Overlaid Arrow Detection for Labeling Regions of Interest in Biomedical Images

K.C. Santosh, *University of South Dakota*

Laurent Wendling, *Université Paris Descartes*

Sameer Antani and George R. Thoma, *National Library of Medicine*

A new template-free, geometric signature-based technique detects arrow annotations on biomedical images. Segmented image regions (fuzzy binarization) are checked for geometric properties and validated with theoretical ones.

Applications that use biomedical images are valuable not only in medical research and education but also in clinical decision support systems. In applying content-based image retrieval (CBIR) technology, the term *content* refers to encoding meaningful regions of interest (ROIs) with visual feature representations of colors, shapes, and textures. A key step, therefore, is to automatically identify such ROIs and annotate them according to concepts from biomedical text.[1] Because medical images tend to be complex, researchers often use pointers (that is, arrows and symbols) to highlight meaningful ROIs (see Figure 1) while minimizing distractions from other, less relevant regions. Additionally, ROIs are often referred to in figure captions and mentioned in the text of biomedical articles. Detecting arrows—a core theme of this article—could help identify meaningful ROIs and improve CBIR performance (see the "Related Work in Arrow Detection" sidebar).[1,2]

Our Proposed Method

As Figure 2 shows, our concept can be summarized as follows, which is a thorough extension of previous work.[3] It relies on a fuzzy binarization process to extract candidate regions. For this, we employ an adaptive tool to produce four different levels of binarized images, to solve the problem that regular binarization techniques miss with overlaid arrows. We take connected components (CCs) from every level of binarization and compute key points representing arrowheads by using the geometrical convex properties of an arrow. For each key point, we compute two major criteria, symmetry and overlap, to select potential candidates. This step helps prune noisy CCs. We then select triplet points that characterize the arrowhead. To make a decision, we then estimate the sets of triplets from the comparison of theoretical and discrete signature models (which we discuss more later in this

1541-1672/16/$33.00 © 2016 IEEE
Published by the IEEE Computer Society

Related Work in Arrow Detection

In the content-based image retrieval literature, there are few techniques to detect overlaid arrows. Beibei Cheng and colleagues separate arrow-like objects from text-like ones, assuming that arrows are shown in either black or white (depending on the background color).[1] From the binary image, arrow-like object separation employs a fixed-sized mask, which is then used to compute features such as major and minor axis lengths, axis ratio, area, solidity, and Euler number.

Another study uses a pointer region and boundary detection to handle distorted arrows,[2] which is followed by edge-detection techniques and fixed thresholding. Fundamentally, edge-based arrow-detection techniques are limited by the weak-edge problem[1,2] due to manual thresholding (either global or local). A hard threshold cue often weakens the decision in pointer detection. For edge detection in binary or grayscale images, most methods use classical algorithms like Roberts, Sobel, and Canny edge detection.[3]

Template-based methods are limited because they require new templates to train new images. Also, it could be necessary to re-evaluate the threshold values when new images are used. Edge-based techniques are still considered because sampling points can be much more compact than solid regions. However, broken boundaries (due to nonhomogeneous intensity distribution, where pointers overlap with content) are sometimes not fully recovered. In this context, techniques that use key points from solid objects hold promise and form the basis of our proposed method. Following the basic concept presented by Laurent Wendling and Salvatore Tabbone,[4] in this article, we use the geometrical definition of an arrow. To detect arrows, we compare discrete signatures that are computed from key points with theoretical signatures.

References

1. B. Cheng et al., "Automatic Detection of Arrow Annotation Overlays in Biomedical Images," *Int'l J. Healthcare Information Systems and Informatics,* vol. 6, no. 4, 2011, pp. 23–41.
2. D. You et al., "A Robust Pointer Segmentation in Biomedical Images toward Building a Visual Ontology for Biomedical Article Retrieval," *Document Recognition and Retrieval*, SPIE 8658, US Nat'l Library of Medicine, 2013; doi:10.1117/12.2005934.
3. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, 2006.
4. L. Wendling and S. Tabbone, "A New Way to Detect Arrows in Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, 2004, pp. 935–941.

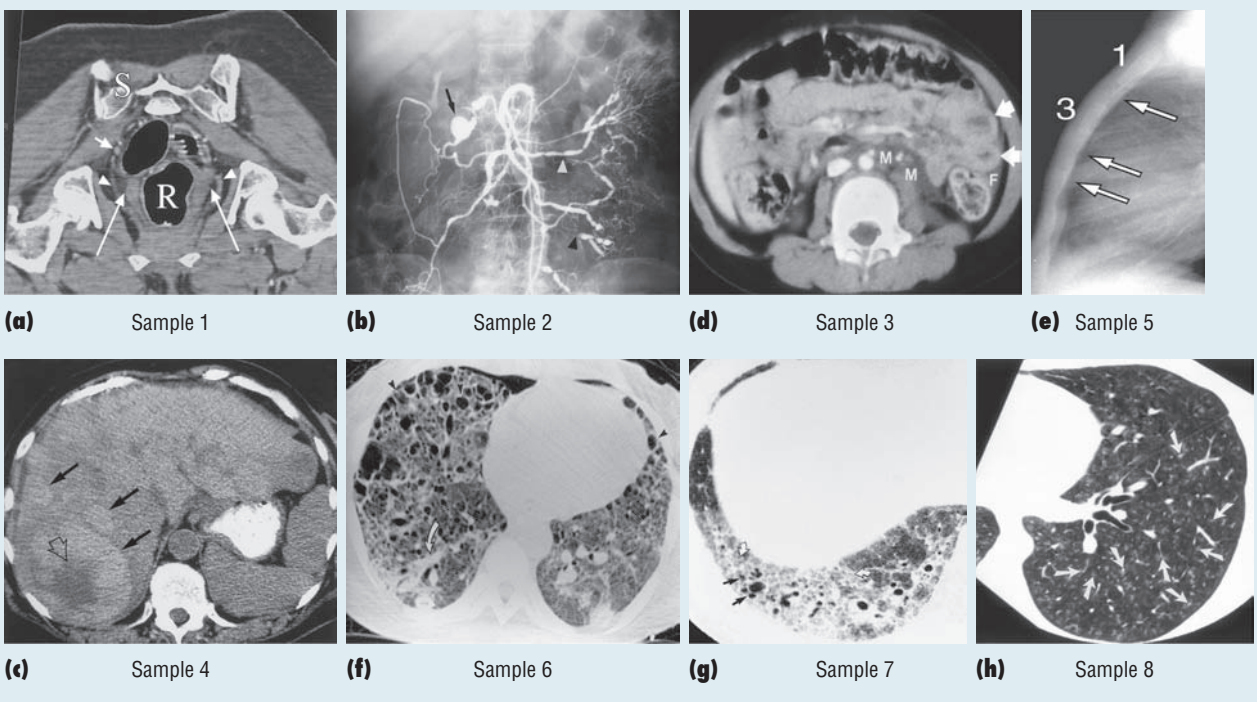

Figure 1. Examples of arrows in biomedical images: (a)–(h) sample 1 through 8. Biomedical publications use a variety of arrow styles to make arrows visible, including filled and nonfilled (black and white) arrows against different background colors.

