

# Efficient Graffiti Image Retrieval

Chunlei Yang<sup>1</sup>, Pak Chung Wong<sup>2</sup>, William Ribarsky<sup>1</sup> and Jianping Fan<sup>1</sup>

<sup>1</sup>University of North Carolina at Charlotte, Charlotte, NC

<sup>1</sup>{cyang36, ribarsky, jfan}@uncc.edu

<sup>2</sup>Pacific Northwest National Lab, Richland, WA

<sup>2</sup>pak.wong@pnl.gov

## ABSTRACT

Research of graffiti character recognition and retrieval, as a branch of traditional optical character recognition (OCR), has started to gain attention in recent years. We have investigated the special challenge of the graffiti image retrieval problem and propose a series of novel techniques to overcome the challenges. The proposed bounding box framework locates the character components in the graffiti images to construct meaningful character strings and conduct image-wise and semantic-wise retrieval on the strings rather than the entire image. Using real world data provided by the law enforcement community to the Pacific Northwest National Laboratory, we show that the proposed framework outperforms the traditional image retrieval framework with better retrieval results and improved computational efficiency.

## Categories and Subject Descriptors

I.2.10 [Artificial Intelligence]: Vision and Scene Understanding - *Perceptual Reasoning*.

## General Terms

Algorithms, Measurement, Experimentation

## Keywords

Graffiti Detection, Character Extraction, Image Retrieval

## 1. INTRODUCTION

Graffiti recognition and retrieval, as an application in public safety, has drawn more and more attention of researchers in the broad field of information retrieval [5]. Graffiti may appear in the form of written words, symbols, or figures, and it has sprung up in most metropolitan areas around the world. Gang-related graffiti is typically composed of mostly characters, conveys lots of information, and often identifies a specific gang territory or threatens law enforcement. The retrieval and interpretation of such information has become

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMR '12, June 5-8, Hong Kong, China

Copyright ©2012 ACM 978-1-4503-1329-2/12/06 ...\$10.00.



Figure 1: Sample graffiti images: Three of the four images contain text, while the bottom right image contains no textual information but only the “play-boy” symbol

increasingly important to law enforcement agencies. With the prevalence of hand-held devices, digital photos of graffiti are easily acquired and enormous data collections of graffiti images are rapidly growing in size. Sifting through and understanding each image in a collection are very difficult, if not impossible, for humans to do. Thus, there is an urgent need to build a visual analytic system that can be used for automatic graffiti image recognition and retrieval from large-scale data collections.

It may seem that simply applying traditional optical character recognition (OCR) on graffiti characters would address the problem. However, because of the artistic appearance of many graffiti characters and the various types of surfaces that graffiti can be painted on, understanding graffiti characters presents many more challenges than traditional OCR can solve. As a compromise, researchers take a shortcut by not utilizing any particular treatment to localize the graffiti objects in the image. Instead, traditional object localization methods are applied, for example, to extract the so-called “interesting” objects [6] or conduct the retrieval task with local feature matching on the whole image without object localization [4]. There are two primary flaws of such treatment: 1) Graffiti objects may not be “interesting” under the view of traditional object localization and 2) Textual information within the image is missing, making semantic-level understanding of the graffiti impossible. While investigating an actual graffiti image collection as shown in Figure. 1, we observed that most of the collected images have tex-



Figure 2: Challenging graffiti images

tual information (people’s names or locations), while some have figures or symbols that are also meaningful, such as the “playboy” and “crown” symbol (in bottom left image in Figure. 1; the crown image is a well-known symbol of a gang member). These observations led us to integrate the research work of both semantic and visual understanding of the data, similar to the idea of fusing visual and textual information [2].

To best describe the research tasks of graffiti recognition and retrieval, we need to inspect the challenges and differences between graffiti recognition and traditional OCR as shown in Figure. 2. The images (a) to (f) illustrate different aspects of the challenges in character detection, recognition, and image retrieval of graffiti images. Image (a) suggests that graffiti may appear on any type of surface, including walls, wooden fences, door frames, light poles, windows, or even tree trunks. The roughness and complexity of the background may bring in a lot of noise, making the task of character detection very challenging. Image (b) illustrates that graffiti usually appears outdoors and is exposed to various lighting conditions. Shadows and sunlight may dramatically affect the ability to correctly detect characters. Graffiti “words” often appear to be nonsensical because they are formed from acronyms or specially created combinations of letters as shown in image (c). In traditional OCR, the recognition result for certain letters could be used to predict the unrecognized letters by forming potential meaningful words. In graffiti recognition, we do not have such a prediction. Given that we have the ability to detect strokes of painting, we still need to further differentiate texture strokes and non-textures, such as the “playboy” symbols as shown in image (d). As illustrated in images in (e) and (f), the font and artistic writing style of characters make the same words have a very different appearance. This marked variation would impede the technique of template matching or local feature matching.

