Chapter 22

CONTENT-BASED IMAGE RETRIEVAL FOR DIGITAL FORENSICS

Y. Chen, V. Roussev, G. Richard III and Y. Gao

Abstract Digital forensic investigators are often faced with the task of manually examining a large number of (photographic) images to identify potential evidence. The task can be daunting and time-consuming if the target of the investigation is very broad, such as a web hosting service. Current forensic tools are woefully inadequate: they are largely confined to generating pages of thumbnail images and identifying known files through cryptographic hashes. This paper presents a new approach that significantly automates the examination process by relying on image analysis techniques. The strategy is to use previously-identified content (e.g., contraband images) and to perform feature extraction, which captures mathematically the essential properties of the images. Based on this analysis, a feature set database is constructed to facilitate automatic scanning of a target machine for images similar to the ones in the database. An important property of the approach is that it is not possible to recover the original image from the feature set. Therefore, it is possible to build a (potentially very large) database targeting known contraband images that investigators may be barred from collecting directly. The approach can be used to automatically search for case-specific images, contraband or otherwise, and to provide online monitoring of shared storage for early detection of specific images.

Keywords: Digital forensics, image analysis, image retrieval

1. Introduction

Digital forensic investigations often require the examination of pictures found on target media. Two typical tasks are the identification of contraband images and the identification of case-specific images, the presence of which can establish a fact or a logical link relevant to the investigation. The essential problem is that current forensic tools are often ill-equipped to deal with the scale of the task. To illustrate, we

recently recovered approximately 34,000 image files on a randomly selected machine in our general-purpose computing laboratory. Note that this was a relatively old system with a very modest 6 GB hard drive and the images were mostly stored in the browser's cache. Even if an investigator were to spend a fraction of a second on each image, it would still require several hours to browse through all the images. The dramatic drop in prices of storage devices coupled with the leap in capacity (a 200 GB hard drive now costs about $100), will make the examiner's task even more difficult by removing any incentive for users to delete images. Thus, it is not unreasonable to expect that the hard drive of a home user could contain hundreds of thousands of images, while a web hosting service can have tens of millions of images. Clearly, examining all these images is virtually intractable, and investigators will need some means to narrow the search space.

The driving problem behind this work has been the identification of contraband images. This task consumes a significant fraction of the resources of our partners at the Gulf Coast Computer Forensics Laboratory (GCCFL). They have a clear and pressing need for a forensic tool that would allow the automated examination of images on a massive scale. Similar problems in traditional forensics (e.g., fingerprint identification) have been tackled by building large reference databases that allow evidence from previous cases to be automatically searched. Clearly, a system capable of automatically identifying contraband images on target media by cross referencing a database of known images could be of significant help to investigators. The problem, however, is that unlike other forensic artifacts, contraband images typically cannot be stored, even by law enforcement agencies, for future reference. Aside from the legal barriers, building a sizeable reference database to be used routinely by numerous agencies would be a challenging task. The storage and bandwidth requirements would be staggering. Scalability would be difficult to achieve as the replication and distribution of such highly sensitive images would have to be limited. Finally, a security breach at an image storage facility or misuse by authorized personnel could have serious implications.

A well-designed system should not rely on having access to the original images during the lookup process. Rather, it should have a single opportunity to access the original when it can extract and store some identifying ("fingerprint") information for later reference. Clearly, the fingerprint information should permit a high-probability match; but it should also be impossible to reconstitute any recognizable version of the original image. We believe that analytical methods for content-based image retrieval can address image analysis needs in digital forensics. This