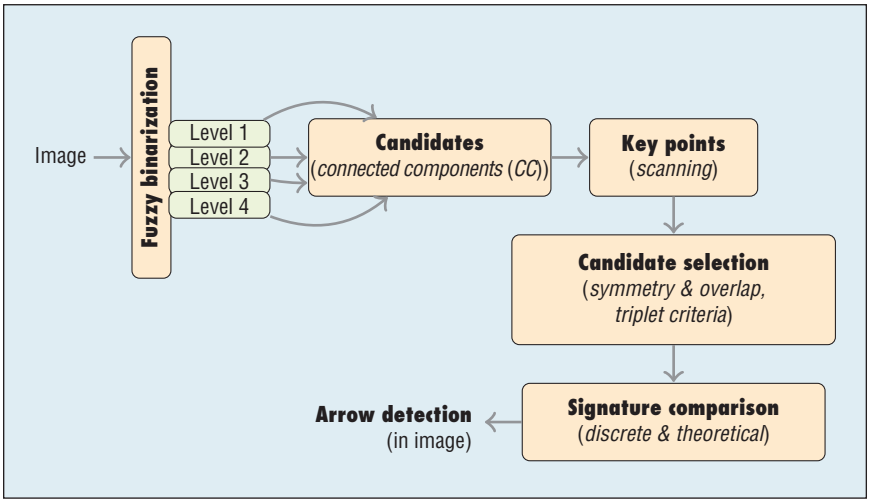

Figure 2. The overall workflow of our system in block format. It first illustrates candidates (that is, connected components) extracted by using fuzzy binarization at four different levels. These candidates are then processed to determine whether they're arrows via a series of steps, including key points detection and signature comparison.

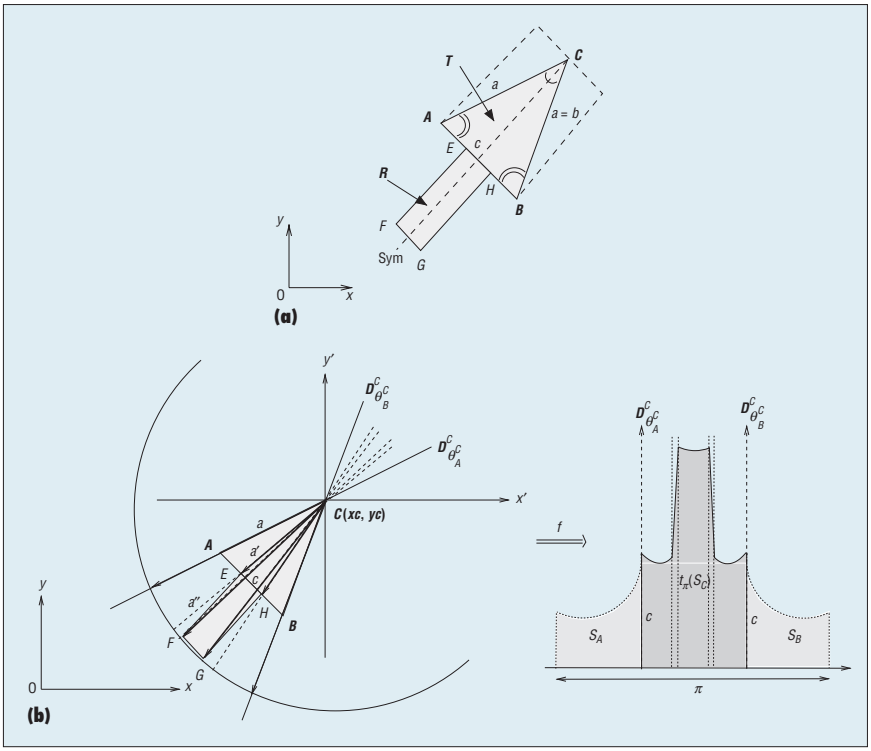

Figure 3. An example illustrating (a) a discrete arrow and (b) a signature from point C (that is, $S_C(\vartheta)$ of an arrow using pencil of lines in that particular sector). Note that $t\pi(S_c)$ is the translated signature by π; the angles are the opposite (a mirrored image) of the image frame.

article). In our assessment, the corresponding candidates are said to be detected as arrows if their similarity ratio crosses the empirically set threshold.

Geometric Properties of the Arrow

As Figure 3 shows, an arrow is modeled as an isosceles triangle **T** linked to a rectangle **R**:

$$\mathbf{T} = \left(A_{(x_a,y_a)}, B_{(x_b,y_b)}, C_{(x_c,y_c)}\right) \text{ and}$$

$$\mathbf{R} = \left(E_{(x_e,y_e)} F_{(x_b,y_b)} G_{(x_g,y_g)} H_{(x_f,y_f)}\right),$$

where we set $c = d(A, B)$ and $a = d(A, C) = d(B, C)$, and d refers to a Euclidean distance metric. The aim is to model both discrete and theoretical arrow signatures[4] and to integrate them into a broad (scalable) arrow-recognition model.

Discrete signature. Let $\text{sec}(a, c)$ be the sector defined from the segments $[A, C]$ and $[A, B]$, which includes T from A. We can then set the angle between (A, B) and (O, x) as

$$\theta_B^A = a\tan\left(\frac{y_B - y_A}{x_B - x_A}\right) + m\pi,$$

where $x_B - x_A \neq 0$ and $m \in N$, as well as the angle θ_C^A between (A, C) and (O, x).

