A Multimodal Data Mining Framework for Revealing Common Sources of Spam Images

Chengcui Zhang, Wei-Bang Chen, Xin Chen, Richa Tiwari, Lin Yang, Gary Warner
Department of Computer and Information Science, University of Alabama at Birmingham, Birmingham, AL, USA
Email: {zhang, wbc0522, chenxin, rtiwari, galabing, gar}@cis.uab.edu

Abstract— **This paper proposes a multimodal framework that clusters spam images so that ones from the same spam source/cluster are grouped together. By identifying the common sources of spam images, we can provide evidence in tracking spam gangs. For this purpose, text recognition and visual feature extraction are performed. Subsequently, a two-level clustering method is applied where images with visually similar illustrations are first grouped together. Then the clustering result from the first level is further refined using the textual clues (if applicable) contained in spam images. Our experimental results show the effectiveness of the proposed framework.**

Index Terms—**spam image, clustering, multimodal analysis, botnet, computer forensics**

I. INTRODUCTION

Spam analysis is one of the most important topics in cyber security since these unsolicited emails greatly impact our daily life. We can passively filter these spam emails in the mailbox to alleviate the vexation or actively stop them at the origins to eradicate the torment. As a matter of fact, the filtering technique is by far the most effective approach used in controlling spam emails [1, 2, 3]. The approach can only differentiate spam emails from non-spam ones; however, it cannot tell the origins of spam. On the contrary, finding the spam origins, namely spam gangs, is a different approach from filtering. The implementation of such approaches is not a trivial task since spammers hide their identities by masquerading email header with falsified sender information.

Hence, in order to stop unsolicited spam emails, it is essential to trace the origins of spam and bring down the botnets, a malicious program hidden in a group of computers which are remotely controlled by spammers (www.cnn.com/2007/TECH/11/29/fbi.botnets). This process also raises a legal issue where law enforcement officers are actively involved in spam eradication. The goal of this paper is to facilitate this process by providing scientific proof to the origins of spam.

There are relatively few organizations creating the vast majority of these unsolicited emails, using a variety of intentional obscuring techniques. One of the techniques is to use image spam which presents texts primarily as an image to avoid text-based filtering. Mehta et al. [4] reported that the occurrence rate of spam image in all spam emails is more than 30% in 2006, which also raises the problem of tracing the origins of spam images.

Spammers may vary the space between words and lines and also randomly add speckles to make each messages look unique to fingerprinting techniques such as MD5 (Message-Digest algorithm 5) [5], though all of them have the same texts. Other techniques to defeat traditional anti-spam technologies include the use of different colors, varying font size, or splitting up one word into two halves with a gap in between. In addition to texts, a spam image may also contain illustrations. For example, a watch selling spam may contain a picture of a watch to illustrate the product that is being advertised.

There are relatively few works in spam image identification [4, 6, 7]. All of them address the spam image filtering problem. For example, Byun et al. [6] proposed a classification method to model and identify spam images. In this study, we go one step further to track down the common sources of the spam distributors based on spam image clustering.

There exists a large amount of work on general image clustering [8, 9]. Just as in our study of spam image clustering, there are two key factors in performing image clustering. The first one is feature extraction, which is also a deciding factor since the effectiveness of the method controls how precise an image can be represented. It is hard to design a universal feature extraction method since it is heavily application-dependent. Some commonly used features include color [9], texture, and shape. In this study, our focus is on spam images, which have their own characteristics. By carefully choosing feature extraction methods, we are able to precisely represent these characteristics in a precise way. Another key factor is to find a similarity metric [8] that can best interpret the relationship among image data and thus render accurate clustering results. As an example in the literature, affine invariant metric [10, 11] is one of the most well-known in the computer vision field. Other commonly used metrics include Euclidean distance and Mahalanobis distance.

It is worth mentioning that spam image clustering is neither like a general natural image clustering nor a content based image retrieval problem for the following reasons. First, spam images are man-made images which do not have continuous texture features as natural images [6]. Second, spam images are generally produced as low resolution images with many random noises. Thus, compared to natural images, they have poorer quality and lack visual details. Third, large portions of spam images contain only text messages, which is seldom to see in

natural images. Hence, the spam image clustering is not a special case of general image clustering.

Since visual features and texts are two different media modalities, a multimodal clustering algorithm is proposed in this paper to take advantage of their combined strength. Through clustering, spam images whose visual effects and/or textural contents resemble each other are grouped into clusters, revealing common origins of those images.

The image spam clustering results provide a strong evidence to identify and validate spam clusters or phishing groups for investigating cyber-crimes. Chun et al. [11] proposed an approach that used clustering techniques to form relationships among email messages and group them into spam clusters. The spam clusters were evaluated using a visual inspection method of the corresponding fetched website thumbnails, which these emails pointed to. Our proposed method can not only automate this visual validation process, but also link visual similarity directly to the presence of spam clusters.

