535046
Page content detection for the purposes of achieving accurate cropping

Printed book, magazines and documents are being digitized for the purpose of creating digital copies for electronic delivery, print on demand and other purposes. These items are scanned using scanners or digital cameras. Regardless of the scanning process used the resulting scanned pages end up with lots of artifacts including content from the adjacent page, shadows on the edges, finger prints on edges, edge lines, paper defects such as blotches, tears, or creases, clamp marks etc.

The scanned images with the above mentioned artifacts need to be cleaned up before they can be presented on the web or printed. Cleaning of scan

The idea behind the technique we propose is to eliminate artifacts from scanned pages that occur outside of the actual content of a given page. Current techniques involve identifying noise elements and applying color correction algorithms to remove the noise. Sometimes it is very difficult to identify the noise from content and even when noise is detected it is not always possible to remove the effects of this noise completely.

FIG. 1 shows some example scanned pages which illustrate the various artifacts introduced during the scanning process. These images have already been cropped at the boundary of the page edge, which is always done prior to our content detection.

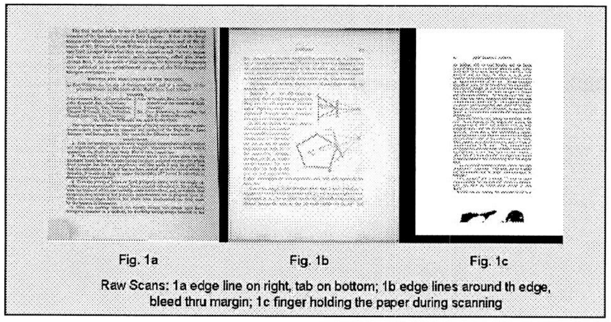

Fig. 1a Fig. 1b Fig. 1c

Raw Scans: 1a edge line on right, tab on bottom; 1b edge lines around th edge, bleed thru margin; 1c finger holding the paper during scanning

FIG. 1: Example pages already cropped at page boundary.

The idea is to detect the content boundary and crop the page at the detected content boundary essentially eliminating the noise that exists outside the content boundary. This has the advantage that we do not rely on correcting the detected noise but completely eliminating it. This results in much cleaner scanned pages. FIG. 2 shows an example of a scanned image which has some artifacts introduced during the scanning process and how they are cleaned up.

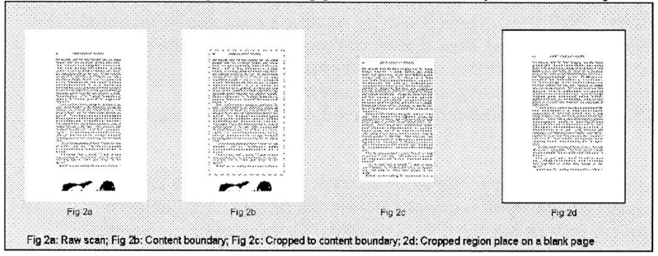

Fig 2a Fig 2b Fig 2c Fig 2d

Fig 2a: Raw scan; Fig 2b: Content boundary; Fig 2c: Cropped to content boundary; 2d: Cropped region place on a blank page

FIG. 2: Example of artifacts introduced during scan, and how they are cleaned up.

Clearly the processed page is without artifacts and is completely cleaned up. The key challenge then is to detect content that belongs to a given page.

The process we use to detect content boundary given a collection of scanned pages involves the following fundamental steps, each of which is discussed in subsequent sections. The process makes multiple passes over some or all of the book pages, and will iterate over various steps to obtain optimal results.

1. Start with an image cropped at the physical page boundary.
2. De-skew content (crop image at rectangle aligned with content).
3. Segment the image [1,2] (separate background from non-background).
4. Group related segments [3], determine boundary, crop image.
5. Extract and utilize additional information and indicators from the image data.

1. Start with an image cropped at the physical page boundary.

The input to our content detection process is a collection of images that have previously been cropped at the page boundary (e.g., via automated page detection process, or external crop information). This eliminates most everything outside of the page, such as the book cover, page stack edges, and the background surface, as shown in FIG. 3.