In this paper, we focus on the research task of graffiti image retrieval. After deep investigation of the challenges of the graffiti recognition task compared to OCR, we design a series of techniques for effective character detection. Next, we conduct semantic-wise and image-wise retrieval on the detected character components rather than the entire image to avoid the influence of the background noise. The visual and semantic matching scores are combined to give the final matching result. The paper is organized as follows: we

discuss related work in section 2 and then introduce the proposed framework of graffiti image retrieval in section 3. We conduct a series of experiments and evaluate the proposed framework in section 4, and in section 5, we summarize our research and discuss plans for future work.

## 2. RELATED WORK

Graffiti image retrieval research lies in the intersection of OCR and image retrieval. The techniques from both fields may benefit the graffiti retrieval work.

Graffiti image retrieval is closely related to the handwriting recognition and retrieval work in OCR. The graffiti characters are essentially handwritten characters, although they often have an artistic appearance and are usually found in more challenging environments. OCR techniques recognize and match characters based on their shape and structure information, such as skeleton feature [11, 13], shape context [1], and order structure invariance [3]. The foundation for the effectiveness of these techniques is the correct separation of the characters from strings or words detected. The encoding of the word is not an easy task, and the methods available are often trivial and may not apply to the graffiti data. Another issue is that simply measuring the similarity between two individual characters as designed in [11] is inadequate. We intend to evaluate the similarity between two strings or words to derive semantic-level understanding. The proposed evaluation metric, longest common subsequence (LCS), is designed to overcome this flaw by considering the sequence of the characters in the string [14].

The visual difficulties introduced in section 1 and the artistic appearance of graffiti images have motivated researchers to try routes other than OCR. Jain et al. [4] have proposed a system named Graffiti-ID and treats the graffiti purely as images on the retrieval task. The Graffiti-ID system does not specifically locate the character components in the images, and thus some false-positive matches may occur. Furthermore, the potential semantic relationship between the graffiti characters is completely ignored; thus, Graffiti-ID does not distinguish itself from general image retrieval frameworks.

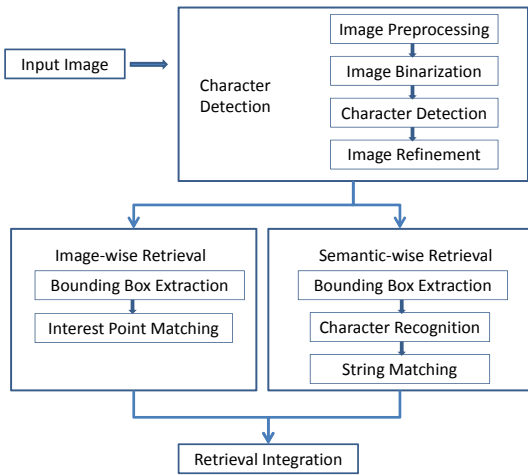
Our proposed system works on the graffiti character components that are detected in the image. Our proposed framework may have the potential to not only achieve better retrieval accuracy by eliminating as much background noise as possible, but also significantly reduce the computation burden by eliminating unrelated interest points.

## 3. GRAFFITI RETRIEVAL SYSTEM

The proposed graffiti retrieval system comprises two major components: character detection, and string recognition and retrieval. The string recognition/retrieval component is further broken down by the image-wise retrieval and semantic-wise retrieval. The work-flow of the entire system is shown in Figure. 3. In this section, we will describe the design detail of the steps, as shown in the framework diagram. We will use the top left image in Figure. 1 as an example input throughout this section.

### 3.1 Image Preprocessing

We have some basic requirement for the quality of the images. The character components should be contrasting from the background and the background is not extremely



**Figure 3: Graffiti retrieval framework**

cluttered or colorful. Otherwise, the graffiti lost its meaning to pass on messages. For preprocessing of the images, we conduct a series of sequential operations, including image resizing, grayscaling, and smoothing. The sizes of the collected graffiti images are usually large (larger than 2000 by 1500), which is difficult to display and inefficient to process. Therefore, we keep the aspect ratio and resize the image to make sure its largest dimension is smaller than 800 pixels. An image of this size shows clear graffiti characters and is small enough for efficient processing. The resized image is then changed to gray-scale<sup>1</sup> and smoothed with a 5 by 5 gaussian filter. The smoothing operation dramatically reduces large amounts of unnecessary background noise.