In brief, the proposed framework first performs the segmentation by which a spam image is segmented into text areas, foreground illustration areas, and background areas. To extract visual features, we construct color-code histograms of foreground illustrations as the color feature, use foreground illustration layout as another visual feature, and extract the texture features of foreground illustrations as the third visual feature. The proposed two-level clustering algorithm first calculates the image similarities in a pair-wised manner with respect to the visual features, and the images with similarities sufficiently high are grouped together. In the second level clustering, text clues are also considered. A string matching method is used to compare the closeness of texts in two images, which is used as a criterion to refine the clustering results from the first level clustering. Our experimental results show the effectiveness of the proposed framework.

In the rest of this paper, we introduce the proposed method in Section 2. Section 3 presents the experimental results. Section 4 concludes this paper.

II. The Proposed Method

As aforementioned, spam is mainly used as an advertisement tool, i.e. Email Direct Marketing (EDM), for marketing a specific product or service, and thus, the embedded content most likely include the description and/or the illustrations of the product or service. Spammers receive advertising materials, such as sale information, product descriptions, and product images, through the manufacturers or sellers and then create the blueprint for spam image by embedding text messages into images which are often accompanied by illustrations and/or background textures for the advertisement. Finally, they generate multiple versions of those images using various obscuring techniques. Hence, we may assume that a set of similar spam images with various minor changes implies a common origin of image spam. Thus, the text content and foreground illustrations embedded in image spam play a key role in identifying the connection between spam images.

In this paper, a multimodal framework is proposed to reveal the origins of spam images through the following three steps: (1) image segmentation, (2) feature extraction and similarity calculation, and (3) spam image clustering, i.e., to perform a two-level agglomerative hierarchical clustering algorithm to associate related spam images.

A. Text Content and Foreground Illustration Extraction

To extract features from image spam, it is essential to distinguish foreground objects, i.e. text content and illustrations, from the background. To extract foreground objects, our strategy is to extract text areas through optical character recognition (OCR), followed by a threshold-based background detection step, and the rest of the areas can be considered as illustrations.

The text area segmentation is achieved by adopting the Microsoft Office Document Imaging (MODI) to identify recognizable characters in a spam image. MODI returns the recognized text content and their bounding rectangles. The foreground illustrations can be considered as sub-images which differ widely in their visual content across different images, and thus, difficult to characterize with any fixed set of visual features when it comes to differentiating illustrations from background. In this paper, we propose a simple yet effective method to differentiate foreground illustrations from background. The proposed method is based on two assumptions. The first assumption is that spam images must have sufficient foreground/background contrast in order to provide readability, which is usually the case as indicated by Byun et al. in [6]. The second assumption is that the background area is composed of one or more dominant colors which occupy significant portions of an image, often the largest or at least comparable to foreground illustrations. In addition, background usually demonstrates more uniformity than foreground such that background pixels tend to cluster together in the pixel intensity histogram while foreground pixels show a wide range of intensities.

On the basis of these assumptions, we first convert each pixel in a color image into a 6-bit color-code by taking the 2 most significant bits of each R, G, and B color components. This process replaces similar colors within a range by a single value, and transforms a RGB image to an index image.

In order to maximize the image contrast, we apply histogram equalization which conceptually spreads out the most frequent intensity values into adjacent empty bins and makes the histogram a uniform distribution [9] on the index image. Based on the second assumption, background pixel intensities usually have a relatively smaller range than that of the foreground and thus correspond to high frequency bin(s) in the histogram. Hence, we first calculate the average frequency of all bins as well as their standard deviation, and then use (mean + 2×Std) as a cutoff value to find all the dominant colors. This cutoff value will keep only the top 2% high frequency bin(s) which represent the dominant color(s) in the image, and thus background.

The segmentation masks procured at the end of this step will be used to extract visual features and build the pair-wise similarity matrices in the subsequent step. In Fig.1, the original spam image and the segmentation masks of the extracted text, foreground illustrations, and background are presented in (a)-(d), respectively, where the white areas correspond to the detected target areas.

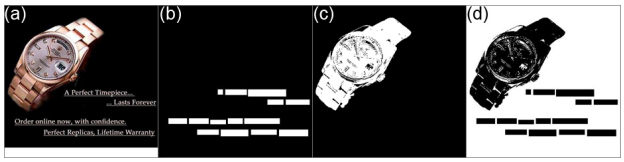

Figure 1. An example of spam image (a) together with the detected text content (b), foreground illustration (c), and background mask (d).

B. Feature Extraction and Similarity Measurement

The proposed framework requires four essential pair-wise similarity matrices to be constructed in advance based on the extracted features – color features, layout features, texture features, and text content. The first three matrices describe the closeness of foreground illustrations between pairs of spam images. The fourth matrix represents the similarity in terms of the text content between each pair of spam images. The detail of constructing each matrix is described below.

● Color features

Changing the color scheme and/or the layout of foreground illustrations are two commonly used tricks to create seemingly different spam images. Fig. 2 exemplifies two spam images with different illustrations at their upper right corners.