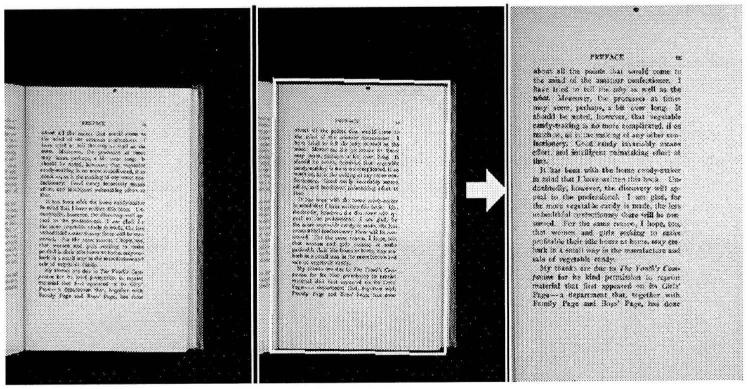

FIG. 3: Raw image (left), page boundary (middle), and image cropped at page boundary (right).

2. De-skew content (crop image at rectangle aligned with content).

Content skew is determined by analyzing horizontal projection profiles across the image while rotating the image through small angles. Projections that align with lines of text will have larger variation across the profile than those that misalign. Once the skew is calculated, a rectangle rotated by this skew is fit inside the page boundary, and the image is cropped at this rectangle, as shown in FIG. 4.

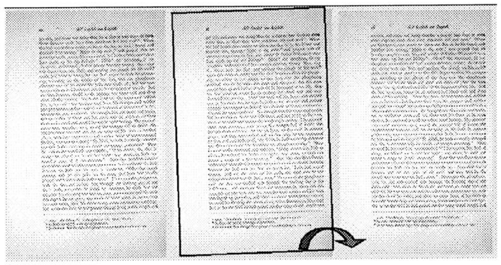

FIG. 4: Image cropped at page boundary (left), determine skew, fit skewed rectangle (middle), crop at this skewed rectangle (right).

Although this approach works well for single column text pages, other methods are required for multiple columns or mixed text and graphics/images. FIG. 5 shows a well aligned page (left) that was incorrectly de-skewed (right) due to two columns where the lines and spaces misalign across the page. This common situation can cause the projections with largest variation to occur at angles that align with neither column.

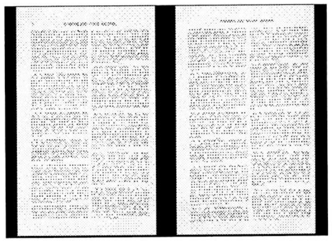

FIG. 5: Well aligned page (left), incorrectly de-skewed (right) due to two misaligned text columns.

Better results are obtained by analyzing each column separately, along with the full page. FIG. 6 shows the original image (left), the splitting of this into two images, one per column (middle left, middle right), and the result of basing de-skew on analysis of each column along with analysis of the full image (right).

Although this works well for text, different approaches must be used for pages with mixed text and graphics or images, or pure graphics or images, as shown in FIG. 7. Our skew detection algorithm, described above, detects when it fails on pages such as these, and can then fall back to other skew detection means. These include locating areas of pure text, and using those, or searching for minimal bounding box, or searching for rectangular edge lines using an approach similar to our page detection algorithm (Sec 1).

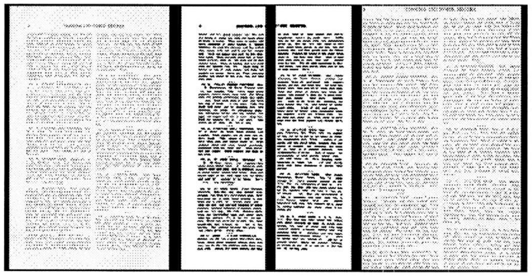

FIG. 6: Well aligned page (left), split into separate images for left and right columns (middle left, middle right), and improved de-skew result (right) as compared with FIG. 5

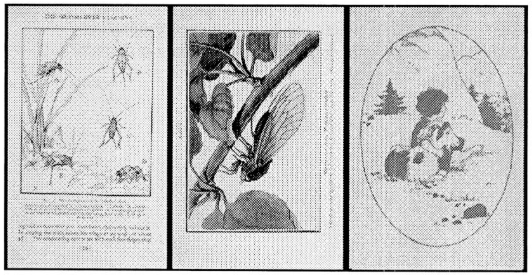

FIG. 7: Examples of mixed text and graphics or images. The text skew and graphics skew can differ (left), the text may be vertical (middle), or there may be no text and no rectangular box at all.

3. Segment the image (separate background from non-background).

The segmentation process involves the following steps, shown in FIG. 8 and FIG. 9:

- Apply a Gaussian smoothing function, FIG. 8 (left).
- Compute the gradient, FIG. 8 (middle).
- Remove excessive noise, FIG. 8 (right).
- Apply a watershed segmentation with dynamic threshold settings, FIG. 9 (left).
- Remove extreme segments (e.g., at extreme locations such as edges, or of extremely small size, or of extreme extent), FIG. 9 (right).