### 3.2 Image Binarization

The grayscale image has pixel values ranging from 0 to 255. For the purpose of character detection, we need to partition the image into the potential object area (the character area) and the background area, which is a binarization process. Image binarization is realized by the global thresholding algorithms such as Niblack [15]. The intensity of the pixel of the input image is compared with a threshold of  $T$ ; the value above the threshold is set to white (1; the potential object area pixel), otherwise black (0; the obvious background pixel). The Niblack’s algorithm calculates a pixelwise threshold by sliding a square window on the grayscale image. The size of the window is determined by the size of the image, which is based on the fact that the character components are visible and thus occupy a certain proportion of the image area. The threshold  $T$  is calculated with the mean  $m$  and standard deviation  $s$  on each of the sliding windows. The pixel intensity is compared with threshold  $T$ , calculated as:

$$T = m + k * s \quad (1)$$

where  $k$  is a positive number between 0 and 1, if we are detecting white characters on black background or  $k$  is a negative number between -1 and 0, if we are detecting black characters on white background. In the scenario of graffiti

<sup>1</sup>We have also tested using the H component from the HSV color space, which is known to be a more robust visual attribute to pixel intensity variation, hence the lighting variation; and we observed very similar results compared to grayscale framework on RGB color space.



**Figure 4: Image binarization. Left: image after pre-processing; right: image after binarization**

detection, we have observed cases of dark ink characters on a light colored surface and vice versa, so we are actually conducting pixel-wise comparisons with both thresholds:

$$T_{1,2} = m + k_{1,2} * s \quad (2)$$

where  $k_1 \in [0, 1]$  and  $k_2 \in [-1, 0]$ . The parameters, such as the size of the sliding window and the value of two  $k$ , will affect the binarization results. Because of the various visual representations of the large number of images in the data set, we may predict that there is no global configuration that can fit all the data. As a result, we determine the parameters by specific input images; for example, the size of the sliding window is based on the size of the input image and the value of  $k$  is based on the entropy of the image, specifically, linearly correlated with the entropy value. We can see from Figure. 4 that the Niblack algorithm will delete a large area of background patches that have a smooth visual appearance and keep the object areas that always appear with a high standard deviation of intensity.

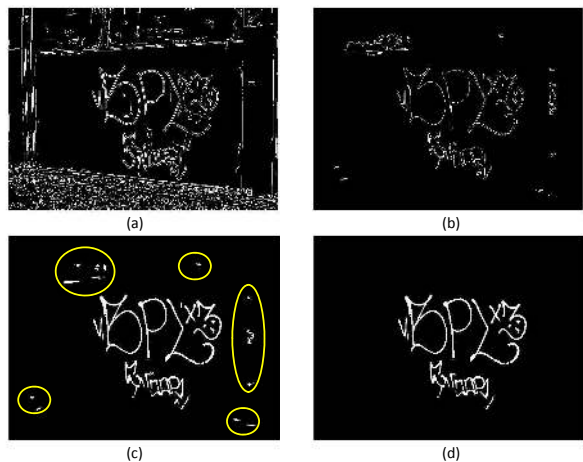
### 3.3 Character Detection

The binary image is organized by the connected components that are recognized as candidate objects. These candidates could be either the actual graffiti characters or the noisy background patches that cannot be deleted from the previous steps. The goal of the object detection task is to delete all these distracters and retain as many positive candidates as possible. We find that several visual attributes of character objects differ from the background objects, and the most important attribute is the edge contrast, with the idea derived from [15]. Edge contrast is defined as follows:

$$T_{edge\_contrast} = \frac{\{border\_pixels\} \cap \{edge\_detection\}}{\{border\_pixels\}} \quad (3)$$

The above threshold is defined based on the observation that character objects’ border pixels have a large portion of overlapping with the edge detection result from the original image, while the borders of background objects do not overlap much with the edge detection result. We can easily observe this property in Figure. 5. Figure. 5, (a) and (b) show the edge detection result and border detection result respectively. We can also see that the character components coincide with each other in (a) and (b) while the background components do not. Therefore, we will delete all the connected components whose edge contrast value is smaller than this threshold  $T_{edge\_contrast}$  and the image is further refined as shown in Figure. 5 (d).

Other attributes, such as the aspect ratio, length ratio, size ratio, border ratio, number of holes, smooth ratio, skeleton distance, and component position may also differentiate the positive objects from the noisy objects. Below we briefly introduce the functionality of the above criteria:



**Figure 5: Edge contrast.** (a) edge detection; (b) border detection; (c) noisy patches (in yellow circle) before comparing edge contrast; (d) after elimination.