Figure 2. Spam images with illustration substitution

A large number of spam images contain foreground illustrations, such as artworks, pictures, and tables. We assume that these foreground illustrations have vivid color features to engage the viewers. In this study, the color-code histogram of foreground illustration areas is built to describe the color composition of foreground illustrations in an image. The similarity score between two images I_i and I_j in terms of color-code histogram is defined in Equation (1).

$$C(I_i, I_j) = 1 - \sqrt{\sum_{b=1}^{N} \left(H_i(b) - H_j(b)\right)^2} \qquad (1)$$

where H_i and H_j are the color-code histograms for images I_i and I_j, respectively; b is the bin index; N=64 is the total number of bins in the color-code histogram.

● Layout features

Spammers may change image spam by slightly adjusting the color composition of the foreground illustrations without significantly affecting the visuals perceived by human eyes. This kind of obfuscation may fail the color histogram-based detection, which motivates the consideration of shape and layout information of the illustrations. To compare the illustration layout difference between two images, the illustration segmentation masks of the two images (e.g., Fig. 1(c)) are used. We normalize their size, and then perform a XOR operation on the two masks for each pair of images. A small difference value between the layouts indicates high similarity in terms of the foreground illustration layout. The formal definition of the layout similarity matrix L is defined in Equation (2).

$$L(I_i, I_j) = 1 - \frac{\#\text{ of ls(trues) in } XOR(\text{mask}_i, \text{mask}_j)}{\text{The size of the normalized mask}} \qquad (2)$$

● Texture features

The third visual feature used in this study is the Gabor filter texture feature which simulates human vision in recognizing collinearity, parallelism, connectivity and repetitivity [14]. Gabor features provide better spatial localization and are often considered orientation and scale tunable edge and line (bar) detectors. A bank of such Gabor filters with an appropriate number of orientations and dilations are referred to as Gabor wavelets. A 2-D Gabor function is mathematically defined in Equation (3).

$$\begin{aligned} g_{\lambda,\theta,\varphi}(x,y) &= \exp\left(-\frac{(x'^2+\gamma^2 y'^2)}{2\pi\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\varphi\right) \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \qquad (3)$$

where λ is the wavelength; θ indicates the orientation; φ denotes the phase offset; σ is equal to 0.56λ by default.

To extract the texture features from the foreground illustration areas, we design a set of 5×5 2-D Gabor filters with 8 different orientations and a phase offset of $(0, -\pi/2)$, and then, convolve the 2-D Gabor filter bank on spam images, which results in 8 Gabor feature images. For each Gabor feature image, we calculate the mean and standard deviation of the foreground illustration areas as two features. Therefore, a 16-dimensional texture feature vector is produced for each spam image. The definition of the similarity matrix for Gabor filter texture features G, where v_i and v_j are the texture feature vectors of images I_i and I_j, respectively, is shown in Equation (4).

$$G(I_i, I_j) = 1 - \sqrt{\sum_{b=1}^{16} \left(v_i(b) - v_j(b)\right)^2} \qquad (4)$$

● Text features

It is essential to find some features other than color, layout and texture, which enable us to associate related images regardless of the alteration of these features. For instance, in Fig. 3, we show a pair of spam images that have entirely different foreground illustrations and layout but almost identical text content. This motivates us to consider using text content as a feature for spam image clustering.

Figure 3. An example of spam images with almost identical text content but totally different visual features.

The most commonly used tricks for text alteration include: (1) change text color, (2) adjust character

spacing, (3) increase or decrease line spacing, (4) modify margins, (5) switch between upper and lower case, and (6) substitute words in a template. Fig. 4 shows two very similar spam images with several minor differences. For example, the image on the left has two more illustrations on the top and a differently colored illustration in the bottom-left corner when compared with the image on the right. In addition, the image on the left has more background noise and its text content is slightly different from the one on the right by changing the wordings in the title and the text.

Figure 4. Examples of word substitution, illustration alteration/ replacement, and text & background color changes.

Comparing the similarity of text contents is achieved by using string alignment. Unlike exact string matching, string alignment algorithms compute the edit distance between two strings in order to find out how two strings could be optimally matched. We adopt the Needleman-Wunsch algorithm [12], which is one of the popular global alignment techniques for string matching. When applying the Needleman-Wunsch algorithm on two given query strings, we need to build a 256×256 substitution matrix (also called the scoring matrix) which describes how likely one character in a string changes to other states/characters over time. In the substitution matrix, each row (column) represents a character in the American Standard Code for Information Interchange (ACSII) table. This matrix plays a crucial role in computing the edit distance since OCR is error-prone. For example, the characters in each of the following sets could be substituted interchangeably in order to accommodate the known inaccuracy of OCR.

