

A Robust Pointer Segmentation in Biomedical Images Toward Building a Visual Ontology for Biomedical Article Retrieval

Daekeun You, Matthew Simpson, Sameer Antani, Dina Demner-Fushman, George R. Thoma

National Library of Medicine, National Institutes of Health, Bethesda, MD 20894

ABSTRACT

Pointers (arrows and symbols) are frequently used in biomedical images to highlight specific image regions of interest (ROIs) that are mentioned in figure captions and/or text discussion. Detection of pointers is the first step toward extracting relevant visual features from ROIs and combining them with textual descriptions for a multimodal (text and image) biomedical article retrieval system.

Recently we developed a pointer recognition algorithm based on an edge-based pointer segmentation method, and subsequently reported improvements made on our initial approach involving the use of Active Shape Models (ASM) for pointer recognition and region growing-based method for pointer segmentation. These methods contributed to improving the recall of pointer recognition but not much to the precision. The method discussed in this article is our recent effort to improve the precision rate. Evaluation performed on two datasets and compared with other pointer segmentation methods show significantly improved precision and the highest F_1 score.

Keywords: Biomedical image analysis, biomedical article retrieval, content-based image retrieval, image overlay segmentation, image binarization

1. INTRODUCTION

Biomedical images are critical in establishing diagnoses, acquiring technical skills, implementing best practices for education, and many other areas of medicine. This has drawn attention toward developing retrieval engines for biomedical images and articles and on related research topics [1]. Most retrieval approaches are text-based, extracting relevant information from citations, figure captions, and/or text discussion. Visual content extracted from images has also been recently used to achieve better retrieval. Most of the approaches that show improvement, however, use visual content combined with text information, since visual content alone is inadequate due to the “semantic gap” reported in the literature [2].

In [3] we proposed a new article retrieval approach as a potential solution to improve techniques based on conventional text or content-based image retrieval (CBIR). We focus on local image regions of interest (ROIs) instead of the entire image based on our hypothesis that local image regions indicated by pointers (arrows and symbols) may contain more relevant information than other regions in the image. Toward developing the approach, we developed algorithms that recognize various overlaid pointers in biomedical images and identify their corresponding ROIs [3–5].

Recently we reported preliminary results of our ongoing research effort on building a visual ontology for a biomedical retrieval system [6], which utilizes pointers and ROIs in biomedical images. In the proposed system, pointers are detected and their corresponding ROIs are then linked with textual concepts extracted from figure captions by combining rule-based and statistical natural language processing techniques. Textual information includes type of pointers (“arrows”, “arrowheads”, “asterisks”, etc.), pointer attributes (color, size, plural, etc.), and descriptions of the ROIs. The pointer recognition algorithm proposed in [3] was applied to automatically extract visual ROIs from images.

A key improvement necessary in our methods was in the precision of pointer recognition. Low precision (high false positive) made it difficult to select the true pointer(s) and pair them with correct textual concepts. Here we report on our success in improving the precision by a new pointer segmentation algorithm. Previously, our MRF-based recognition algorithm [3] achieved an average of about 80% recall but much lower precision (about 25%), when it is combined with our edge-based segmentation [4]. Subsequent algorithms incorporate noise removal,

but they yield very minor improvement. Recently we applied thresholding-based pointer segmentation in [7], which achieved better precision but lowered the recall than our edge-based segmentation. Our primary goal in this research is to achieve higher precision while retaining the high recall of our edge-based method.

The remainder of this article is organized as follows. Section 2 provides a summary of our prior pointer segmentation methods. Section 3 describes our new method, and evaluation results and discussion are given in section 4. Section 5 provides conclusions and discusses future work.