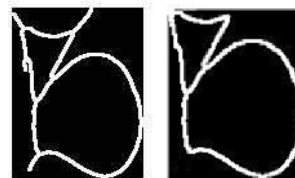
1) Aspect ratio: The printed characters usually have the aspect ratio of 7:5 or other ratios close to this value, which relies on the font or style of the character. The graffiti characters, with no exception, will follow this approximate aspect ratio. So we can exclude those connected components with much larger or smaller aspect ratios, because they are very unlikely to be characters.

2) Length ratio, size ratio, and position: Several background objects in the graffiti images can be deleted based on their extreme length or size. The character components in the graffiti images usually appear in formal shape and will locate as the focus of the photo. Extremely large or long components and components on the border of the images are usually noisy components. These noisy components often are windows or door frames.

3) Number of holes: Image patches from the background may have very rough textures or crude surfaces. The components derived from these areas have an irregular pattern with lots of holes or loops. The components derived from characters, on the other hand, have a more stable and consistent pattern with a limited number of holes. We will empirically define a threshold  $T_{holes}$  to exclude components with too many inner loops.

4) Smooth ratio: Graffiti characters are painted with oil or ink, and the oil paint itself is rough regardless of what kind of surface it is painted on. On the other hand, the background patches could be part of a very smooth surface. We define the smoothness of a connected component by its standard deviation value. The graffiti components show a moderate level of smoothness as indicated by a modest standard deviation; while some background components show perfect smoothness indicated by a near-zero standard deviation, which means the intensity values throughout the components are almost the same. We thus can exclude those components with a very small standard deviation value, because they are very unlikely to be a graffiti component.

5) Border ratio: The refinement criteria of border ratio are derived directly from the field of traditional character recognition. The characters, whether they are handwriting or graffiti, are composed of strikes, and the shape of the



**Figure 6: Unnecessary branch cut**

strikes is different from random patches. If the border ratio is defined as the proportion of border pixel to the total pixel, the components of strikes should have a much larger border ratio than random patches. Therefore, we will exclude the components with a small value of border ratio because they are very likely to be background components.

6) Skeleton distance: The notion of skeleton distance is also derived from the traditional character recognition field. We first conduct the inside loop filling operation as introduced in the following subsection, then extract the skeleton of the components, and further calculate the distance for each of the skeleton pixels. The distance of a skeleton pixel is defined as the minimum distance of the skeleton pixel to a pixel that is not in this component. Next, we gather the mean and deviation statistics of all the skeleton distances. If the component is a character, then the mean and deviation of the skeleton distance should both be small because of the consistent thickness of the strikes. Otherwise, it is more likely to be a noisy component.

The above background exclusion criteria are used sequentially and lead to a joint result that excludes all of the background components and retains as many character components as possible. For the threshold value used in each detection criteria, we conservatively select the one that keeps all the positive components and eliminates as many negative components as possible for all the images.

### 3.4 Image Refinement

The extracted character components need to be further refined to better serve the future steps of recognition or matching. A series of techniques is designed and introduced as follows:

1) Inside loop filling: Even though we have excluded the background components that have large numbers of inner loops, the remaining character components will inevitably have holes due to the rough quality of the painting. The oil paint or ink is not thick so small spots will be left unpainted within the strokes of the character. We apply the filling algorithm that detects the small holes inside the strokes and fill the holes. This step is essential for the later step of skeleton extraction, because the small holes inside the stroke may cause unnecessary branches of the skeleton.

2) Skeleton extraction: We use the thinning algorithm [7] to extract the skeleton structure of the character. We set the number of iterations to infinite so that the iteration repeats until the image stops changing and results in a single-pixel-width skeleton. If we define the degree of a pixel as the number of non-zero pixels from its 8 neighbors, then we can further define pixels in the components as endpoints if their degree equals 1, inline points if their degree equals 2, or junction points if their degree is more than 3.