- Set 1: 'l','1','|','I','i'
- Set 2: 'o', '0', 'O', 'e', 'Q', 'c'

In addition, the adopted global alignment method is not 'case sensitive'. With the pair-wise global alignment dynamic programming technique, we can effectively and efficiently handle not only the character and line spacing alterations, but also the margin adjustments and word substitutions. This is because the pair-wise global alignment dynamically inserts or deletes space characters between regular texts in order to find the best match.

The formal definition of the similarity score S of two given strings is described as follows:

Assume two text strings A of length m and B of length n, where a_i and b_j indicate the i^{th} and j^{th} characters in strings A and B, respectively. Let D denote a dynamic programming matrix.

$$D \in \Re^{m\times n} \Leftrightarrow D = (d_{ij}) = \begin{bmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & & \vdots \\ d_{m1} & \cdots & d_{mn} \end{bmatrix} \quad d_{ij} \in \Re$$

The first step defines the base conditions of the matrix, which provides the initial values in the first row and the first column of D. In this study, the base conditions are defined as in Equation (5).

$$\begin{cases} d_{(i,1)} = SM(a_i, b_1) \\ d_{(1,j)} = SM(a_1, b_j) \end{cases} \quad i = 1:m; j = 1:n \tag{5}$$

where i and j are the row and the column indices of the matrix, representing the i^{th} position in the string A and j^{th} position in the string B; SM is the substitution matrix. There is no penalty for space insertion since we ignore the spaces that spammer adds to the text content.

After the matrix D is initialized, the next step is to iteratively fill the matrix according to Equation (6).

$$d_{(i,j)} = \max \begin{cases} d_{(i-1,j)} \\ d_{(i,j-1)} \\ d_{(i-1,j-1)} + SM(a_i, b_j) \end{cases} \quad i = 1:m; j = 1:n \tag{6}$$

Once the entire matrix is filled, the maximal score for aligning strings A and B can be retrieved at $d_{(m,\ n)}$. We calculate the normalized similarity score S as follows.

$$S = \frac{d_{(m,n)}}{2 \times \max(m,n)} \tag{7}$$

The formal definition of the similarity matrix of the text content features T is shown in Equation (8).

$$T(I_i, I_j) = 1 - S(I_i, I_j) \tag{8}$$

where I_i and I_j are two images; $S(I_i, I_j)$ is the similarity score of text content between I_i and I_j.

To this end, we have constructed all the four matrices (color code histogram, foreground illustration layout, Gabor filter textures, and text content) for describing the similarity between spam images based on different features.

C. Spam Image Clustering

Based on the extracted text content and foreground illustrations, we can categorize spam images into four different types. The first type of spam images contains mainly text (type-T); the second type contains mainly foreground illustrations (type-I); the third type contains a mixture of text and illustrations (type-M); while in the fourth type, neither text nor illustrations can be detected.

The proposed multimodal clustering is a two-level agglomerative hierarchical clustering algorithm. The first level uses the color code histogram, visual layout features, and Gabor filter textures for generating the initial clusters based on the sub-images, i.e. foreground illustrations, and the second level clustering merges related clusters from the first level based on the text content similarity. By means of the multimodal clustering, we take advantage of the complementary nature of the two modalities (visual and text clues).

In detail, the first level of the proposed clustering algorithm is intended for illustrated images (Type-I and Type-M). For illustrated images, if two images have highly similar illustrations in terms of their color schema,

spatial layout, and foreground textures, they are likely created by the same spam group. The three visual features, i.e. the color, layout, and texture of the foreground illustrations, play an important role in illustrated image spam clustering. These features have their own advantages and complement each other in clustering similar spam images.

Assume that there are n illustrated images, among which m illustrated images are similar to a given image c Given the query image I_x, we can generate three ranked lists (*LH*, *LL and LT*) with the top image in each list being the query image itself. Images in the three lists are ranked according to their similarities to the query image I_x in terms of the color-code histogram feature (*LH*), foreground layout feature (*LL*), and foreground texture feature (*LT*), respectively.

We then compare the three lists and merge similar images in them to form the answer set for the query image I_x. An example is given in Fig. 5 to illustrate how to find similar images for a query image. Suppose all the images in *LH* above index y form the set SH_y; all images in *LL* above index y form the set SL_y, and all images in *LT* above index y form the set ST_y. We collect all the y values where $SH_y = SL_y$ or $SL_y = ST_y$ or $ST_y = SH_y$, and the normalized similarity values for the images in all three sets (SH_y, SL_y, and ST_y) must be greater than 0, i.e., the similarity is greater than 50%. A set Y is formed to hold all those y values.

Suppose y_{max} is the maximum value in Y, where $y_{max}<n$ ($y_{max} = n$ indicates the equality of the entire image set; $y_{max} = 1$ indicates the equality of the query image to itself). y_{max} indicates that there are at least two lists, in which the foreground illustrations of the top y_{max} images are very similar to the query image in terms of the visual features that the two lists represent. The top y_{max} images that the two (or three) lists agree on thus form a cluster with a high confidence and are removed from the image set. This process will be performed iteratively by randomly selecting a query image from the remaining images until the entire image set is processed, resulting a set of image clusters that highly agree on at least two visual features.