2. PRIOR WORK

Our previous pointer segmentation methods utilized both pointer boundary contour and body region. Our observation identified two general pointer properties; pointers generally have i) sharp contour edges and ii) homogeneous body intensity. We developed an edge-based segmentation method to take advantage of the first property. Our edge-based method that applies edge detection (e.g., Roberts, Sobel, or Canny operators) first and then binarizes the edge image by adaptive thresholding produces fairly clear pointer boundaries and shows robust performance; however, it sometimes fails to segment pointers with weak edges and the failure eventually results in recognition failure. Figure 1(b) shows an example pointer with weak edges. The pointer seems to have clear boundary, but the circled boundary part has relatively lower edge intensity than other part and disappeared in the final result.

Our next method utilized the second property and tried to segment a homogeneous intensity area by region growing from seed regions found by the edge-based segmentation. This method attempted to amend distorted pointer boundaries detected by the edge-based segmentation. It achieved higher precision and recall than edge-based segmentation; however, computation time was dramatically increased due to the region growing.

Cheng et. al. [7] segmented arrow candidates by global thresholding-based method followed by edge detection. A global threshold was computed by Otsu’s method and edge detection and several preprocessing were performed to eliminate noisy candidates. This approach showed slightly lower recall but achieved significant improvement in precision compared to our edge-based method. One drawback of this method is that it fails to segment pointers with gray intensity less than the global threshold which is obtained by taking a greater value between Otsu’s threshold and a fixed value (e.g., 200). Such pointers often have high contrast boundary edges and hence may be successfully segmented by considering edge features.

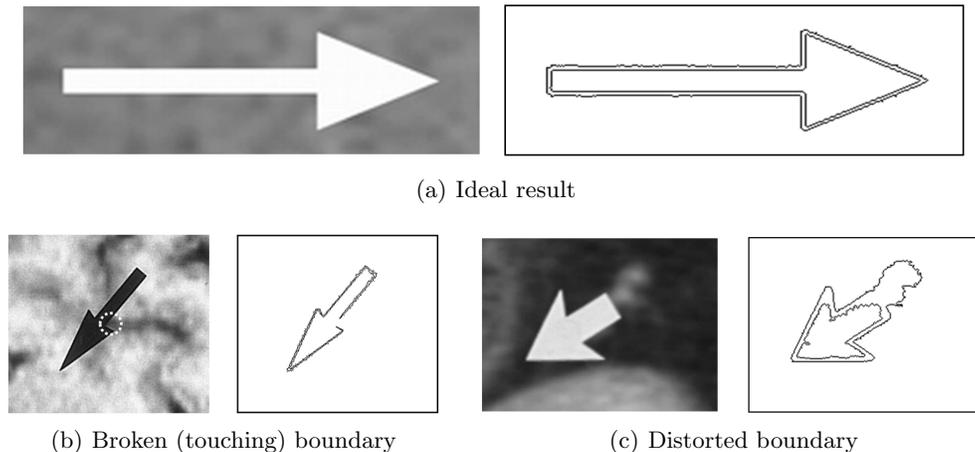


Figure 1. Examples of detected pointer boundary contour

3. METHOD

In this article a new pointer segmentation method is proposed to solve several drawbacks of our prior work and address the reported challenges in pointer segmentation. Obtaining a good quality single contour for each pointer

is another goal of the proposed method. Our prior edge-based method produces two separate contours (Figure 1(a)). In case those two are touching or severely distorted, neither of them can be used for recognition. Figure 1 (b) and (c) show samples of such cases, respectively.

Figure 2 depicts our proposed method. It has two separate paths to detect pointer candidates from pointer boundary contours and body regions. Then the two results are merged to find concrete candidates and threshold values for final pointer segmentation. Color images are converted to 8-bit grayscale images by an equation discussed in [4].