A triangle is, by definition, a convex polygon. For any segment joining two points in **T**, every point on the segment must also lie within **T**. From A, take the pencil of lines $\mathcal{L}$ and the corresponding set V defined by **T** from A, in the sector $\text{sec}(a, c)$:

$$\mathcal{L}^A = \left\{D_\theta^A\right\}_{\theta \in [0,\pi]} \text{ and}$$

$$V_A = \left\{I_A^\theta\right\}_{\theta \in \left[\theta_B^A, \theta_C^A\right]},$$

where D_θ^A is the set of lines at angle θ, and I_A^θ is the segment made by the pencil of lines contained in that sector. Note that C_A on the circle centered in A of radius $r = \max(a, c)$ and all the segments I_A^θ are belonging to C_A. We repeat the same process to define V_B. In addition, V_C includes the definition of **R** (using C_C centered in C of radius $r = \max(a, d(C, F))$). In this case, **T** and **R** are completely processed (scanned), and $CEFGH$ is convex. For efficiency reasons, we used Bresenham's algorithm[5] to quickly minimize errors in drawing lines on integer grid points.

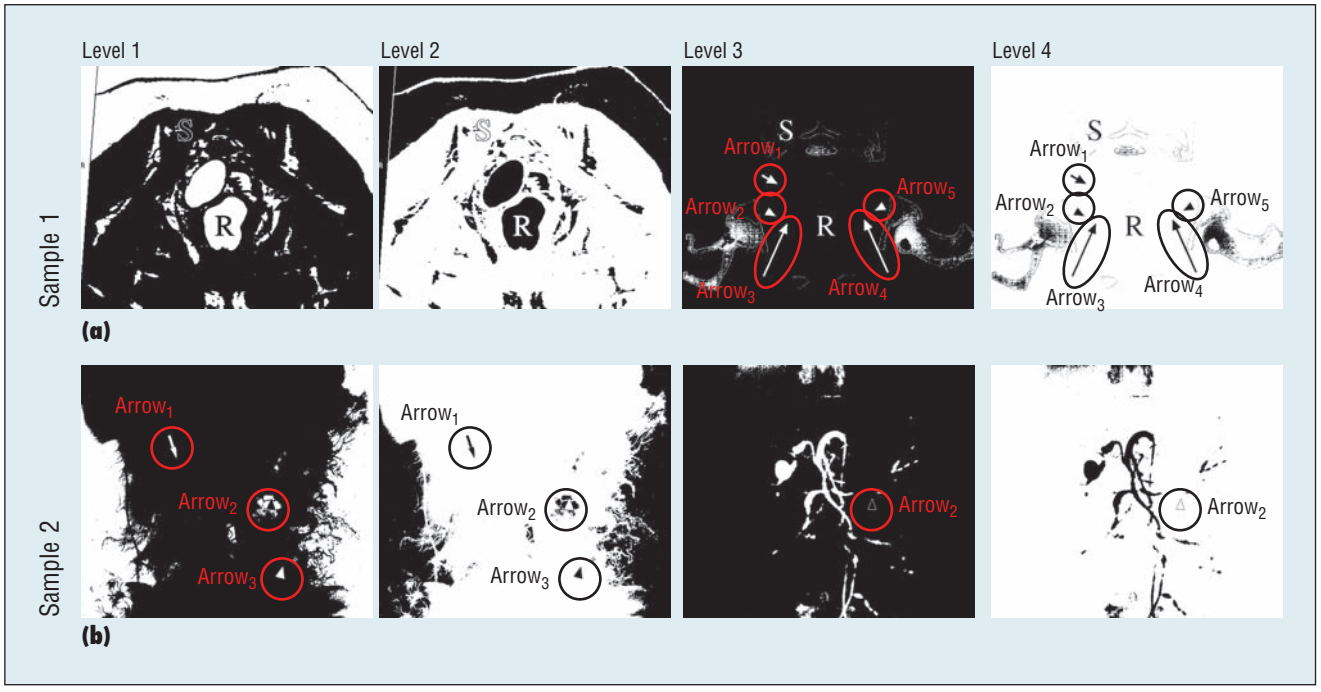

Figure 4. Fuzzy binarization at four different levels using two samples from Figure 1: (a) sample 1 and (b) sample 2. The segmented arrows are circled in red or black, depending on the background color.

Theoretical signature. We define the theoretical signature, S_X, corresponds to the discrete ones, V_X. In other words, we consider any triangle, $\mathbf{T}'$, that consists of three unaligned points X_1, X_2, and X_3. We assume X_1 to be the origin of the orthogonal frame and θ' and θ'' to be the angles described by the segments $[X_1, X_2]$ and $[X_1, X_3]$ in the frame. We also set $x = d(X_1, X_2)$ and $y = d(X_1, X_3)$. Let f be the function that defines the new representation of $[X_2, X_3]$ from X_1:

$$S_{x_1}(\theta):[\theta'+\theta'']\to\mathbb{R}_*^+ :$$
$$f\left(\theta\left\langle\left(x,y,\theta',\theta''\right)\right\rangle\right)$$
$$=\frac{x.y.\sin(\theta'+\theta'')}{x.\sin(\theta+\theta')-y.\sin(\theta-\theta'')}. \quad (1)$$

Based on Equation 1, for the points A, B, and C associated with the triangle $\mathbf{T}$, we have

$$S_A(\theta):\left(\theta_B^A,\theta_C^A\right)\to \mathrm{R}_-^+, S_A(\theta)=f\left(\theta,\left\langle a,c,\theta_B^A,\theta_C^A\right\rangle\right),$$
$$S_B(\theta):\left(\theta_C^B,\theta_A^B\right)\to \mathrm{R}_-^+, S_B(\theta)=f\left(\theta,\left\langle b,c,\theta_C^B,\theta_A^B\right\rangle\right),$$
$$S_C(\theta):\left(\theta_A^C,\theta_B^C\right)\to \mathrm{R}_-^+, S_C(\theta)=f\left(\theta,\left\langle a,a,\theta_A^C,\theta_B^C\right\rangle\right),$$

where $\theta_C^A \le \theta \le \theta_B^C$ for $\mathbf{R} = \varnothing$ in the case of $S_C(\theta)$. This doesn't hold true when $\mathbf{R} \neq \varnothing$, and therefore, five different triangles are processed (see Figure 3):

$$S_C(\theta)\to\begin{cases}\theta_A^C\le\theta<\theta_E^C & f\left(\theta,\left\langle a,a',\theta_A^C,\theta_E^C\right\rangle\right)\\ \theta_E^C<\theta\le\theta_F^C & f\left(\theta\left\langle a',a'',\theta_E^C,\theta_F^C\right\rangle\right)\\ \theta_F^C<\theta\le\theta_G^C & f\left(\theta,\left\langle a'',a'',\theta_F^C,\theta_G^C\right\rangle\right)\\ \theta_G^C<\theta\le\theta_H^C & f\left(\theta,\left\langle a'',a',\theta_G^C,\theta_H^C\right\rangle\right)\\ \theta_H^C<\theta\le\theta_B^C & f\left(\theta,\left\langle a',a,\theta_H^C,\theta_B^C\right\rangle\right),\end{cases}$$

where $a' = d(C, E) = d(C, H)$ and $a'' = d(C, F) = d(C, G)$. Overall, two different cases defined in the context of $\mathbf{R}$ must be used to compute $S_C(\theta)$. More details about the arrow signature are available elsewhere.[4]