The second level of the proposed multimodal clustering analysis is intended for grouping spam images containing text content (Type *T* and Type *M*). The basic idea at this level of analysis is borrowed from the pyramid scheme which increases the size of scheme by enrolling more and more members into it. Assume we have a set of images S. By giving a query image q($q \in S$) with a threshold value *Th* (in our case its value range is 95% - 99%), we can find a set of images N ($N \subset S$) such that the text content distance $d_{(q,n)}$ (where $n \in N$) between the query image q and each image n in N is less than *Th*. The newly discovered images are then used as the new query images for finding more similar images in S that meet the requirement on threshold distance values. This expanding process will stop when no more qualified new image can be added. These related images form a cluster *CT* based on the distance of the text content and are removed from the image set. This iterative process stops when the entire image set is processed.

In the cluster merging step at the second level, assume the initial clusters found in first-level are CI_x where x = 1 to n, and n is the number of the initial clusters. Each time when a new cluster *CT* at the second level is formed, we examine whether *CT* and CI_x have common member images by taking the set intersection. We then merge each such *CI* into *CT* and remove those *CI*(s) from the initial cluster set. This merging step is performed for each newly generated *CT* and stop when they are all processed. The resulting clusters will thus include not only those refined *CT*s output from the second level clustering, but also those 'dangling' *CI*(s) from the initial clustering. A *CI* is said to be 'dangling' if it does not have any common member with any *CT* from the second level clustering.

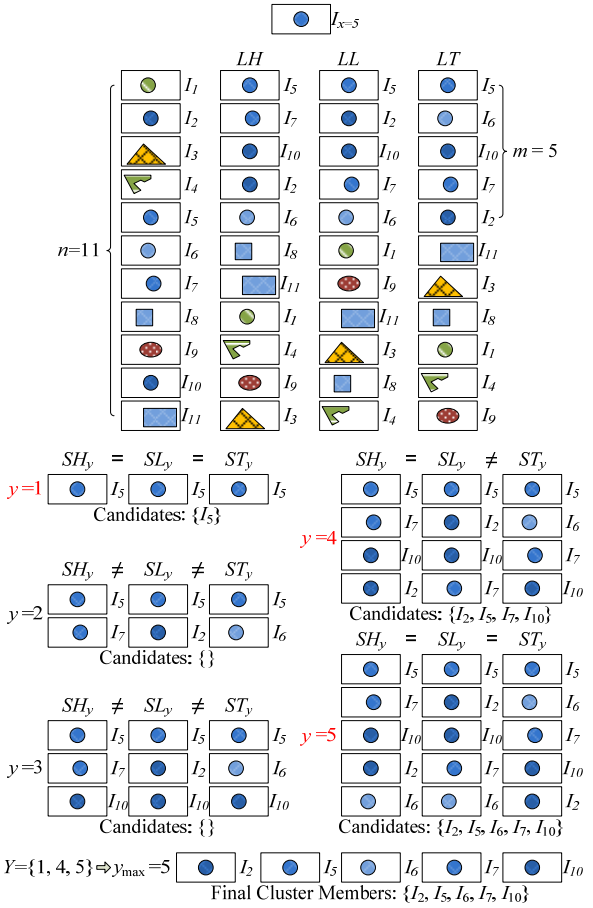

Figure 5. An example to illustrate the details of the first level clustering.

III. EXPERIMENTAL RESULTS

The spam images used in our experiments consist of those extracted from three month of emails manually identified as spam. We collect a high volume of spam through the use of "catch all" email addresses. A "catch all" configuration accepts mail for all possible addresses at a given domain. One common technique spammers use to "harvest" new target addresses is to send emails to randomly generated user IDs at well-known domains. Mail which does not "bounce" or reject is assumed by the

spammer to have been delivered. Because a "catch all" address configuration accepts ALL mail, spammers treat all tested addresses as valid for its domains.

We test our algorithm on 2063 spam images and 19 simulated spam images which are reproduced from 6 spam images as a test dataset by varying the color scheme and rearranging the foreground objects. In order to evaluate our algorithm, we manually classified our dataset into 475 classes based on their visual similarities and textual content. We use this manual clustering result as the ground truth to evaluate the proposed algorithm and compare its performance with other methods.

To compare our clusters with the ground truth, we use V-measure, a weighted harmonic mean of homogeneity (*hm*) and completeness (*cm*), presented by Rosenberg and Hirschberg [13]. V-measure is a conditional entropy-based method to evaluate the clustering results and is independent of the clustering algorithm being used. The mathematical expression for V-measure can be written as

$$V_\beta = \frac{(1+\beta^2)\times hm \times cm}{(\beta^2 \times hm)+cm} \quad (9)$$

where β is a constant, which if greater than 1 would mean that the *cm* is weighted β times more strongly; otherwise *hm* is weighted more in the calculation. In this study, we compare our clustering results with the ground truth using this measure with varying β values. The experimental results are as detailed below.