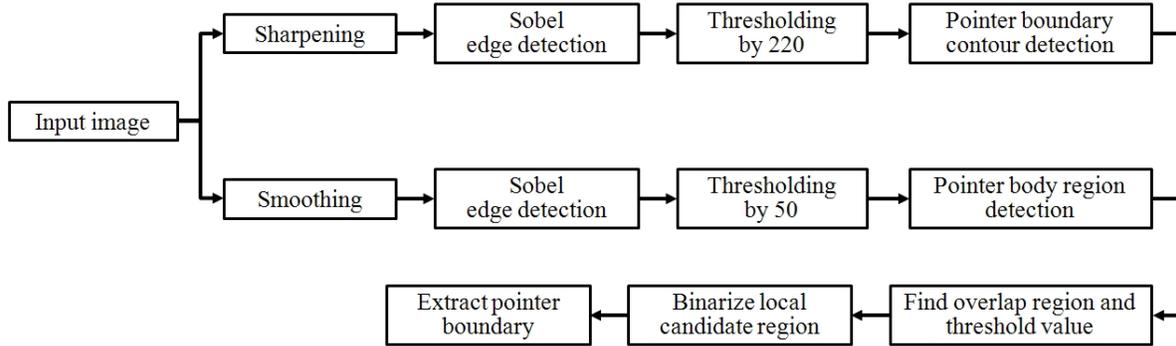


Figure 2. The proposed pointer segmentation method

3.1 Pointer candidate detection by boundary detection

Our prior work showed a weakness in detecting pointers with weak edges, resulting in broken boundaries. This is shown in Figure 1(b) where most of the boundary parts are well detected while a small portion is missing. Different thresholding values or methods or applying any post-processing to amend broken boundaries could be a good solution for this sample; however, eventually we may not be able to avoid broken boundary problem through any edge- or binarization-based approaches. It is our observation, however, that even though edge-based segmentation may be weak, it is still a powerful method for pointer boundary segmentation and it is worth considering post-processing methods to overcome this weakness.

Instead of applying adaptive thresholding to the edge detection output (Figure 3(b)), we use a global threshold to obtain a binary image (Figure 3(c)). This may result in more broken boundaries than obtained by adaptive thresholding; however, our main goal here is to detect evident (entire or partial) pointer boundary candidates by finding high intensity pixels in edge images. In order to enhance detection, image sharpening (a high pass filter) is applied to the input image [8]. Figure 3(c) shows binarization result of the edge image of the input by threshold 220 . As shown, most of the pointer boundary is detected. Outer contours of the white connected components (CC) are then extracted as pointer boundary candidates by using the contour extraction function in OpenCV [9].

3.2 Pointer candidate detection by body detection

Most pointers overlaid in biomedical images have a body consisting of pixels of similar gray intensity (i.e., solid pattern). Detecting regions of homogeneous intensity could be a good start for pointer body detection. To this end, we apply a lower threshold (such as 50) to the edge image to detect regions with small edge magnitudes. Such regions mostly consist of pixels with similar intensity and their textures would be close to a solid pattern. In Figure 3(d) that is obtained by thresholding the edge image by 50 , black pixels represent regions satisfying the above condition.

Here another important condition needs to be considered to detect a pointer body from the binarization result shown in Figure 3(d). In the result white pixels correspond to pixels with abrupt intensity changes and they are in boundaries between any two distinct regions (in color) such as gray and black regions or pointer and background in the input. Hence detecting a black CC within (or surrounded by) a white CC can eliminate

most of the noisy regions that satisfy the first condition but are not true pointer bodies. Figure 3(f) shows only four contours of candidate pointer body regions. Most of the black pixels in Figure 3(d) consist of the largest background region that is not surrounded by white pixels and hence excluded as noise.