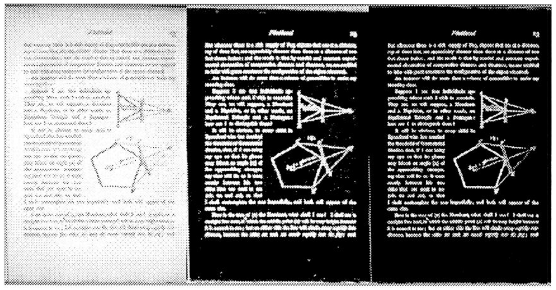

FIG. 8: Gaussian smoothed image (left), gradient (middle), and with noise removed (right).

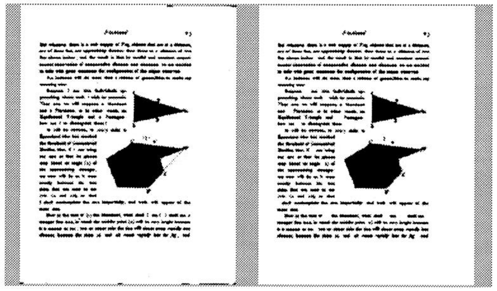

FIG. 9: Watershed segmentation (left), and with extreme segments removed (right).

4. Group related segments, determine boundary, crop image.

The process of grouping the segments to produce a final rectangular boundary involves the following steps, shown in FIG. 10 and FIG. 11:

- Group segments horizontally, FIG. 10 (left).

- Group segments vertically, FIG. 10 (middle).
- Keep largest group(s).
- Find the border pixels of these groups.
- Fit minimum rectangular boundary to these border pixels, FIG. 10 (right).
- Crop image at this boundary, FIG. 11 (right).

This result is not optimal since the bottom and left sides contain excessive page that is clearly not part of the real content. This is due to noise that was incorrectly included in the segmentation and grouping.

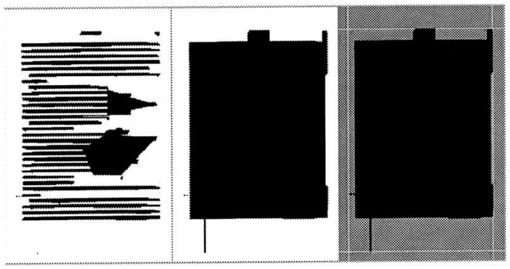

FIG. 10: Horizontal grouping (left), vertical grouping (middle), minimal bounding box (right).

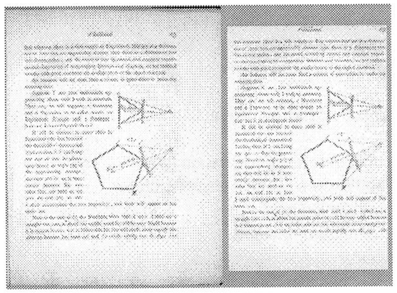

FIG. 11: Original image (left), and result of cropping this at bounding box from FIG. 10 (right).

5. Reducing the impact of noise.

To reduce the noise included in the segmentation, a higher threshold can be used. Then, to reduce the noise included in grouping of segments, the removal of extreme small segments can be done more aggressively.

Horizontal grouping of this new segmentation is shown in FIG. 12 (right), with the original result (left). Note, in the new result, the lower left side noise spike (jutting out to left) is gone, and the very tiny segment at bottom left (near the page bottom edge) is also gone. Vertical grouping is shown in FIG. 13 (right), with the original result (left). Note, in the new result, both of the lower left "noise jetties" are no longer present. As shown in FIG. 14, the final content boundary now no longer includes those noise jetties, and the resulting final cropped image is now improved, being much tighter against all sides of the real content.

Thus, more aggressive "noise removal" (e.g., higher threshold, more aggressive extreme segment removal), can tighten up a boundary, the comparison shown in FIG. 15. However, this alone can cause valid content to be cut off, since not all small, sparse segments represent noise. They often represent content. More information is needed.

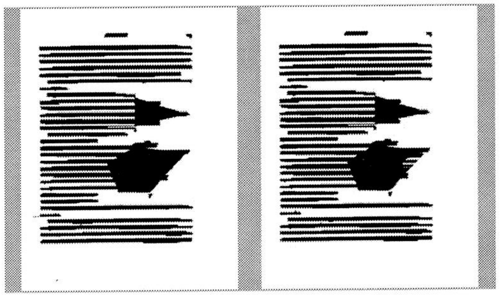

FIG. 12: Original horizontal grouping (left), and new grouping using result (right).