A. Parameters Determination

Since our goal is to reveal the common origins of spam images, it is reasonable to emphasize the completeness more than the homogeneity in V-measure. This claim can be justified in the experimental results as detailed in **Error! Reference source not found.**. From Table 1, we can observe that with the decrease of the threshold value (*Th*), the homogeneity rapidly decreases while the completeness slowly increases. In addition, we tested different β values (ranging from 1 to 10) in order to find the best weight for the completeness in the V-measure. By carefully examining the results in **Error! Reference source not found.**, we found that the clustering performance is quite reasonable when β = 3 for the following reasons: 1) in all cases, the increase (the second to fifth rows in Table 1) or decrease (the first row in Table 1) of V values slows down when β >= 3. In other words, the performance starts to converge when β >= 3; 2) using a too large β value may cause bias in performance evaluation since the current (limited) dataset may not reflect the true data distribution of image spam.

TABLE 1. THE PERFORMANCE OF THE PROPOSED METHOD

Th	#	h	c	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
99%	528	0.98	0.97	0.976	0.974	0.973	0.973	0.973	0.973	0.973	0.973	0.973	0.973
98%	492	0.96	0.98	0.970	0.975	0.977	0.977	0.978	0.978	0.978	0.978	0.978	0.978
97%	471	0.95	0.98	0.963	0.974	0.978	0.979	0.980	0.980	0.981	0.981	0.981	0.981
96%	454	0.93	0.98	0.953	0.971	0.977	0.979	0.981	0.981	0.982	0.982	0.982	0.982
95%	419	0.88	0.98	0.931	0.962	0.972	0.977	0.979	0.980	0.981	0.982	0.982	0.982

Th: Threshold value for level-2 clustering
#: number of cluster
h: homogeneity
c: completeness
V: V-measure with beta = 1…10

After the weight for the completeness is determined, we examine the impact of different threshold values on the second level clustering. A higher threshold value (*Th*) indicates that the criterion for grouping images based on similar text content is more stringent. The experimental results show that the proposed method with various threshold values (ranging from 95% to 99%) produces 528, 492, 471, 454, and 419 clusters, respectively, while the ground truth has 475 clusters. The experimental results show that as the threshold value decreases, more images were merged such that less cluster were produced. In addition, when the threshold value decreases, the corresponding V_3 values (see Table 1) are 0.973, 0.977, 0.978, 0.977, and 0.972, respectively. This shows that when *Th* = 97%, the proposed method can produce 471 clusters, which is very close to the ground truth, with the highest V_3 value.

B. Performance Evaluation

In this section, we conduct two experiments to evaluate the performance of the proposed clustering framework. The first experiment compares the performance of the proposed spam image clustering framework with two existing general image clustering methods which adopt Scale Invariant Feature Transform (SIFT) features and GIST features for clustering, respectively.

Scale Invariant Feature Transform (SIFT)

We use SIFT features [15] to correspond spam images and group them into different clusters. SIFT detects image features by building an octave of difference of Gaussian (DOG) images and finding local extrema in a scale space. The extracted SIFT feature descriptors are not affected by image scaling or rotation, and are proved to be robust in matching objects across images by many previous works including our own [16]. We compute SIFT features for each spam image and detect matches for all pairs of them. Due to the high precision of SIFT matches, we consider images with more than 20 matches to contain the same content. After pair-wise matching, spam images are divided into a set of clusters.

GIST Feature

To cluster spam images from the same origin, we also use GIST as a global feature to describe the image content [17]. GIST encodes the image structure by accumulating oriented edge energy at multiple scales into coarse spatial bins. It has been proved effective in detecting images with similar structures and semantics [18, 19].

We compute GIST for each image over three scales (from coarse to fine) using 8, 8 and 4 orientations aggregated into 4×4 spatial bins. We also down-sample the RGB channels of each image into 16×16 pixels and concatenate it to GIST. The descriptor is then normalized to unit length. To cluster images, we use hierarchical clustering on mean distances among clusters, and the optimal results are observed at a cutoff value at 1.15.

The experimental results are demonstrated in Table 2 in which the V_3 values of the PM (the proposed method), SIFT and GIST are 0.972, 0.913, and 0.766, respectively. It is obviously that the proposed method has the highest V_3 value among all three methods.

TABLE 2. THE PERFORMANCE OF PM, SIFT AND GIST

Method	# Cluster	Homogeneity	Completeness	V_3-measure
PM	412	0.881	0.984	**0.972**
SIFT	690	0.892	0.916	0.913
GIST	326	0.807	0.761	0.766

Clustering spam images with SIFT features often introduces false positive merges, which may be due to the low quality and noise obscuring in the spam images. Moreover, the experimental results show the poor performance of using GIST features in spam image clustering. This is because GIST features are commonly used to recognize scene in natural images; however, as we mentioned in Section 1, the properties of natural images and spam images are quite different. The experimental results suggest that spam image clustering is not a special case of general image clustering. In the second experiment, we evaluate the performance of the proposed clustering method (PM) by comparing it to the following approaches: (1) the first level of the proposed clustering framework (PM-L1), i.e. without using text content feature, (2) the second level of the proposed clustering framework (PM-L2), i.e. using text content feature only, (3) replace the first level clustering in the proposed framework with the SIFT-based clustering. In particular, in Table 3, SIFT-A represents the SIFT-based first level clustering applied to Type-I images and Type-M images, while SIFT-B means that the SIFT-based first level clustering is applied to all the images. Table 3 shows the results from the second experiment.