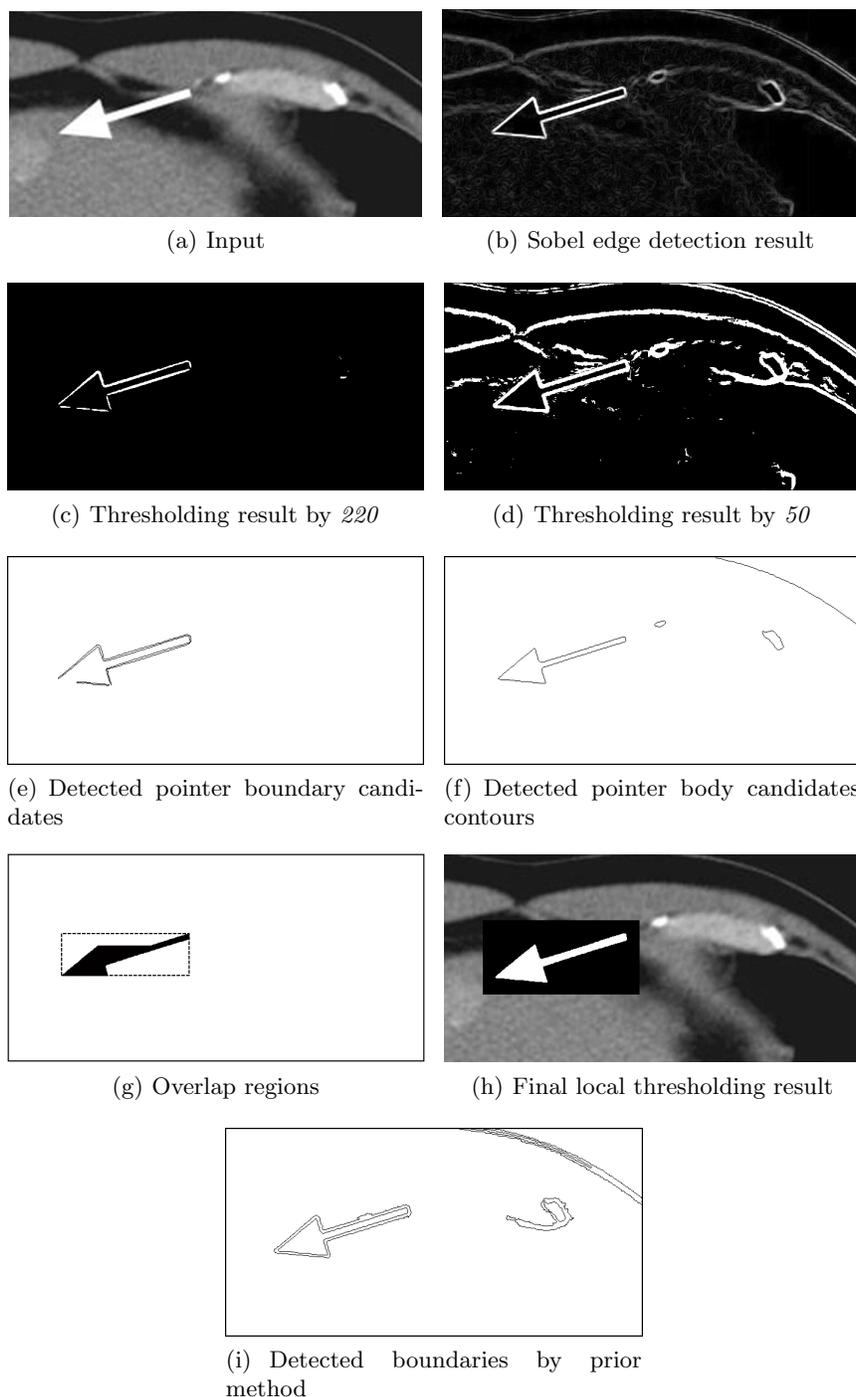


Figure 3. Intermediate result images of our new pointer segmentation method

Our proposed boundary and body detection methods are not greatly affected by pointer color itself since binarization is performed on the edge image where only sharp changes in pixel intensities (or brightness) in an

input image are captured. Those changes (edges) are mostly captured in the pointer boundaries but not in the pointer body and hence pointer color is not a critical factor. Pointers can be successfully detected as long as their body has clearly separable gray intensity from background, which is a common property of overlaid pointers in biomedical images.

3.3 Overlap region detection

As shown in Figure 3(e) and 3(f), one result has broken pointer boundary and another one has several noisy contours. Our previous edge-based method detected similar results with the body detection result in Figure 3(f), as shown in Figure 3(i). The arrow could be recognized by the inner contour, however, none of the detection results are satisfactory.