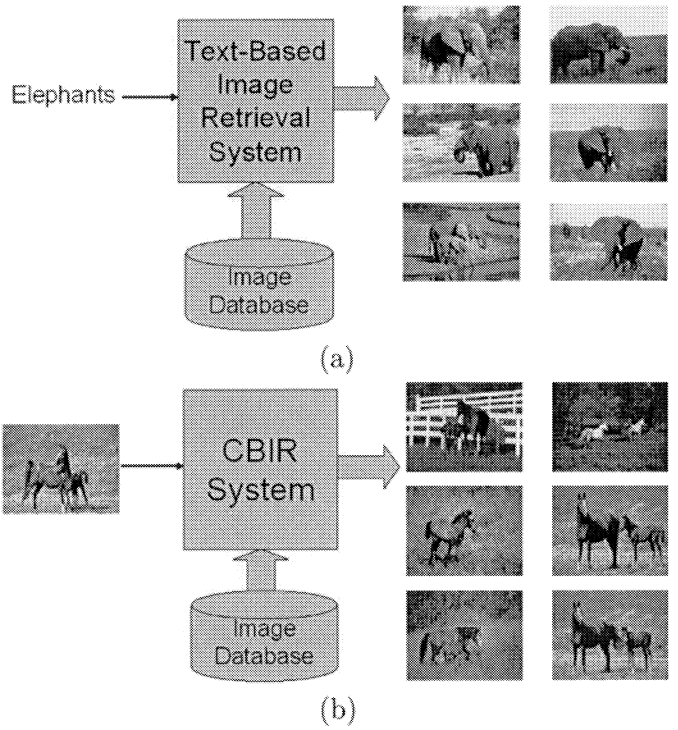

Figure 1. Schematic diagrams of: (a) text-based image retrieval system; (b) content-based image retrieval system.

paper is a first effort to evaluate the suitability of the approach and to present an architectural framework that would allow the deployment of a working system.

The following section describes previous work in content-based image retrieval, which forms the basis of our own work. Next, a set of experimental results is presented to validate the use of content-based image retrieval in digital forensic investigations. Finally, an architectural design (currently under implementation) is specified.

2. Content-Based Image Retrieval

2.1 Overview

Depending on their query formats, image retrieval algorithms roughly fall into two categories: text-based approaches and content-based methods (see Figure 1). Text-based approaches associate keywords with each stored image. These keywords are typically generated manually. Image retrieval then becomes a standard database management problem. Some

commercial image search engines, e.g., Google Image Search and Lycos Multimedia Search, are text-based image retrieval systems. However, manual annotation for a large collection of images is not always available. Furthermore, it may be difficult to describe image content with a small set of keywords. These issues have motivated research on content-based image retrieval (CBIR), where retrieval of images is guided by providing a query image or a sketch generated by a user (e.g., a sketch of a horse).

Many CBIR systems have been developed over the past decade. Examples include the IBM QBIC System [4], the MIT Photobook System [12], the Berkeley Chabot [11] and Blobworld Systems [1], the Virage System [7], Columbia's VisualSEEK and WebSEEK Systems [14], the PicHunter System [2], UCSB's NeTra System [9], UIUC's MARS System [10], the PicToSeek System [6] and Stanford's WBIIS [16] and SIMPLIcity systems [15].

From a computational perspective, a typical CBIR system views the query image and the images in the database as a collection of features, and ranks the relevance between the query and any matching image in proportion to a similarity measure calculated from the features. The features are typically extracted from shape, texture, intensity or color properties of the query image and the images in the database. These features are image signatures and characterize the content of images, with the similarity measure quantifying the resemblance in content features between a pair of images.

Similarity comparison is an important issue. In general, the comparison is performed either globally, using techniques such as histogram matching and color layout indexing, or locally, based on decomposed regions (objects). As a relatively mature method, histogram matching has been applied in many general-purpose image retrieval systems, e.g., IBM QBIC, MIT Photobook, Virage System and Columbia VisualSEEK and WebSEEK. A major drawback of the global histogram search is its sensitivity to intensity variations, color distortions and cropping.

In a human visual system, although color and texture are fundamental aspects of visual perceptions, human discernment of certain visual contents could potentially be associated with interesting classes of objects or semantic meanings of objects in the image. A region-based retrieval system segments images into regions (objects), and retrieves images based on the similarity between regions. If image segmentation is ideal, it is relatively easy for the system to identify objects in the image and to match similar objects from different images. Next, we review a CBIR system called SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) [15], which we use in our forensics experiments.

2.2 SIMPLIcity System

In the SIMPLIcity system, the query image and all database images are first segmented into regions. To segment an image, the system first partitions the image into non-overlapping blocks of size 4×4. A feature vector is then extracted for each block. The block size is chosen as a compromise between texture effectiveness and computation time. Smaller block sizes may preserve more texture details but increase the computation time. Conversely, larger block sizes reduce the computation time but lose texture information and increase segmentation coarseness.

Each feature vector consists of six features. Three of them are the average color components in a 4×4 block. The system uses the well-known LUV color space, where L encodes luminance, and U and V encode color information (chrominance). The other three represent energy in the high frequency bands of the wavelet transforms [3], i.e., the square root of the second-order moment of wavelet coefficients in high frequency bands.

To obtain these moments, a Daubechies-4 wavelet transform is applied to the L component of the image. After a one-level wavelet transform, a 4×4 block is decomposed into four frequency bands: LL (low low), LH (low high), HL and HH bands. Each band contains 2×2 coefficients. Without loss of generality, suppose the coefficients in the HL band are $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$. One feature is

$$f = \left(\frac{1}{4} \sum_{i=0}^{1} \sum_{j=0}^{1} {c_{k+i,l+j}}^2 \right)^{\frac{1}{2}}.$$

The other two features are computed similarly from the LH and HH bands. The motivation for using the features extracted from high frequency bands is that they reflect texture properties. The moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture. The intuition behind this is that coefficients in different frequency bands show variations in different directions. For example, the HL band shows activities in the horizontal direction. An image with vertical strips thus has high energy in the HL band and low energy in the LH band.