TABLE 3. THE PERFORMANCE OF PM, PM-L1, PM-L2, SIFT-A, AND SIFT-B

Method	# Cluster	Homogeneity	Completeness	V_3-measure
PM	412	0.881	0.984	**0.972**
PM-L1	469	0.582	0.893	0.847
PM-L2	306	0.787	0.977	0.954
SIFT-A	385	0.830	0.981	0.963
SIFT-B	425	0.821	0.972	0.955

Table 3 shows that the V_3 values of PM, PM-L1, PM-L2, SIFT-A, and SIFT-B are 0.972, 0.847, 0.954, 0.963, and 0.955, respectively. The proposed framework combining both visual and text features significantly outperforms the other four approaches. In addition, recall that the V_3 value of SIFT is 0.913 in the first experiment. By comparing SIFT with SIFT-A and SIFT-B, and comparing PM with PM-L1, the results suggest that using the text content feature in the second level clustering greatly improves the clustering results. After carefully examining the clustering results, we also observe that, the images in Fig. 4 can be successfully merged by using the text content feature. Further, the experimental results suggest that the use of both visual and text features performs better than the use of individual features from one single dimension.

C. The Performance on Different Type of Spam Images

Recall that we classify spam images into four different types. In our dataset, the composition of the four types of spam images is Type-T (53%), Type-M (35%), Type-I (11%), and the other unclassified images (1%). We analyze the performance of the proposed algorithm on each type of spam images. The V_3 values for Type-T, Type-M, and Type-I spam images are 0.963, 0.985, and 0.936, respectively. Table 4 summarizes the performance analysis on spam images of different types.

TABLE 4. PERFORMANCE ANALYSIS ON DIFFERENT TYPES OF IMAGES

Image Type	# Image	Homogeneity	Completeness	V_3-measure
T	1096	0.909	0.976	0.968
M	731	0.988	0.985	**0.985**
I	236	0.915	0.939	0.936

The results show that the Type-M images has the highest V_3 value since clustering images of this type can take the most advantage of both visual and text features.

D. The Performance on Simulated Spam Images

To test the robustness of the proposed framework, we further generate 19 simulated images from 6 spam images in our dataset by changing their foreground color scheme and the visual layout simultaneously. These simulated images are added to our dataset to test the robustness of the proposed framework. The experimental results show that the proposed framework can correctly merge all the simulated spam images into their correct clusters.

The simulated spam experiment suggests that our method can handle various modifications on spam images such as adjusting color schema, altering image layout, and/or changing the location of text areas. This result demonstrates the effectiveness and robustness of the proposed multimodal clustering framework which successfully integrates information from two different modalities, visual and text content, for producing better result in image spam clustering.

IV. CONCLUSIONS AND FUTURE WORK

The proposed multimodal framework clusters image spam with common traits to reveal their origins. This framework consists of three steps: (1) image segmentation, (2) feature extraction and similarity measurement, and (3) spam clustering. First, spam images are segmented into text, foreground illustration(s), and background. Second, visual and text features from the foreground illustration(s) and text content are extracted for constructing similarity matrices. Subsequently, an agglomerative hierarchical clustering approach with two levels is performed. In the first level, the foreground illustration features, including color-code histogram, foreground illustration layout, and Gabor filter texture, are used to group visually similar images. In the second level, the results are further refined based on the text content similarity.

It is worth mentioning that the visual features and textual features exploited in this work complement each other in identifying commons origins for image spam. This is evidenced by our experimental results on 2063 spam images collected over a three-month period. The clustering results are verified against the manually generated ground truth using the V-measure. Through visual inspection, spam images in the same cluster are found to be closely related, regardless of the variations in the image scale, background color and/or texture, and spatial placement of text and/or illustrations in the foreground.

The next stage of the research is to incorporate other text clues extracted by OCR into the clustering process,

such as URLs (if any) from image spam. There are two possible ways of using URLs in image spam: 1) using the similarity of two URLs as another indication of the common source of spam, and 2) examining the visual similarity of the two website images pointed to by the two URLs as another indication of common spam origin.

We believe the clustering results could provide law enforcement officers with important scientific evidence for investigating the illegal spam propagations.

ACKNOWLEDGMENT

The work of Chengcui Zhang was supported in part by the UAB ADVANCE program and NSF DBI-0649894.

