion preformed on the word blobs; Results after from inverse image holes filling; Results from image dilation; Results from image erosions and final image used from creating primary article zones.

Article Zone Correction

This is the final step and is used to correct the edge errors created in the preliminary article zone creation during the closing step.

1. A zone overlap correction is done as in sections Word Zone Creation and Article Zone Creation, however this time, the word zones are analyzed to see if they overlap the article zones and are set aside as sets of word zones.
2. New zone boundaries where produced from the over maximum and minimums of the boundaries of the word zone sets.
3. Zone overlap correction. As described in section step 3 of Word Zone Creation

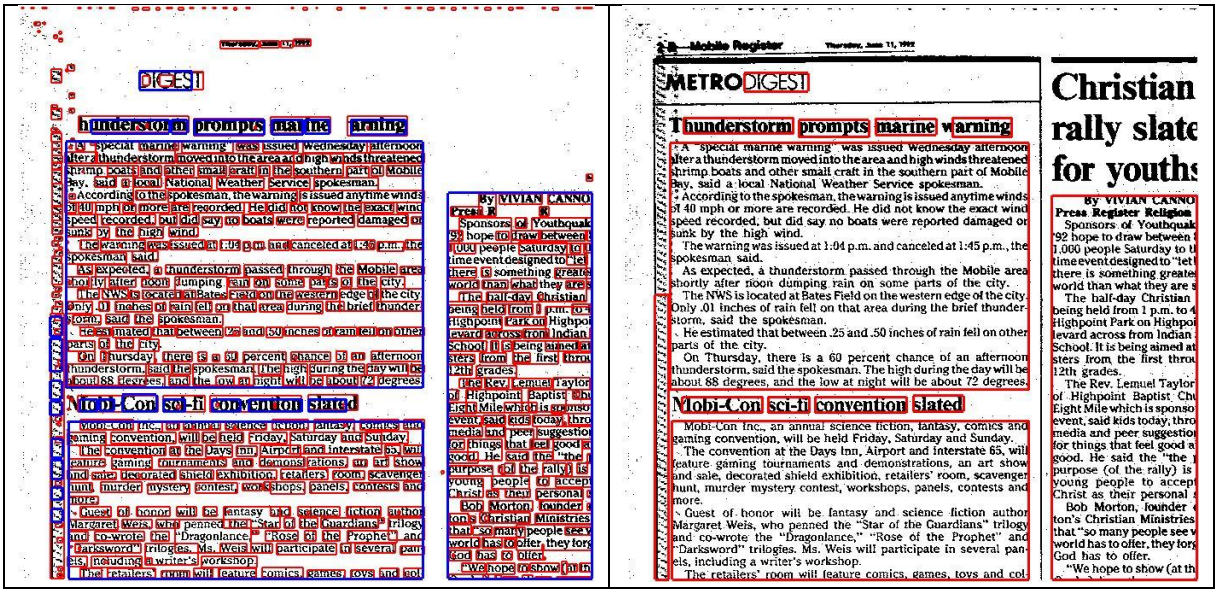


Figure 8: This figure shows a larger snippet from the same image the snippets came from in Figure 7. Left panel shows word zones in red and preliminary article zones in blue. These are superimposed on results after clutter and noise correction. Right panel shows final article zones after correction using word zones. Note the clutter removal also removed almost all of the bold titles, another example of possible improvements.

Assessment and Future Work

This process worked really well for the purpose of producing zones for OCR training. We are pleased with the results so far. It was used to produce the images from NewsBank and Chronicling America that

were sent to be transcribed December 2014 and produce 29,000,000 transcribed characters which will be used for to train OCR systems. Without it, there would have been no way of creating that many zoned images under the time restraints given. The transcription will be used in robo-typing research to automatically produce and provide information from obituaries to be searchable in FamilySearch.

There are many improvements that could be made to this process. The creation of a scoring system for zoning was mentioned before and there are possible improvements mentioned in some sections. Other improvements include the fat that some of the parameters were hard coded and because of this, some of the images produced not usable results. Also, this process was done on the fly and all potential solutions for each step may not have been vetted. Though the overall outline of the process seems solid, most of these improvements will, hopefully, be addressed as research on auto-zoning is continued to assist robo-typing.

References

[1] B. Gatos, K. Ntirogiannis, and I. Pratikakis, “ICDAR 2009 Document Image Binarization Contest (DIBCO 2009)”, Proc. ICDAR2009, Barcelona, Spain, July 2009.

[2] A. Antonacopoulos, D. Bridson, C. Papadopoulos and S. Pletschacher, “ICDAR2013 Competition on Historical Newspaper Layout Analysis – HNLA2013”, Proc. ICDAR2013, Washington DC, USA, August 2013.

[3] A. P. Witkin, "Scale space filtering," Proceedings of International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, pp. 1019 – 1021. 1983.

[4] P. J. Burt, T-I Hong and A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-11, No. 12, pp. 802 – 809, Dec, 1981.