We observe that in the intermediate contour detection results there exist regions that can be found from both pointer boundary and body detection results. To be found in both results, a region should have both clear (high contrast) boundary and homogeneous intensity region surrounded by the clear boundary, which is a condition that satisfies both the aforementioned two ideal pointer properties. Candidates that satisfy only one condition, however, may not always be true pointers. Hence combining the two detection results and finding overlap regions could be an optimal solution to find true pointers with less noise.

In order to find overlap regions, we may simply compare bounding boxes of pointer boundary and body candidates (Figure 3(e) and 3(f)) and examine the overlap regions. Instead we scan each pixel in each row to find not only potential pointer region but also average intensity value of the region. This value is used as a threshold later to binarize the local candidate region to obtain a pointer contour. Figure 4 depicts our detection method. Assume that two contour images (Figure 3(e) and 3(f) in this example) are combined and contour points are distinguishable by their contour identification numbers (for example, pixel intensity values *10* and *11* for boundary and body contour points in Figure 4, respectively). In each scan line, the first and last contour points of pointer boundary or body contours can be identified as shown by solid and dotted arrows, respectively. Then pixels between the two points are counted to compute pointer area, and gray intensity of each pixel is examined to compute average gray intensity of pointer body region. This method, however, sometimes includes pixels outside the pointer body as shown in Figure 3(g). Those noisy pixels may affect the accuracy of the obtained average pointer intensity value; however, in most cases they do not ruin the detected pointer region bounding boxes (overlaid dotted rectangle in Figure 3(g)). Pixels surrounded by the pointer boundary and body contours (more precisely, pixels between any two boundary or body contour pixels in each row) are considered separately and two average intensity values are obtained. An average of the two average threshold values is used to binarize the local region examined. Figure 3(g) shows examined pixels for overlap region detection and threshold value computation.

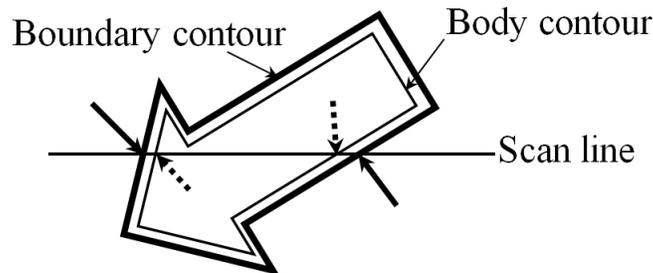


Figure 4. Illustration of overlap region detection

3.4 Local region binarization and pointer contour extraction

The threshold value obtained in section 3.3 can be assumed as pointer intensity (color) and used as a threshold value for binarization of the local region to segment a pointer from background. Some marginal space (e.g.,

20 pixels) is added to the outside of the detected region (e.g., dotted line box in Figure 3(g)) and local region binarization is performed on the final region. Figure 3(h) shows local binarization result and as shown in the result a single clear pointer boundary is obtained. A local region may have several contours besides a true pointer boundary, and in such cases the largest and center-located contour is selected as a pointer boundary. Compared to the prior result shown in Figure 3(i), the new method extracts only one pointer boundary and the boundary quality is even better than before.

4. EVALUATION

4.1 Test setup and evaluation method

Two subsets of ImageCLEF2010 [10] dataset are used for evaluation. Set_1 contains chest CT images and it was used for the evaluation in [6]. Set_2 contains various image modalities (e.g., CT, MRI, Ultrasound, etc.) and images were randomly selected from the entire dataset. All images in the two sets contain one or multiple pointers. Ground truth of the pointers was created and it includes various items of information such as pointer type, color, location, and pointing direction.

Two pointer segmentation methods are considered for performance comparison. One is our edge-based method proposed in [4] and the other is Cheng’s method discussed in [7]. We first apply each segmentation method to the datasets and then feed the output contours to our pointer recognition algorithm. All methods perform several rule-based filtering methods to eliminate small noisy candidates (e.g., minimum candidate area, contour length, bounding box size, etc.). This evaluation scheme could be fairly reasonable since segmentation result itself is not a final output but an input to the recognition algorithm. A segmentation result that produces a better recognition result than others can be assumed the better one.