Fuzzy Binarization

Overlaid arrows appear at either high or low intensity to enhance their visibility in biomedical images. In many cases, arrows are blurred, overlapped, or surrounded by textured areas. In such contexts, typical binarization tools based on manual threshold values are unable to extract candidates of interest. Therefore, we use an adaptive binarization tool, which is based on a fuzzy partition of a 2D histogram of the image by taking gray-level intensities and local variations into account.[6] We then compute 2D Z-function criteria from the histogram to automatically set the threshold. This is based on the optimization of fuzzy entropy. The Z-function employs two kernels: low- and high-level cuts. But in our case, we also use their inversions as illustrated in Figure 4.

Image inversion takes opposite image intensities into account: arrow candidates are encircled in both red and black (according to the background color). The main idea of using four different levels of binarization is not to miss the overlaid arrows. Furthermore, deformed arrows can be avoided because the arrows are repeated in other levels of binarization. For example, in Figure 4b, arrow 2 is fully surrounded as well as overlapped by noisy textures at the first two levels but is visible at the other two levels.

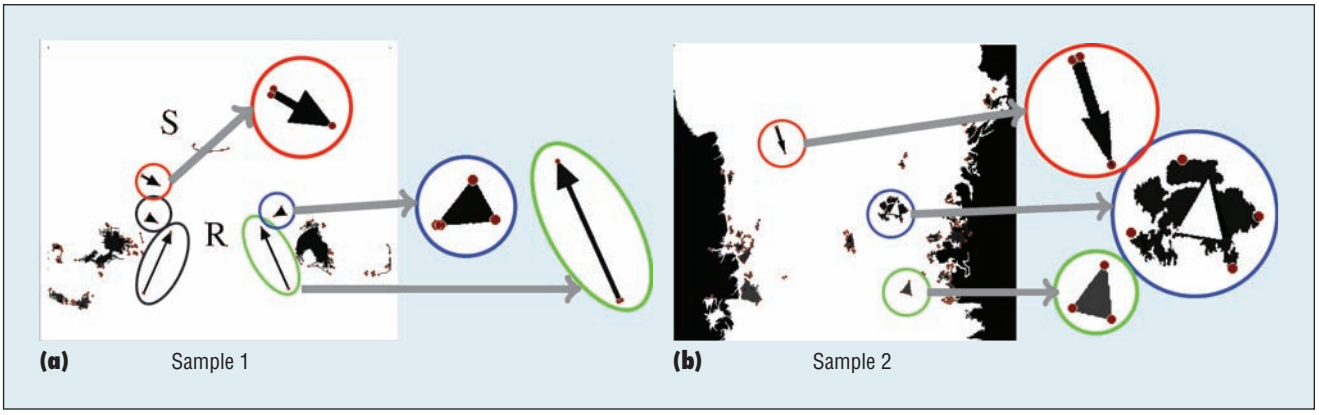

Figure 5. Two examples illustrating the candidates and their corresponding key points that result from orthogonal scanning (from Figure 4): (a) level 4 of sample 1 and (b) level 2 of sample 2. Several regions of interest are magnified to make key points visible.

While checking with the ground truth, we can segment 95.1 percent (recall value) of arrows from our complete dataset. To detect the arrow-like candidates from the complete set, we perform the major tasks of key points selection (representing arrowhead points) and arrow assessment based on symmetry and overlap computation, and signature comparison with the theoretical one.

Key Points Selection

Our aim is to collect key points (see Figure 3) by studying four scans over the orthogonal frame. It includes two corners (top-left and right-bottom) of the image. From each corner, we perform column- and row-wise scanning.

Consider a set of candidates $\{\mathbf{G}_k\}_{k=1,\ldots,K}$. For any kth candidate, we perform scanning for all pixels I_{ij} of the studied image I. For a scan s, points are added to the list L_s^k of k candidate, as follows:

$$L_s^k \leftarrow L_s^k \cup \{(i,j)\} \text{ iff } \mathbf{I}_{ij} = k \wedge \forall s, (i,j) \notin L_s^k \cdot$$

In other words, L_s^k is the list of points associated with kth candidate (see Figure 5).

Candidate Selection: Arrow Assessment

For any candidate k, as described previously, the discrete signature relies on the location of the arrowhead key points A, B, and C. During the recognition process, we compute geometric properties of an arrow considering all combinations of points.

Symmetry and Overlap

From a list of key points L_s^k for any kth candidate, we select point C by using the straightforward convex property of the arrowhead (see Figure 3a). If several points are selected, we confirm C by estimating the symmetric axis of the arrow, sym(C).

Consider the set of n points X_i included in the list. We compute discrete distribution V_{X_i} for p number of bins (each bin corresponds to a specific direction in the specified frame):

$$Sym(X_i) = \sup_{t \in \left[1,\ldots,\frac{p}{2}\right]} \left\{ \frac{\Sigma_{i=1}^{p} \min\left(V_{X_i}^{t+i}, V_{X_i}^{t-1}\right)}{\Sigma_{i=1}^{p} \max\left(V_{X_i}^{V_{t+1}}, V_{X_i}^{t-1}\right)} \right\}.$$

We then select a unique point, $Sym(C) = \arg\max_{i=1,\ldots,n} \{Sym(X_i)\}$. Note that sym is maximum when angle t corresponds to the signature's axis of symmetry. The symmetry axis $\Delta(C)^t$ at angle t is computed to split the remaining points. From point C, our aim is to find two other points, A and B. Because the base of the triangle largely covers the rectangle, we can consider the farthest points A and B as belonging to the list, such that $\Delta(C)^t \perp (A, B)$ with

$$Cardinality = \frac{\min(|V_A|, |V_B|)}{\max(|V_A|, |V_B|)}, \left\{V_A^i, V_B^i, t_\pi(V_C^i)\right\} \simeq 1,$$

where V_X refers to the area of the signature, and $t_\pi(V_X)$ (see Figure 3b) is the translated signature by π because angles are opposite to the image frame.