3) Unnecessary branch cut: For certain printed uppercase English letters, there are at most 4 endpoints, such as the letters “H” and “K”. On the other hand, the skeleton ex-



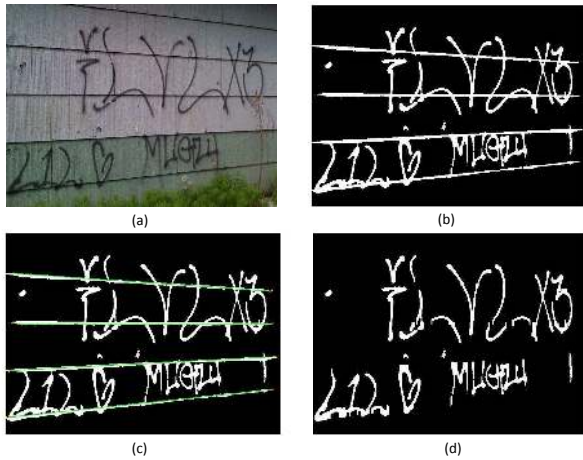


Figure 7: Background stripe elimination example

traction results usually have a much larger number of endpoints. The large number of unnecessary branches is usually caused by the skeleton extraction results from raw edges of the original character. The branches are defined as the edges linked by an endpoint and a junction point, so we examine all the branches and compare their length with the neighboring edges and longest edge. We then cut the branches shorter than a threshold because they are very likely to be unnecessary branches. A sample branch cut result of letter “B” is shown in Figure. 6.

4) Background stripe elimination: Background stripes (as shown in Figure. 7 (a)) have a very similar pattern to the character strokes, so they usually cannot be eliminated during the initial character extraction stages (as shown in (b)). These background stripes usually come from some solid background structure such as the edges and frames of the architecture. We choose to use hough transform because it is a good detector of straight lines. The hough transform line detection results are shown in (c) in green. We then apply algorithms to eliminate the four detected horizontal lines without breaking the vertical character strokes, as shown in (d). Specifically, we delete all the pixels connected along the detected hough lines, then reconnect the components which are originally connected, such as the separated vertical stroke.

### 3.5 Graffiti Retrieval

The key operation that links the character detection process to graffiti retrieval is to effectively bound each of the individual connected components (candidate characters) into meaningful strings with a larger bounding box.

The left image in Figure. 8 is the bounding box result for each of the individual connected components. We can see that each character is bounded by a single box; however, the retrieval result of each character doesn’t help the retrieval of all the graffiti images. Similar to OCR, we are seeking meaningful character sets, or a string of characters, that can be considered as a proper retrieval unit. We have proposed rules to combine multiple geographically aggregated components into a larger component, such as components close enough to each other in horizontal direction. Specifically, we merge two individual components together into a larger bounding box if the  $y$  coordinate value of the center of one component falls into the range of the other component in  $y$  direction. Then we repeat this operation until no more com-

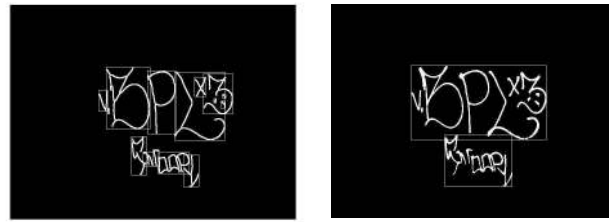


Figure 8: String construction with bounding box

ponents are added in. The combination results are shown as the right image in Figure. 8. We can see the proposed combining rule results in two strings, which are “vBPLx3” and “Snoopy”. The proposed rule does not apply to characters that are written in a vertical or diagonal direction.

After this step, any traditional OCR techniques, such as handwriting recognition techniques, can be applied to recognize the characters in the extracted strings. The characters are recognized based on the individual connected components extracted as in Figure. 8: left. Then the recognition results of each character are organized together based on the string extracted in Figure. 8: right in horizontal order. We are using the template matching method that matches the character patch with each of the templates (0-9, A-Z and a-z, created as universal template [12, 10]) and find the best match. The matching score, or the semantic-wise retrieval score, between two strings is defined as the length of the Longest Common Subsequence (LCS),

$$D_s(s, t) = |LCS(s, t)| \quad (4)$$

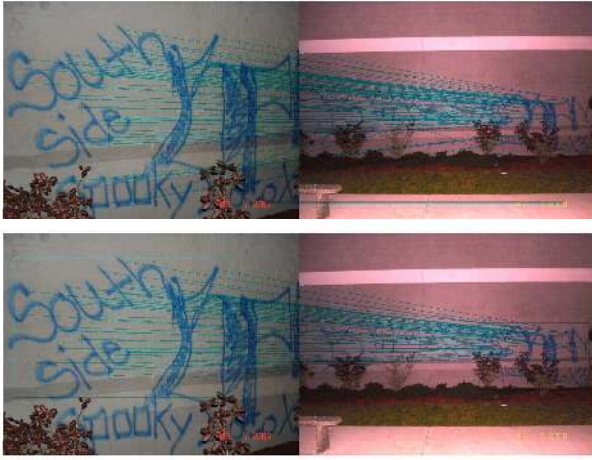
where  $s$  and  $t$  are two strings. The LCS does not merely count the frequency of appearance of the character; it also requires the sequence of the appearance of the corresponding character to be the same. The proposed metric is more reasonable for the semantic-level matching of graffiti words. Other types of recognition techniques can also be applied here; however, these are not the focus of this work.