REFERENCES

[1] X. Carreras and L. Mrquez, "Boosting trees for anti-spam email filtering," in *Proc. International Conference on Recent Advances in Natural Language Processing*, 2001, pp. 58-64.

[2] J. Clark, I. Koprinska, and J. Poon, "A neural network based approach to automated e-mail classification," in *Proc. IEEE/WIC International Conference on Web Intelligence*, Beijing, China, 2003, pp. 702-705.

[3] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, pp. 1048-1054, 1999.

[4] B. Mehta, S. Nangia, M. Gupta, and W. Nejdl, "Detecting image-based email spam using visual features and near duplicate detection," in *Proc. 17th International World Wide Web Conference*, 2008, pp. 497-506.

[5] R. Rivest, "The md5 message-digest algorithm," RFC 1321, 1992.

[6] B. Byun, C.-H. Lee, S. Webb, and C. Pu, "A discriminative classifier learning approach to image modeling and spam image identification," in *Proc. 4th Conference on Email and Anti-Spam*, 2007.

[7] C.-T. Wu, K.-T. Cheng, Q. Zhu, and Y.-L. Wu, "Using visual features for anti-spam filtering," in *Proc. IEEE International Conference on Image Processing*, 2005, pp. III-509-512.

[8] S. Zhang, C. Shi, Z. Zhang, and Z. Shi, "A global geometric approach for image clustering," in *Proc. of 18th International Conference on Pattern Recognition*, 2006, pp. 1244-1247.

[9] B. Malinga, D. Raicu, and J. Furst, "Local vs. global histogram-based color image clustering," University of Depaul, Technical Reports: TR06-010 (2006).

[10] A.W. Fitzgibbon and A. Zisserman, "On affine invariant clustering and automatic cast listing in movies," in *Proc. the Seventh European Conference on Computer Vision*, 2002, pp. 304–320.

[11] W. Chun, A. Sprague, G. Warner, and A. Skjellum, "Mining spam email to identify common origins for forensic application," in *Proc. 23rd Annual ACM Symposium on Applied Computing*, 2008, pp. 1433-1437.

[12] S. B. Needleman and C.D. Wunsch, "A general method applicable to the search for similarities in the amino of two proteins," *J. Mol. Biol.*, 1970, 48(3), pp. 443-453.

[13] A. Rosenberg and J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure," in *Proc. 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pp. 410–420.

[14] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of data," *IEEE Trans. Pattern Analysis Machine Intelligence*, 1996, 18(8), pp. 837-842.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, 2004, 60(2): pp. 91–110.

[16] C. Zhang, X. Chen, W.-B. Chen, L. Yang, and G. Warner, "Spam image clustering for identifying common sources of unsolicited emails," *Intl. J. of Digital Crime and Forensics*, in press.

[17] A. Friedman, "Framing pictures: The role of knowledge in automatized encoding and memory for gist," *Journal of Experimental Psychology: General*, 1979, vol. 108, pp. 316–355.

[18] J. Sivic, B. Kaneva, A. Torralba, S. Avidan, and W. T. Freeman, "Creating and exploring a large photorealistic virtual space," *Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[19] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics (SIGGRAPH 2007)*, 2007, vol. 26, no. 3, pp. 1-7.

Chengcui Zhang is an Assistant Professor of Computer and Information Sciences at University of Alabama at Birmingham (UAB) since August, 2004. She received her Ph.D. from the School of Computer Science at Florida International University, Miami, FL, USA in 2004. Her research interests include multimedia databases, multimedia data mining, image and video database retrieval, bioinformatics, and GIS data filtering.

Wei-Bang Chen is a Ph.D. candidate in the Department of Computer and Information Sciences at the University of Alabama at Birmingham. He received a Master's degree in Genetics from National Yang-Ming University in Taipei, Taiwan and a Master's degree in Computer Sciences from UAB. His main research area is bioinformatics. His current research involves microarray image and data analysis, biological sequence clustering, and biomedical video and image mining.

Xin Chen received her Ph.D. degree in Computer Science from the University of Alabama at Birmingham, USA, in 2008. Her research interests include Content-based Image Retrieval, multimedia data mining, and spatiotemporal data mining.

Richa Tiwari received her Bachelor's degree in Information Technology from India and received her Master's degree in Computer and Information Sciences from UAB in 2006. She is currently pursuing her PhD in the area of Knowledge Discovery and Data mining CIS department of UAB.

Lin Yang is a Ph.D. student in the Computer and Information Sciences Department at UAB. His research interests lie at the intersection of computer vision and graphics, which include multiple view geometry, visual surveillance system, and multimedia data mining.

Gary Warner joined UAB in 2007 as their first Director of Research in Computer Forensics. For the past seven years, he has been active in Information Sharing with Law Enforcement. He serves on the Technology Committee of the FBI's Digital PhishNet, co-chairs the Working with Law Enforcement Committee of the Anti-Phishing Working Group, and serves as a Handler with the CastleCops PIRT Team.