Additionally, we consider area assessment (an overlapping criterion),

$$Area = 1 - \left(\frac{\mathbf{K} - \mathbf{H}}{\mathbf{H}}\right),$$
$$\mathbf{H} = \sqrt{(l(l-a)(l-b)(l-c))},$$

where $\mathbf{K}$ is the common area identified from the scans performed from A, B, and C, and $l = 1/2(a+b+c)$ is the semiperimeter to check the well-defined isosceles triangle. By definition, if the triplet describes a triangle in the image, the value of $\mathbf{K}$ must be close to the calculation of the Heron's formula, $\mathbf{H}$.

Signature Comparison

In our test, we calculate a basic similarity ratio (SR) based on the Tanimoto index (the minimum over maximum) between the global theoretical and discrete signatures. To compute a global signature, we

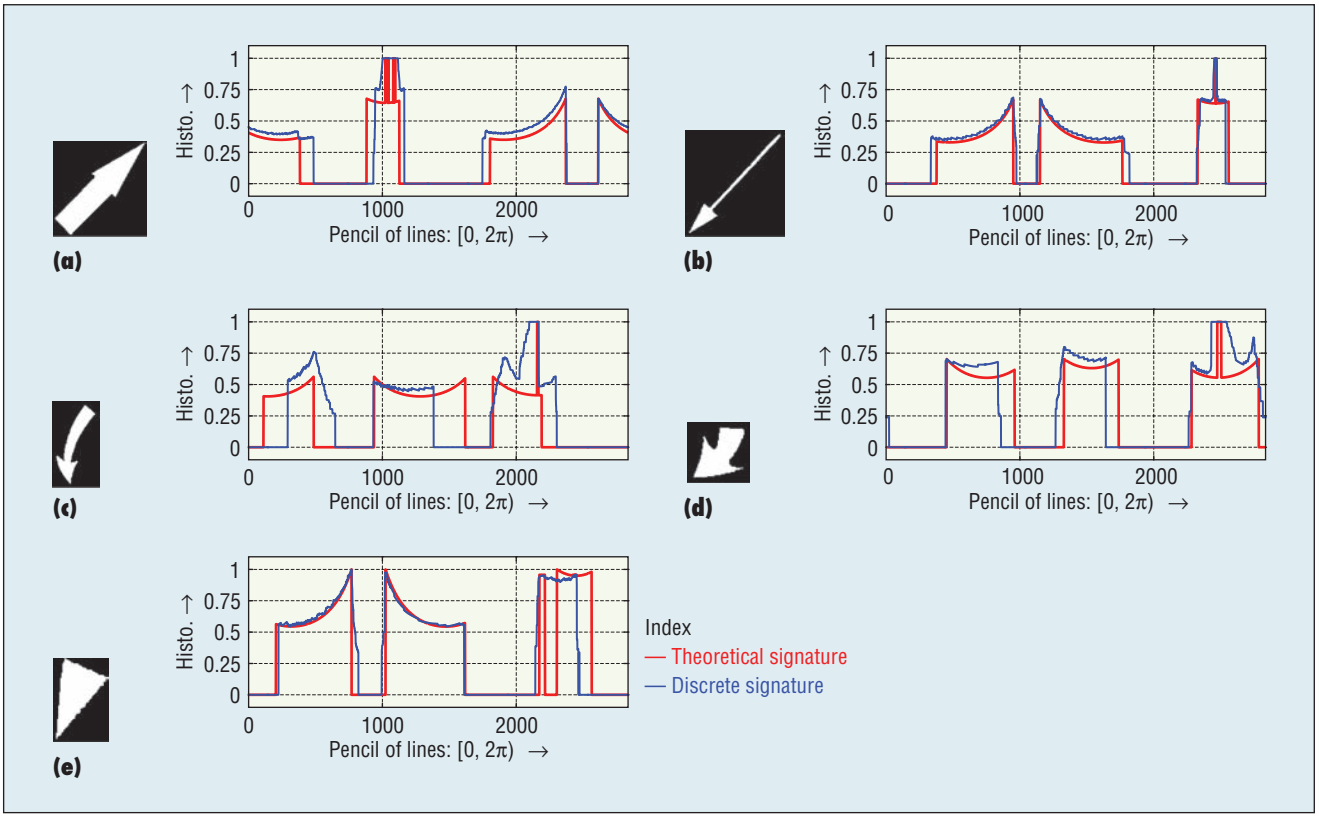

Figure 6. Discrete and theoretical, blue and red lines, respectively, where (a)–(e) signatures are compared after separate concatenation of their sets V_A, V_B, V_C and S_A, S_B, S_C. This shows the corresponding magnitude differences between signatures (due to the size of the arrowhead and the rectangle): V_X versus S_X, and $X = A, B,$ or C.

superimpose V_A, V_B, and V_C because the pixel distributions are computed in the specified frame: $[0,\pi)$. In doing so, unlike in the theoretical signatures, some of the elements in the discrete signatures overlap. This occurs primarily due to noise or from other objects connected with background and image regions. Therefore, SR is weighted by an overlapping assessment in order to compute recognition rate:

$$Reco = SR \times \left(1 - \frac{\sum_{i=1}^{p} \sup\left(\begin{matrix}\min\left(V_A^i, V_B^i\right), \min\left(V_A^i, t_\pi\left(V_C^i\right)\right), \\ \min\left(V_B^i, t_\pi\left(V_C^i\right)\right)\end{matrix}\right)}{\Sigma_{i=1}^{p} \sup\left\{V_A^i, V_B^i, t_\pi\left(V_C^i\right)\right\}}\right). \quad (2)$$

The more overlap there is, the lower the recognition rate. To avoid noisy or distorted candidates, we employ a threshold λ (empirically designed) to support the decision as to whether the shape (that is, any labeled candidate from binarization) is recognized as an arrow.

For visual illustration, we demonstrate in Figure 6 how similar discrete and theoretical signatures are. These are concatenated signatures, which are different from our superimposed images, as described earlier. This shows the corresponding magnitude differences between signatures (due to the size of the arrowhead and the rectangle): V_X versus S_X, and $X = A, B,$ or C.

Candidate Redundancy Elimination

Because we have four different levels of binarization, there are four possibilities for detecting the same arrow. To eliminate this redundancy, we use two criteria: arrow location and recognition rate. We select the option that produces the best recognition rate for the specific location,

$$\arg\max_{1 \le r \le 4} Reco_r^{loc},$$

where *reco* refers to the recognition rate of the studied arrow in any fixed location, *loc*.

Experiments

We performed a series of experiments and compared our algorithm with state-of-the-art methods.

Dataset and Ground Truths

We used a well-known 2010 dataset, imageCLEFmed, which is composed of 298 chest CT images. It contains 1,049 pointers total. For all images in the dataset, we created ground truths of the pointers, and each ground truth includes information such as arrow type, color, location, and direction.