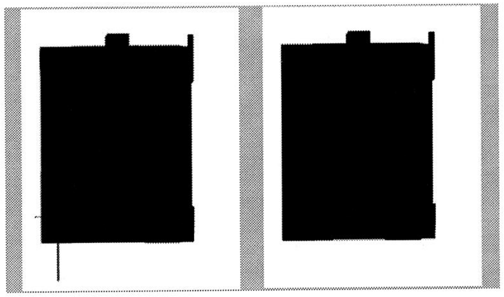

FIG. 13: Original vertical grouping (left), and new grouping using result of FIG. 12 (right).

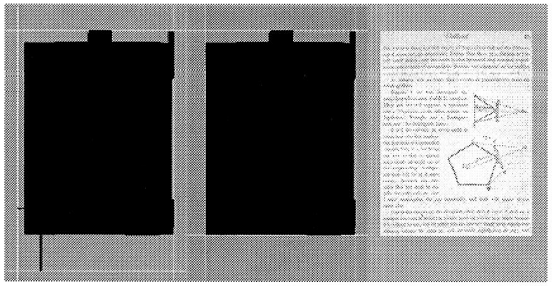

FIG. 14: Original final boundary (left), new boundary (middle), and new final cropped result (right).

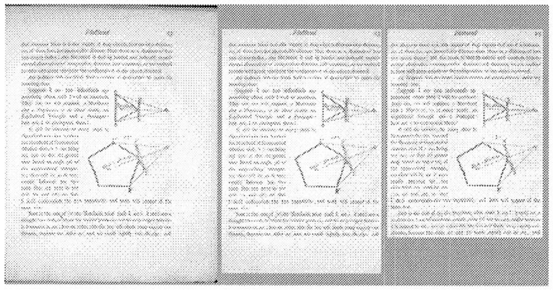

FIG. 15: Original page (left), original cropped result (middle), new tighter cropped result (right).

6. Extracting and utilizing additional indicators

Additional information and indicators are required to correctly distinguish content from non-content on each side of the content boundary. For example, the bottom side might include unwanted page noise, while the left side might crop valid content. This can be explicated as the need to address the following issues:

- How to determine if a given side of a content boundary is "reasonable" or not?
- How to determine if more, or less, aggressive noise removal would help?
- The right/wrong decision can make that side better/worse.
- Either decision can change other sides.
- How to determine if the new result is better or worse than the old result?

To address these issues, our algorithm further analyzes the image data to extract numerous other types of information and indicators, which are integrated and utilized in concert to arrive at the final content boundary. This includes multiple methods of calculating content boundary, analysis and comparison of these results, iterations with more aggressive noise removal to improve content boundary, multiple passes over book pages, and statistics gathered from all of these. We define ***margin*** as the distance between the edge of the box-in-page boundary and the content boundary on that *margin side*.

6.1. Phase 1 (first pass over book pages)

The following are performed on each page:

- The content skew and rotated box-in-page boundary is calculated.
- Margins are calculated using multiple methods:
- ***SEG Margins*** are calculated using the previously described process of segmentation, grouping, and bounding box detection.
- ***PS Margins*** are calculated from projection profiles of segmented image data.
- ***PPV Margins*** are calculated from projection profiles of the pixel variance (***PPV Profiles***) of the raw image data.
- Each margin includes a strength (or confidence) measure, calculated from the PPV Profiles of the data used to create that type of margin
- These results are analyzed to determine if SEG Margins should be recalculated with stronger noise removal. If so, this step is repeated, up to N times.
- Since all margins are relative to the box-in-page dimensions, which depend on the box-in-page skew, raw margins cannot be compared across pages. For this, a set of normalized margins is created, referenced back to the original page boundary.
- The difference between the normalized margins and the resulting content dimensions for the various content boundary calculation methods is calculated.
- All of the above results for each page are saved for analysis. This includes original page dimensions, box-in-page skew and dimensions, raw and normalized margins and content dimensions from each content boundary calculation methods, differences, etc.