For image-level matching of the two string components, we count the number of matches of interest points in the two string patches as the retrieval score. We have found that this matching metric performs better than having the score normalized with the total number of interest points detected. For one interest point  $k$  in string patch  $S_i$ , we will calculate the Euclidean distance of the SIFT descriptor<sup>2</sup> [8] from  $k$  to all the interest points in the other string patch  $S_j$ , and find the closest distance  $d_1$  and the second closest distance  $d_2$ . A match is considered to be found if the ratio  $d_1/d_2$  is smaller than a threshold (0.7 in this work). We count the total number of matchings between the interest points of two string patches.

$$D_i(s, t) = |Match(s, t)| \quad (5)$$

There are two major benefits for the proposed interest point matching scheme compared to the traditional interest point matching scheme that is conducted on the entire image. First, the interest points in the proposed framework are only extracted from the neighborhood of the character components as discovered in Figure. 8. Such design will dramatically decrease the influence of the interest points from the

<sup>2</sup>SIFT descriptor is known to be scale and rotation invariant, thus a suitable descriptor for local texture matching. A 128-dimensional feature vector is used in this experiment.



**Figure 9: Top: Interest point matching on the entire image with matching score 113; Bottom: interest point matching on the string patch (bounding box area) with matching score 71.**

background as shown in Figure. 9. The matching score for the top image is 113 and reduced to 71 for the bottom image. We have observed false-positive matches from the top image in Figure. 9, such as matches on date tags from the camera, matches from background trees, and false matches of the object. Such matches are eliminated with the proposed framework as shown in the bottom image. Second, the number of interest points in the bounding box is much smaller than in the entire image; thus, the number of comparisons and computation time are dramatically reduced.

The matching score  $R(i, j)$  between two images therefore can be represented by the maximum matching score between the string pairs from the two images. Specifically,

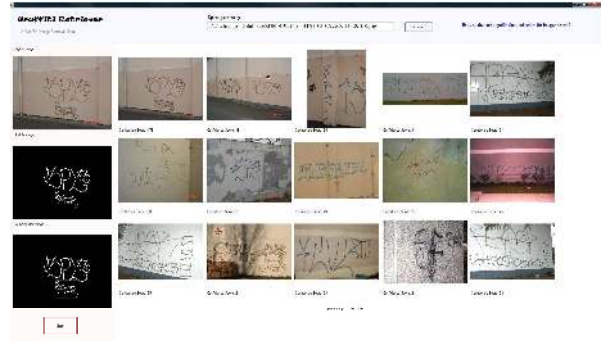
$$R(i, j) = \max_{s \in I_i, t \in I_j} (\alpha \overline{D}_i(s, t) + (1 - \alpha) \overline{D}_s(s, t)) \quad (6)$$

where  $R(i, j)$  is defined as the maximum matching score of the string pairs from two images.  $\overline{D}_i$  and  $\overline{D}_s$  are normalized image-wise retrieval score and semantic-wise retrieval score across all the available images in the database.  $I$  is the string set for a specific image.  $\alpha \in [0, 1]$  is the weight for image-wise retrieval score.  $\alpha$  is learned by maximizing the accumulated matching score across all the correct matches; while minimizing the accumulated matching score across all the false matches.

## 4. EXPERIMENT AND EVALUATION

The graffiti database used in the paper was provided by the law enforcement community of the Pacific Northwest. 62% (120/194) of the images in the database have clear character component detection; 38% (74/194) do not have clear detection of the character components, which means either the characters are eliminated with the proposed image refining methods or they cannot be distinguished from the background. Specifically, 4%(8/194) of the images do not have any textual parts, or the textual area is not visible.