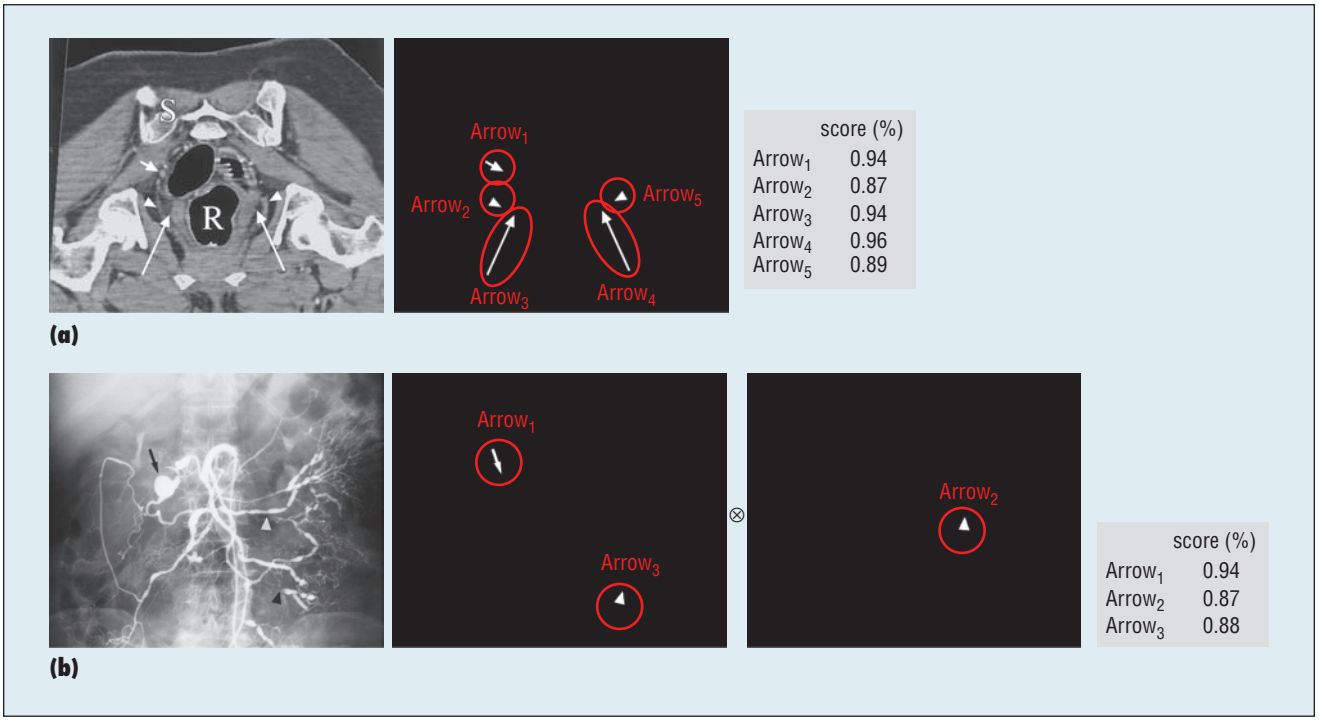

Figure 7. Examples showing (a) and (b) different binarization levels used to detect arrows. This demonstrates the idea of image inversion used in binarization because arrows are not filled with black only.

Evaluation Protocol

For validation, for any given image in the dataset, our performance evaluation criteria are precision, recall, and F_1 score. In general, these metrics can be expressed as follows:

$$\text{precison} = \frac{m1}{M},$$
$$\text{recall} = \frac{m1}{N}, \text{and} \quad (3)$$
$$F_1\ \text{score} = 2\left(\frac{m1/M \times m1/N}{m1/M + m1/N}\right),$$

where $m1$ is the number of correct matches from the detected set M, and N is the total number of pointers (in the ground truth) that are expected to be detected. Precision defines whether the retrieved candidate is relevant (that is, if it is an arrow), and recall defines how relevant the search is.

Results and Analysis

For our algorithm, we start with visual illustrations, aiming to provide an idea about the usefulness of different binarization levels and scores in arrow detection. In Figure 7, we provide arrow-detection output based on the decision defined by $\lambda = 0.85$. Taking signature matching into consideration, we also provide individual arrow-detection scores. To detect all arrows in Figure 7a, we used a single binarization level. But, in Figure 7b, we used two different binarization levels. Figure 7b shows the importance of using multiple levels in detecting arrows. If not, one arrow per image will be missed.

As we have already stated, we have white- and black-filled arrows, and therefore we performed category-wise performance evaluation. The accuracies are 88.43 percent and 84.51 percent for white- and black-filled arrows, respectively. When we use the whole dataset and evaluation protocol defined in Equation 3, precision, recall, and F_1 score are 93.14 percent, 86.92 percent, and 89.94 percent, respectively.

Comparative Study

In our comparative study, our benchmarking methods are categorized into two groups: state-of-the-art methods designed for arrow detection and a template-based method that uses well-known state-of-the-art shape descriptors.

Recent arrow-detection methods. We used two state-of-the-art methods designed for arrow detection: a global thresholding-based method (method 1)[7] and an edge-based method (method 2).[8] The results are provided in Table 1. Method 2 performed best, with precision and recall values at 84.20 percent and 81.60 percent, respectively.

Our template-based method. In the case of our template-based method, we created 11 templates (arrows) having different shapes (including different sizes). The template size can further be extended in accordance with the dataset. Figure 8 shows a few of the arrow templates. To extract shape features,

we took the most frequently used shape descriptors in computer vision. They are the generic Fourier descriptor (GFD),[9] shape context (SC),[10] Zernike moment (ZM),[11] and Radon transform (RT).[12]

For these descriptors, it's important to fit the best parameters. The GFD, for example, requires us to tune the radial (4:12) and angular (6:20) frequency parameters to get the best combinations. Note that such a best combination of radial and angular frequencies can be different from one dataset to another. For SC, we used the 100 sample points (as reported elsewhere[10]) by omitting smaller CCs. In the case of ZM, we applied 36 Zernike functions of order less than or equal to 7. For RT, the projecting range is $[0,\pi)$. These shape descriptors are rotation-, scale-and translation-invariant, and thus are useful in our application because CCs are observed at different sizes, scales, and directions. After extracting features, the aim is to rank the CCs from any studied image based on the order of shape similarity. In our test, we implemented top-10 ranking. Using this framework, results (precision, recall, and F_1 score) are provided in Table 1. Among all shape descriptors, GFD provides the best performance.