The following are performed after all pages have been processed as above:

- Statistics are calculated, including min, max, mean, median, average deviation (***ADEV***), and standard deviation (***SDEV***), for related subset of the above data. These subsets include:
- Each margin (top, right, bottom, left) from each calculation method.
- Each content dimension (width, height) for each calculation method.
- Each normalized margin for each calculation method.
- Differences between normalized margins of different calculation methods.
- Differences expressed as SDEVs and ADEVs out from the average differences.
- The above are grouped by all pages, left side pages, and right side pages.
- Each page result is compared with related statistics to detect possible extreme margins or content dimensions (e.g., page width vs. mean and max width for all pages, or each page margin vs. mean and min margin for the same side pages).
- Based on the correlation of per-page related data (e.g., width, right margin, left margin), extreme margins or content dimensions are flagged.
- Statistics are then recalculated without the extreme data.

6.2. Phase 2 (second pass over book pages)

Phase 2 is similar to Phase 1, except that margins are analyzed and adjusted based on the data and statistics from Phase 1, and no further statistics are calculated at the end.

The following are performed on each page:

- The content skew and rotated box-in-page boundary is calculated, now using more information from Phase 1.
- Margins are calculated using multiple methods:
- *SEG Margins* are calculated.
- *PS Margins* are calculated.
- *PPV Margins* are calculated.
- Each margin includes a strength (or confidence) measure.
- These results are analyzed to determine if SEG Margins should be recalculated with stronger noise removal. If so, this step is repeated, up to N times.
- Margins are adjusted based on margin strengths and information from Phase 1.
- Margins are further adjusted to avoid intersection with content.
- Page is cropped at final margins, eliminating artifacts outside of the content.

7. PS Margins and PPV Margins

PS Margins are calculated from projection profiles of segmented image data. The image is first processed as for SEG Margins, using least aggressive noise removal, and without the grouping steps. This includes Gaussian smoothing, gradient, noise removal, watershed segmentation, and removal of extreme segments, . Horizontal and vertical projection profiles are created from the binary image, where each projection ray is the sum of the binary data along that ray. PS Margins are calculated from these profiles by looking for transitions based on profile statistics, partial profile statistics (e.g., edge areas only), and running window statistics while stepping across the profile.

PPV Margins are calculated from projection profiles of the pixel variance (PPV Profiles) of the raw image data. The raw image is converted to grayscale and reduced in resolution. Horizontal and vertical projections profiles are created from this grayscale image, where each projection ray contains statistics of the pixel data along that ray. Of particular interest are the standard deviation and mean. FIG. 16 (left) shows the PPV Profiles (standard deviations) for raw image data. PPV Margins are calculated from these profiles using an approach similar to that described for PS Margins above.

Margin strength is calculated from the PPV Profile of the data that was used to create that type of margin, and is based on the gradient of the PPV Profile margin transition. Thus, SEG Margin strength is based on PPV Profiles of the segmented and grouped binary image data, PPV Margin strength is based on PPV Profiles of the least aggressively segmented binary image data, and PPV Margin strength is based on the PPV Profiles of the raw image data. FIG. 16 (right) shows the PPV Profile of binary data for PS Margins.

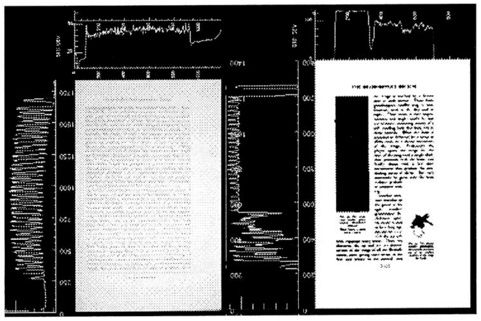

FIG. 16: Profiles of pixel variance for raw image data (left), and binary segmented data (right).

Since PPV Profiles and PPV Margins are created from the raw image data, edge noise can degrade the results. FIG. 17 (left) shows the PPV Profiles (SDEV in yellow) and resulting PPV Margins (red) of a page with top and bottom edge noise. With so little text of so low contrast, the top and bottom edge noise has a strong impact on the PPV Profiles along both the vertical and horizontal projection rays. On the left profile (the result of horizontal projection rays), this is seen at the very top and bottom of the profile. On the top profile (the result of vertical projection rays), it occurs about 20% in from each side (where the edge noise starts horizontally). The top and bottom margin calculations, which use the left profile, can correctly ignore this as edge noise. However, the left and right margin calculations, which use the top profile, interpret this as content (being too far in to be edge noise) and use those transitions for the margin locations.
To deal with this common situation, our algorithm uses the initial PPV profiles to first detect any such edge noise; using the left profile to detect top or bottom edge noise, and the top profile to detect left or right edge noise. It then creates a boundary just inside of the noise, and creates the final PPV Profiles (from which margins will be calculated) from the data within that boundary. This result is shown in FIG. 17 (right). By removing top, bottom, and right edge noise regions, the PPV Profiles now show correct content transitions, and the resulting PPV Margins match the content well.