We have built an interactive interface to conduct the retrieval operation as shown in Figure. 11. The user may upload a query image; the system then performs character detection and shows the binary result in the left column. Next, the user may start the retrieval operation on



**Figure 11: Interactive system screen shot. Top: upload menu; Left: query image processing result; Main: top 15 retrieval results**

the given database and get the top 15 retrieval results on the main panel. There are currently 194 graffiti images in the database and more are expected to come. The ground truth is constructed by human labor to find all the matching pairs or groups. The ground truth includes 14 extracted queries, with each query image having 1 to 4 matches in the database. The cumulative matching accuracy [9] curve is used as the evaluation metric with each value in the graph representing the average accumulated accuracy on a certain rank. The cumulative matching accuracy on a specific rank is calculated as the number of correctly retrieved matches on and before this rank divided by the total number of ground truth. Therefore, this curve is monotonically increasing along the axis of rank. The experiment results are shown in Figure. 12. The proposed bounding box framework achieves an average of 88% on cumulative accuracy on rank top 8, while the image-wise framework achieves an average of 75% on cumulative accuracy on rank top 8. These results show the advantage of the proposed framework on cumulative retrieval accuracy. Both frameworks achieve similar performance on rank top 1. More results of the proposed framework can be found in Table 1.

It is easy to understand why matching on bounding box framework outperforms the matching on the entire image. The query image may share some similar background patches with an unrelated graffiti image in the database; thus, there could be a large number of false-positive matches coming from the background to overwhelm the matching of the actual character area. Similar background patterns can be easily found in graffiti images. It is therefore essential to extract the meaningful character components from the background. Figure. 10 shows the comparison result between the two framework on the example image.

On the other hand, the improvement in computation efficiency for the proposed bounding box framework is also noticeable. Consider the query image in Figure. 11. The number of interest points of the query image and top 1 retrieval result are 1425 and 2425 respectively; after bounding box extraction, the number of interest points in the best matched bounding boxes are 75 and 87 respectively. As a result, the number of actual key comparisons is reduced to less than 1/400 of the original scale under the new framework. Such improvement is significant in large-scale image retrieval tasks.

The semantic retrieval score contributes less than the image retrieval score. This is because of the inherent difficulty of the semantic-level understanding of the graffiti charac-



Figure 10: Top 6 retrieval results under two frameworks for the query image in Figure. 11: Top row (a) is derived with the bounding box framework; bottom row (b) is derived with the image-based framework. The correct matches are circled with red boxes and corresponding matching score is indicated below each image.

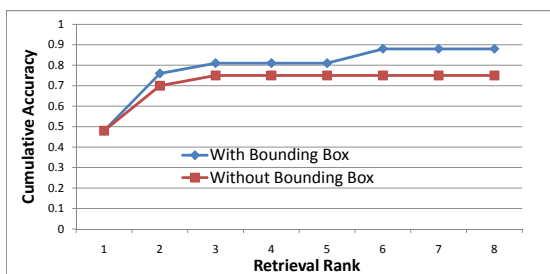


Figure 12: Comparison between cumulative accuracy curve (CAC) with bounding box framework and CAC without bounding box framework.

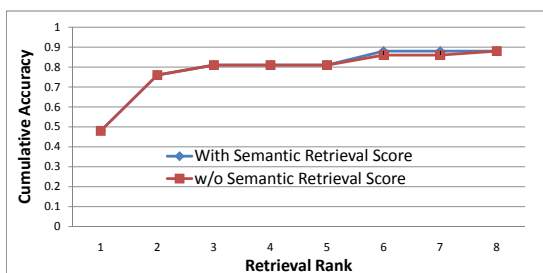


Figure 13: Comparison between cumulative accuracy curve with semantic retrieval score and cumulative accuracy curve without semantic retrieval score

ters. The character recognition results for the two strings in the example image are “vKPLX?” and “VXVJAR”. The maximum semantic retrieval score is 3 in this case (between “vKPLX?” and “?PLx3” from the top 1 retrieval result). The symbol “?” indicates a failed detection; in other words, there is not enough confidence to assign any value. This value is not as convincing as the image-wise retrieval score based on the current result. Correspondingly, the image-retrieval score will dominate the final matching function of Eqn. 6. For example, we got a semantic-level score of 3 and image-level score of 36 for the top 1 score in Figure. 10. The improvement achieved by integrating the semantic-wise retrieval score can be found in Figure. 13.

**Strengths and weaknesses:** The proposed system achieves

better retrieval performance compared to solely applying either image-base retrieval or OCR-related retrieval. The bounding box framework not only improves the accuracy of the local feature matching but also reduces the computation burden by eliminating unnecessary interest points. Reducing the number of false matches by applying geometric constraints is another well known technique to improve matching and retrieval results. However, based on current scale of database, we didn’t observed prominent improvement by applying the geometric constraints. The semantic-level understanding of graffiti images, on the other hand, is not equally satisfactory, as shown in Figure. 13. It requires us to bring in better semantic understanding techniques without sacrificing the computation efficiency. This weakness suggests a path for future work on the graffiti retrieval task as described in the next section.