Another evaluation is performed to evaluate the effect of pointer segmentation algorithm on ROI annotation. Two recognition results obtained from Set_1, viz., edge-based and the proposed segmentation methods, are provided to the textual and image ROI pairing algorithm discussed in [6] and the accuracy is compared. The algorithm automatically pairs recognized pointers from images with textual pointers (e.g., “white straight arrows”, “arrowheads”, etc.) extracted from image’s figure captions. By such a pairing process, extracted image ROIs can be automatically annotated with accurate text descriptions.

Table 1. Evaluation results of pointer segmentation methods

Set	Pointer segmentation method	Number of pointers			Precision (%)	Recall (%)	F ₁ Score
		Ground truth	Detected	Detected True			
Set_1	Cheng’s method [7]	1,049	958	777	81.1	74.1	0.77
	Edge-based method [4]		3,583	816	22.8	77.8	0.35
	Proposed method		1,017	856	84.2	81.6	0.83
Set_2	Cheng’s method	3,162	2,955	2,267	76.7	71.7	0.74
	Edge-based method		10,504	2,758	26.3	87.2	0.40
	Proposed method		2,997	2,593	86.5	82.0	0.84

Table 2. ROI annotation results

Segmentation method	Total detected	Successfully annotated	Precision (%)	Recall (%)	F ₁ score
Edge-based segmentation	854	686	80.3	65.4	0.72
Proposed segmentation	856	825	96.4	78.6	0.87

4.2 Evaluation results and discussion

Precision, recall, and F₁ score are computed for the performance metric. Total number of images and pointers in Set_1 and Set_2 are 298/1,049 and 1,423/3,162, respectively. Table 1 shows the evaluation results.

The proposed method achieved the highest F_1 and precision on the two sets and the highest recall on the Set_1, but the second highest recall on the Set_2. The improvement in precision compared to our prior edge-based method can be explained as follows. Edge-based method generates more candidates from edge image by adaptive thresholding, which enables weak edges, for example edges with intensity of 128, to be detected as candidates, and the increased candidates result in the low precision. One benefit of the adaptive thresholding could be the higher recall rate. Global thresholding-based methods, both our proposed and Cheng’s methods, on the other hand, consider only highly probable candidates and hence can achieve higher precision but slightly lower recall. Our additional experiment using much lower threshold (128) for pointer boundary detection on the Set_1 (the result is not shown in Table 1) showed about 3% improved recall but about 15% lower precision than the result in Table 1.

Compared to Cheng’s method, our proposed method achieved better precision and recall on both test sets. A drawback of Cheng’s method may be found in the global thresholding. Pointers with smaller intensity than the global threshold are eliminated even though they have high contrast edges for edge detection. Our method can detect such pointers by using edge features, producing better precision and recall.

Table 2 shows that our proposed segmentation method improves ROI annotation performance as well. Both segmentation methods produced similar numbers of detected pointers after each recognition result was combined with textual ROIs. However, recognition result using our new segmentation method outperformed the result using old segmentation by about 16% and 13% in precision and recall, respectively.

One weakness of our new method is that it may not be able to completely solve the weak edge problem. Figure 5 shows an example that was not segmented by the new method. The edge-based method successfully segmented the pointer since as shown in Figure 5(b) entire pointer boundary has clearly brighter intensity than other region, and adaptive thresholding could separate the pointer boundary from the background as shown in Figure 5(c). Our new method, however, failed since the entire boundary has edge intensity smaller than 220. The result in Figure 5(d) shows binarization result by threshold 160, which is much smaller than 220, but the segmentation quality is still poor.