For an immediate comparison, we selected the best results from two different groups (see the boldface scores in Table 1): both method 1 and method 2 supercede GFD. Our algorithm outperforms the best of the two, method 2, by more than 9 percent in precision and 5 percent in recall.

Extension

We extended our evaluation on a different dataset (composed of 120 radiographs and available for research purposes by request) that are collected using the Open-i (the NLM's open access biomedical search engine; openi.nlm.nih.gov), which is completely different from the imageCLEFmed dataset (see Figures 10d–10f). Following the evaluation protocol (as described earlier), our results are precision = 87.35 percent, recall = 89.09 percent, and F_1 score = 88.21 percent. The results attest to the fact that the proposed method can be extended and/or generalized.

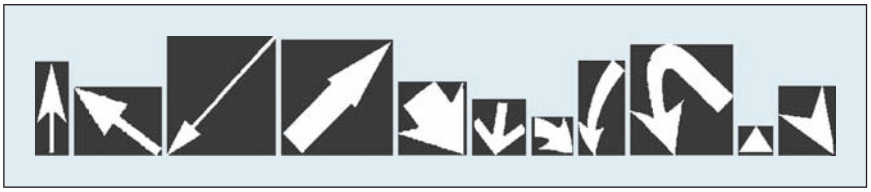

Figure 8. Arrow templates used in our template-based method.

Table 1. Performance comparison.*

Methods		Precision (%)	Recall (%)	F_1 score (%)
State-of-the-art arrow-detection methods	Method 1[7]	81.10	74.10	77.00
	Method 2[8]	**84.20**	**81.60**	**83.00**
Template-based methods	Generic Fourier descriptor[9]	**75.10**	**78.33**	**76.68**
	Shape context10	68.30	71.40	69.82
	Zernike moment[11]	55.20	57.70	56.40
	Radon transform[12]	59.50	63.60	61.48
Template-free method	Our algorithm	**93.14**	**86.92**	**89.94**

*Boldface numbers indicate the best scores for each group.

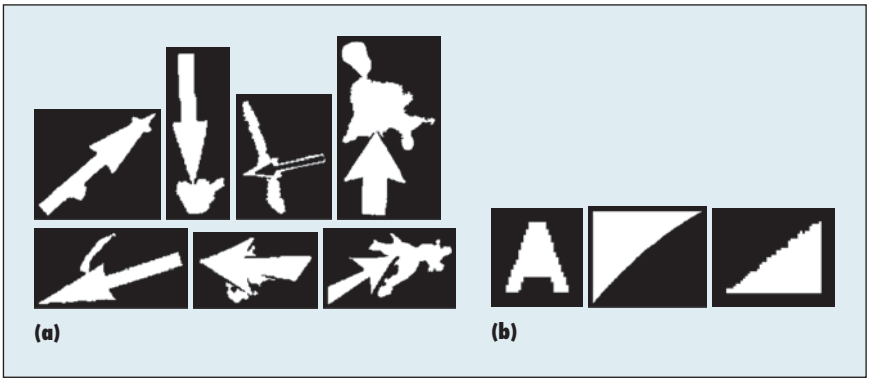

Figure 9. (a) Artifacts that resemble arrow-like objects, including noise overlapping with arrows. This helps increase false positives. (b) Triangle-shaped connected components, which are similar to arrow without tail.

Discussion

Based on the reported results from Table 1, we observed that our method provides higher precision than the benchmarking methods. This reflects the robustness of our arrow-detection algorithm and, of course, the appropriate use of the fuzzy binarization tool. Furthermore, considering recall measure of the binarization tool (95.10 percent), our arrow-detection algorithm misses only 2 percent of arrows, which isn't a significant loss. One of the primary reasons for such misses is that the binarization tool still carries some artifacts resembling arrow-like objects, including noise overlapping with the arrows (see Figure 9a). Similarly, degraded arrows affect the algorithm's performance. Furthermore, theoretically speaking, our algorithm