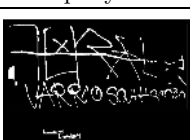

## 5. CONCLUSIONS AND FUTURE WORK

We have proposed an efficient graffiti image retrieval system that uses the character detection results and integrates both image-level understanding and semantic-level understanding of the graffiti characters. The experiment result has shown the bounding box framework is both efficient and effective in the graffiti retrieval task, especially when compared to traditional image retrieval framework. Our proposed system makes 4 primary contributions: a) Effective character extraction and noise elimination techniques to detect the character components in graffiti images; b) semantic-level bounding of meaningful character strings; c) fusing image-wise and semantic-wise scores for integral retrieval result; d) an interactive interface for graffiti exploration and retrieval.

The proposed system potentially can be used on, for example, mobile platforms that take photos as inputs and retrieve related information by connecting to a remote database. It could also be used for off-line tasks like large-scale graffiti image organization or classification.

We want to extend the database to a much larger scale in the future, and we expect the geometric constraints will be necessary for false matches elimination. For another part of the future work, we want to apply more robust techniques to improve the semantic-level retrieval performance.



 query 1	 76	 35	 33	 27
 character extraction	 26	 26	 26	 26
 query 2	 76	 73	 48	 46
 character extraction	 44	 42	 42	 38

**Table 1: Two query examples under the proposed framework: The left 2 images in each case are the query image and character detection result; the other 8 images are the top 8 retrieval results, listed in decreasing order with regard to the matching score. The correct matches are bounded with red boxes.**

## 6. ACKNOWLEDGEMENTS

The graffiti images used in our investigation and demonstrated in this paper were provided by the law enforcement community of the Pacific Northwest. This work has been supported in part by the U.S. Department of Homeland Security Science and Technology Directorate and the National Visualization and Analytics Center<sup>TM</sup> at the Pacific Northwest National Laboratory. The Pacific Northwest National Laboratory is managed for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC05-76RL01830.

## 7. REFERENCES

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on PAMI*, 24(4):509–522, Apr 2002.
- [2] J. C. Caicedo, J. G. Moreno, E. A. Niño, and F. A. González. Combining visual features and text data for medical image retrieval using latent semantic kernels. In *Multimedia Information Retrieval*, pages 359–366, 2010.
- [3] W. Clocksin. Handwritten syriac character recognition using order structure invariance. In *Proceedings on Pattern Recognition*, pages 562–565 vol 2, Aug 2004.
- [4] A. K. Jain, J. eun Lee, and R. Jin. Graffiti-id: Matching and retrieval of graffiti images. *Proceeding on MiFor*, 2009.
- [5] M. Kankanhalli and Y. Rui. Application potential of multimedia information retrieval. *Proceedings of the IEEE*, 96(4):712–720, april 2008.
- [6] S. Kim, S. Park, and M. Kim. Central object extraction for object-based image retrieval. CIVR’03, pages 39–49, Berlin, Heidelberg, 2003. Springer-Verlag.
- [7] L. Lam, S.-W. Lee, and C. Suen. Thinning methodologies—a comprehensive survey. *IEEE Transactions on PAMI*, 14(9):869–885, sep 1992.
- [8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60, pp. 91–110, 2003.
- [9] H. Moon and P. Phillips. Computational and performance aspects of pca-based face recognition algorithms. *Perception*, Vol 30, pages 302–321, 2001.
- [10] M. Nadira, N. I. Nik Kamariah, M. Z. Jasni, and A. B. Siti Azami. Optical character recognition by using template matching (alphabet). In *National Conference on Software Engineering and Computer Systems*, 2007.
- [11] F. S. Panagiotis E. Trahanias, Konstantinos Stathatos and E. Skordalakis. Morphological hand-printed character recognition by a skeleton-matching algorithm. *J. Electron. Imaging* 2, 114, 1993.
- [12] A. Pandey, S. Sawant, D. Eric, and M. Schwartz. Handwritten character recognition using template matching, 2010.
- [13] V. Pervouchine and G. Leedham. Document examiner feature extraction: Thinned vs. skeletonised handwriting images. In *TENCON IEEE Region 10*, pages 1–6, Nov 2005.
- [14] M.-C. Yeh and K.-T. Cheng. A string matching approach for visual retrieval and classification. In *Multimedia Information Retrieval*, pages 52–58, 2008.
- [15] K. Zhu, F. Qi, R. Jiang, L. Xu, M. Kimachi, and Y. Wu. Using adaboost to detect and segment characters from natural scenes. In *Proc. International Workshop on CBDAR*, pages 52–59, 2005.