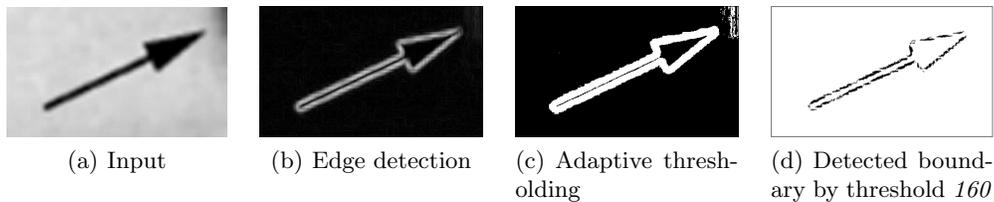


Figure 5. An error case due to overall weak edge

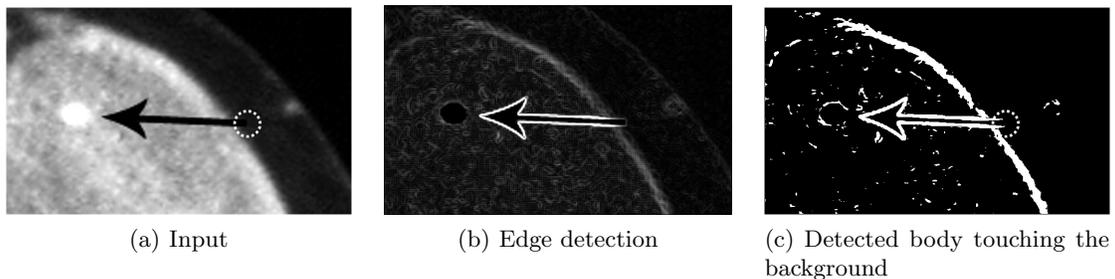


Figure 6. A pointer with similar body and background color

Figure 6 shows another example, which is a more challenging case. In the dotted circle in Figure 6(a), pointer body and background have very similar intensity, and it is hard to separate them by conventional thresholding-based methods. Unlike the arrow shown in Figure 1(b) that has similar partial weak edge but has perfectly

separable body region, our algorithm could not separate the pointer body from the background (see the touching pointer body and background in the dotted circle shown in Figure 6(c)) and hence failed in recognition.

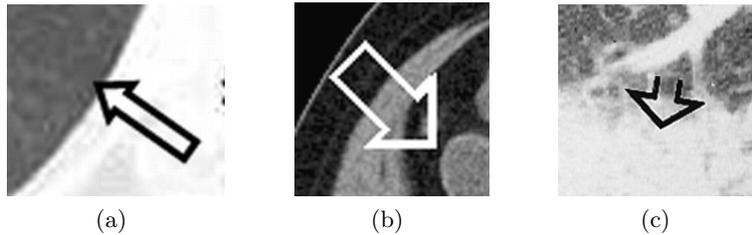


Figure 7. Samples of open arrows

Figure 7 shows samples of *open arrows* that are not always successfully detected due to their clear (transparent) body. Most open arrows have high contrast pointer boundary, which is one necessary condition for successful segmentation, and hence segmentation result depends on the pattern of the background underneath. For example, the pointer in Figure 7(a) can be successfully segmented since the background is similar to a pointer body of a bright solid pattern. However, the open arrow in Figure 7(b) has background of several texture regions, none of which is sufficient to be detected as pointer body. Figure 7(c) is a case that can rarely be segmented by our method. The pointer body is touching the background and hence the body detection algorithm fails in most such cases.

5. CONCLUSION

A new method for robust pointer segmentation in biomedical images is presented in this article. Our research effort was focused on enhancing the performance of our previous segmentation methods. Our prior edge-based approach achieved around 80% recall but much lower precision. Improving the precision rate is important to achieve better retrieval results in our system that utilizes local image regions of interest (ROI) to search for relevant images.

We combined fundamental ideas in our prior edge- and region-based segmentation methods to exploit each method's strength and overcome its weakness. Positive pointer boundary candidates are detected by a similar but slightly modified edge-based method. Adaptive thresholding in the edge-based method is replaced with a global thresholding. This may generate more broken boundaries; however, it reduces noisy boundaries effectively, which is the first step toward achieving better precision rate. Pointer body region is detected by another global thresholding applied to the edge detection result. This approach is similar to our region-based method that expands a seed region to segment homogeneous intensity blobs (potential pointer regions) but much faster than the region growing approach. The two results (detected pointer boundaries and body regions) are then combined to filter out noisy candidates and localize potential pointer regions. Local pointer candidate regions are then binarized by a threshold computed in the combining step, and the final pointer contours are extracted from the binarized local regions.