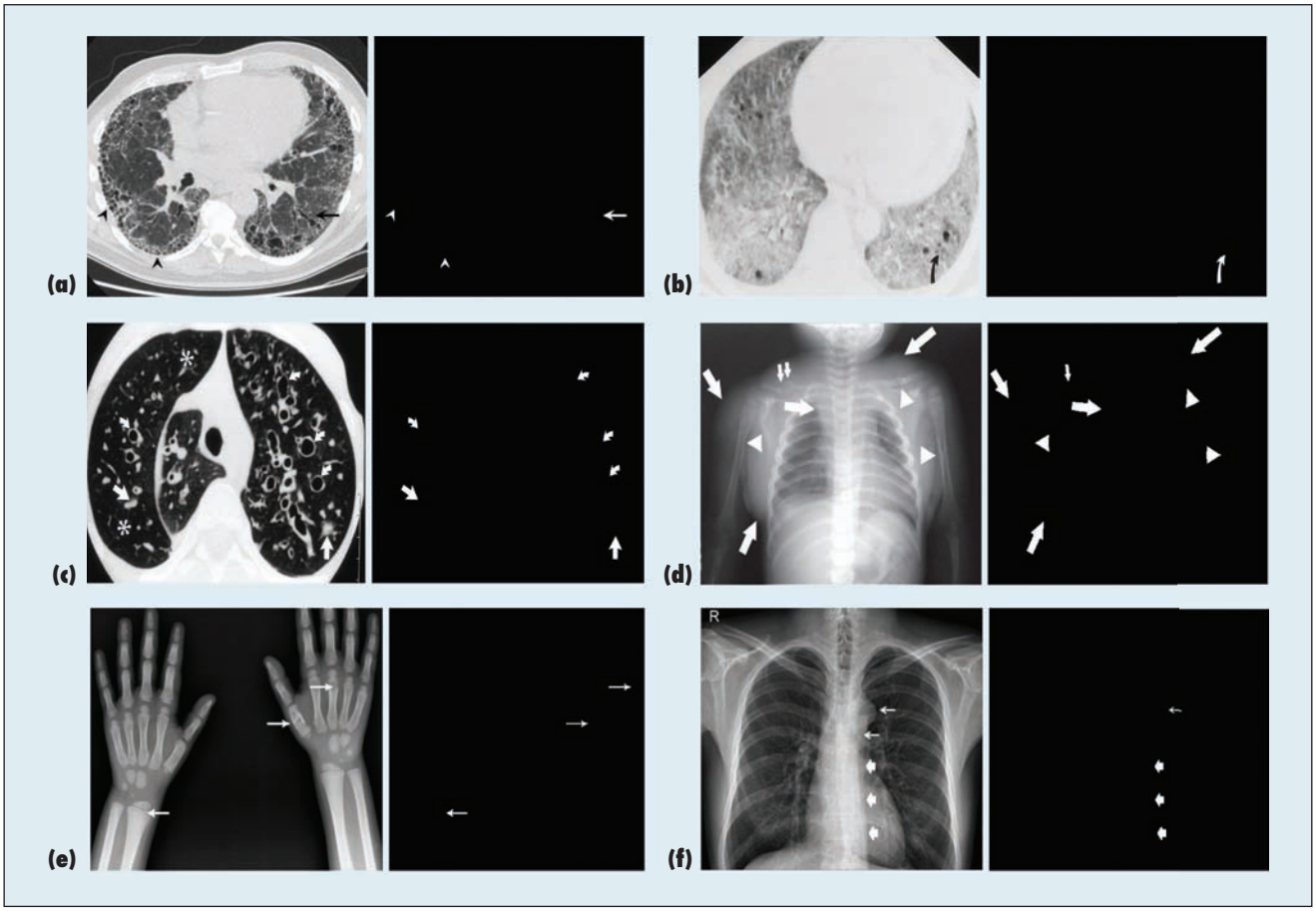

Figure 10. Examples illustrating arrow detection. (a)–(f) Detected arrows (in white with black background on the right) are the combined results of four different binarization levels.

can't reject triangle-shaped CCs (see Figure 9b), and therefore, triangular CCs segmented from the corners of the image are detected as false positives.

For better understanding, Figure 10 shows more output samples, where detected arrows are placed to the right of the original sample images. Our current work does not consider curved arrows or fixed graphical shaped pointers such as star or asterisk (see Figure 10c, as an example). In Figures 10d–10f, arrow detection on radiology images are shown. In this illustration, bigger arrows provide higher recognition rates because their signatures are more robust to noise than smaller ones. Smaller ones do not offer clear arrowheads (see Figures 10d, and 10f).

In contrast, template-based arrow-detection methods achieved high recall. Note that recall can be increased by extending the retrieval scope. But, for a system to be precise and accurate, it needs a list of all detected arrows in the first few ranks without having false positives (see Equation 3). Moreover, unlike other methods, our method can detect degraded and deformed arrows using shape descriptors because they are robust. However, such robustness affects overall performance because it might also detect several artifacts. Shape-based descriptors can handle curved arrows as well if templates are provided.

In future work, some cases of occulted arrows can be improved by considering the impact of junction points as arrowhead candidates. We also plan to extend our arrow signatures to take curved arrows into account by processing arrowhead regions regardless of whether tails are curved or straight. ■

Acknowledgments

The Intramural Research Program of the National Institutes of Health, National Library of Medicine, and Lister Hill National Center for Biomedical Communications supported this work. We thank the National Institutes of Health Fellows Editorial Board for their editorial assistance.

References

1. D. Demner-Fushman et al., "Design and Development of a Multimodal Biomedical Information Retrieval System," *J. Computing Science and Eng.*, vol. 6, no. 2, 2012, pp. 168–177.
2. M. Simpson et al., "Towards the Creation of a Visual Ontology of Biomedical Imaging Entities," *AMIA Ann. Symp. Proc.*, 2012; https://knowledge.amia.org/amia-55142-a2012a-1.636547/

THE AUTHORS

K.C. Santosh is an assistant professor in the Department of Computer Science at the University of South Dakota. His research interests include pattern recognition, image processing, computer vision, and machine learning. Santosh has a PhD in computer science from INRIA – Université de Lorraine. Contact him at santosh.kc@usd.edu.

Laurent Wendling is a full professor at the Université Paris Descartes. His research interests include spatial relations, feature selection, and image segmentation. Wendling has a PhD in computer science from the Université Paul Sabatier. Contact him at laurent.wendling@parisdescartes.fr.

Sameer Antani is a staff scientist at the National Library of Medicine. His research interests include biomedical imaging and analysis, informatics, computer vision, machine learning, computer science, and engineering technology. Antani has a PhD in computer science and engineering from the Pennsylvania State University. He's a senior member of the International Society for Optics and Photonics, a member of IEEE, and vice chair for computational medicine in the IEEE Technical Committee on Computational Life Sciences. Contact him at sameer.antani@nih.gov.

George R. Thoma is chief of the Communications Engineering Branch of the Lister Hill National Center for Biomedical Communications at the US National Library of Medicine. His research interests include document image analysis and understanding, biomedical image processing, and machine learning. Thoma has a PhD from the University of Pennsylvania in electrical engineering. He's a Fellow of the International Society for Optics and Photonics. Contact him at george.thoma@nih.gov.

8658, US Nat'l Library of Medicine, 2013; doi:10.1117/12.2005934.
9. D. Zhang and G. Lu, "Shape-Based Image Retrieval Using Generic Fourier Descriptor," *Signal Processing: Image Comm.*, vol. 17, 2002, pp. 825–848.
10. S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, 2002, pp. 509–522.
11. W.-Y. Kim and Y.-S. Kim, "A Region-Based Shape Descriptor Using Zernike Moments," *Signal Processing: Image Comm.*, vol. 16, nos. 1–2, 2000, pp. 95–102.
12. T.V. Hoang and S. Tabbone, "The Generalization of the *R*-Transform for Invariant Pattern Representation," *Pattern Recognition*, vol. 45, no. 6, 2012, pp. 2145–2163.

t-003-1.640625/f-001-1.640626/a-099-1.640933/a-100-1.640930?qr=1.
3. K.C. Santosh et al., "Scalable Arrow Detection in Biomedical Images," *Proc. Int'l Conf. Pattern Recognition*, 2014, pp. 3257–3262.
4. L. Wendling and S. Tabbone, "A New Way to Detect Arrows in Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, 2004, pp. 935–941.
5. J.E. Bresenham, "Algorithm for Computer Control of a Digital Plotter," *IBM Systems J.*, vol. 4, no. 1, 1965, pp. 25–30.
6. H. Cheng and Y.-H. Chen, "Fuzzy Partition of Two-Dimensional Histogram and Its Application to Thresholding," *Pattern Recognition*, vol. 32, 1999, pp. 825–843.
7. B. Cheng et al., "Automatic Detection of Arrow Annotation Overlays in Biomedical Images," *Int'l J. Healthcare Information Systems and Informatics*, vol. 6, no. 4, 2011, pp. 23–41.
8. D. You et al., "A Robust Pointer Segmentation in Biomedical Images toward Building a Visual Ontology for Biomedical Article Retrieval," *Document Recognition and Retrieval*, SPIE