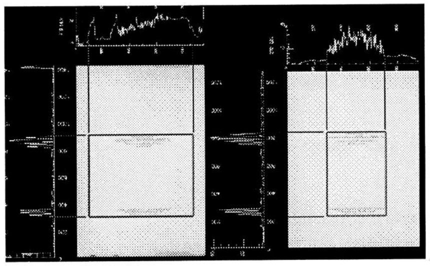

FIG. 17: PPV Profiles of image including edge noise (left), and with edge noise removed (right).

8. Effective use of these additional indicators

The degree to which these indicators can help improve content detection depends on how well they agree on the outcome (what needs to be done), coupled with their strengths. For example, we might have high confidence if all indicators agree and are strong, or if most indicators agree and are strong, and the ones that disagree are weak. But what if half the indicators agree on one outcome, and half on the opposite outcome, yet all are strong? Or, what if most indicators agree on one outcome, but are weak, while the remaining fewer indicators agree on the opposite outcome, but are strong? Also, the intrinsic "value" of each indicator comes into the equation. This is a qualitative measure of how much the indicator is trusted, based on the processing and analysis of many books. FIG. 18 presents a simple example of indicator correlation issues:

- The original image (left) has sparse text, low contrast, and edge noise on the top and bottom, including clamps.
- The PS Margins (middle left) are accurate for all sides (margins shown on top of the segmented binary data used to create them).
- The SEG Margins (middle right) are accurate for top, left, and right, but the bottom cuts off the lower text block (margins shown on top of the segmented, grouped binary data used to create them).
- The PPV Margins (right) are accurate for top and bottom, but not for left and right (margins shown on top of the grayscale raw image data used to create them).
- Neither the PS Margins nor the SEG Margins were impacted by the edge noise since they are created after removing extreme segments, which included the edge noise.

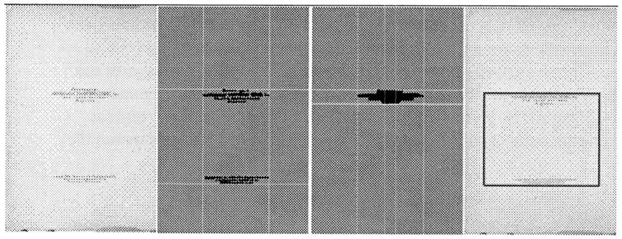

FIG. 18: Original page (left), PS Margins (mid left), SEG Margins (mid right), PPV Margins (right).

In this example, simple majority agreement will yield good results:

- Top margin: All 3 agree on one that is accurate.
- Bottom margin: 2 of 3 (PS and PPV) agree on one that is accurate.
- Left and right margins: 2 of 3 (PS and SEG) agree on ones that are accurate.

In general, majority agreement is insufficient. Our algorithm employs a comprehensive analysis of indicators in the decision process, considering numerous comparisons of deviations from mean values, or median values if extremes should be excluded. In general, if the parameter is expected to converge well (e.g., max content dimensions, left margins) the SDEV is used, otherwise (e.g., right margins), the ADEV is used.. This analysis includes the following, where 'A' and 'B' can be any of SEG, PS, PPV:

- Comparing A Margin with B Margin.
- Comparing difference between A Margin and B Margin with median difference of these margins for same side pages.
- Difference between A margin and average A margin for same side pages.
- Difference between A margin and minimum A margin for same side pages.
- Difference between A height and max A height for all pages.
- Difference between A width and max A width for all pages.
- Assessing whether margin result improves after iteration.
- Reverting margins that have degraded.

. Conclusion

The technique described above has been implemented and has proven to be very effective in cleaning up scanned documents.

[1] Linda G. Shapiro and George C. Stockman (2001): "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[2] Jian Fan, "Enhancement of Camera-captured Document Images with Watershed Segmentation", CBDAR07, p.87-93, Sept. 2007

[3] K. Y. Wong, R. G. Casey and F. M. Wahl, "Document analysis system", IBM J. Develop. Vol. 26,No. 6, Nov. 1982

Disclosed by Hewlett-Packard Company, Prakash D. Reddy, Jian Fan, Michael P. Zeleznik
535046