The proposed method solved several challenging problems that our previous methods could not handle. Broken boundary problem is one of them and solved by thresholding the local pointer region by accurate pointer body color. Low precision could be solved by applying strict conditions that reflect common pointer properties in the candidate detection steps. An additional benefit of the new method is that it provides a single pointer contour. As shown in Figure 1, the edge-based method extracts two contours. In ideal cases both contours are useful for recognition. However, besides the two problematic cases shown in Figure 1, in case recognition results of each boundary are different for any reasons, it is difficult to select one as the final result. We also observed that the new method provides cleaner and smoother contours than the edge-based approach, which can help achieve better recognition accuracy. Preliminary result on ROI annotation test shows that the new segmentation method leads to better ROI annotation performance as well.

Our new segmentation method still needs to be improved for better performance. Cases where pointers have weaker edges than a threshold are on top of the list. Pointers that have a color similar to the background offer

significant challenges. Potential solutions could be found in methods for occluded object detection or approaches handling image pixels in a more granular way than conventional thresholding-based approaches.

ACKNOWLEDGMENTS

This research was supported by the Intramural Research Program of the Lister Hill National Center for Biomedical Communications, an R&D division of the National Library of Medicine, at the National Institutes of Health, U.S. Department of Health and Human Services.

References

- [1] Demner-Fushman, D., Antani, S., Simpson, M., and Rahman, M., “Combining text and visual features for biomedical information retrieval,” tech. rep., Lister Hill National Center for Biomedical Communications, National Library of Medicine (NLM) (2010).
- [2] Deserno, T. M., Antani, S., and Long, R., “Ontology of gaps in content-based image retrieval,” *Journal of Digital Imaging* **22**, 202–215 (April 2009).
- [3] You, D., Antani, S., Demner-Fushman, D., Rahman, M. M., Govindaraju, V., and Thoma, G. R., “Biomedical article retrieval using multimodal features and image annotations in region-based CBIR,” *Document Recognition and Retrieval XVII* **7534**(1), 75340V+, SPIE, San Jose, California, USA (2010).
- [4] You, D., Apostolova, E., Antani, S., Demner-Fushman, D., and Thoma, G. R., “Figure content analysis for improved biomedical article retrieval,” *Document Recognition and Retrieval XVI* **7247**, 72470V–10, SPIE, San Jose, CA, USA (2009).
- [5] You, D., Antani, S., Demner-Fushman, D., Rahman, M. M., Govindaraju, V., and Thoma, G. R., “Automatic identification of ROI in figure images toward improving hybrid (text and image) biomedical document retrieval,” *Document Recognition and Retrieval XVIII* **7874**(1), 78740K+, SPIE, San Francisco, California, USA (2011).
- [6] Simpson, M. S., You, D., Rahman, M. M., Antani, S. K., Thoma, G. R., and Demner-Fushman, D., “Towards the creation of a visual ontology of biomedical imaging entities,” *AMIA Annual Symposium Proceedings (accepted)* (2012).
- [7] Cheng, B., Stanley, R. J., De, S., Antani, S., and Thoma, G. R., “Automatic detection of arrow annotation overlays in biomedical images,” *International Journal of Healthcare Information Systems and Informatics (IJHISI)* **6**(4), 23–41 (2011).
- [8] Sonka, M., Hlavac, V., and Boyle, R., [*Image Processing, Analysis, and Machine Vision*], Thomson-Engineering (2007).
- [9] <http://opencv.willowgarage.com/wiki/>.
- [10] Müller, H., Kalpathy-Cramer, J., Eggel, I., Bedrick, S., Reisetter, J., Kahn, C. E., and Hersh, W. R., “Overview of the CLEF 2010 medical image retrieval track,” *Working Notes of CLEF 2010* (2010).