The k-means algorithm is used to cluster the feature vectors into several classes, each class corresponding to one region in the segmented image. Because clustering is performed in the feature space, blocks in each cluster do not necessarily form a connected region in the images. This way, segmentation preserves the natural clustering of objects in textured images and allows classification of textured images. The k-means algorithm does not specify how many clusters to choose. The system

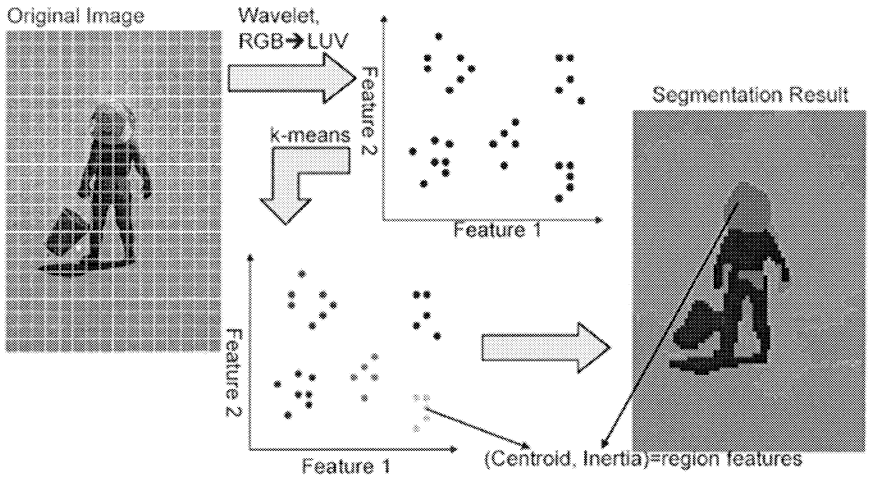

Figure 2. Schematic diagram of the feature extraction process.

adaptively selects the number of clusters, C, by gradually increasing C until a stopping criterion is met. The average number of clusters for all images in the database changes according to the termination criteria. Each region is represented by a feature vector (of dimension 6) that corresponds to the centroid of the cluster.

After segmentation, three extra features, normalized inertia [5] of orders 1 to 3, are calculated for each region to describe shape properties. The normalized inertia is invariant to scaling and rotation. The minimum normalized inertia is achieved by spheres. If an image is segmented into C regions, the image is represented by C feature vectors, each of dimension 9. Figure 2 illustrates the feature extraction process. Only two features for each image block are shown to make illustration easier. In the segmentation result, each region is represented by a distinct color.

The similarity between two images is computed according to an integrated region matching (IRM) scheme [8]. To reduce the influence of inaccurate segmentation, the IRM measure allows for matching a region of one image to several regions of another image (i.e., the region mapping between any two images is a many-to-many relationship). As a result, the similarity between two images is defined as the weighted sum of distances in the feature space, between all regions from different images. Compared with retrieval systems based on individual regions, the IRM approach decreases the impact of inaccurate segmentation by smoothing over the imprecision in distances.

3. Experimental Results

To evaluate the suitability of CBIR methods, we performed a number of experiments with the SIMPLIcity system. The experiments were designed to test its robustness against a number of typical transformed versions of the image that can be expected during an investigation. The first two were reductions in quality by varying the quality factor in JPEG images to 30% and 10%, respectively. Such variations can be expected for two reasons – to reduce storage requirements without noticeably impairing the visual perception (at screen resolution) and to provide (visibly) lower quality samples. Depending on the initial quality of the source images, the suitable values will vary. In our case, the vast majority of the pictures were taken with a 5 megapixel digital camera and we judged qualitatively that a quality value of 30% approximates the first scenario, whereas 10% approximates the second one.

Resizing is another common transformation applied for similar reasons, as well as to fit pictures into web pages. We tested three different versions at 512, 256 and 96 pixels (for the longer dimension) with the last one designed to simulate the common "thumbnailing" process. The last three transformations are 90 degree rotations and mirroring (vertical and horizontal) of images that can be expected during the processing of raw images.

The target database consisted of 5,631 photo images in JPEG format. The goal was to demonstrate the ability of the system to recognize an image when its altered version is submitted as the query. We applied image alteration to an image (called target image i) in the database. The resulting image i' is used as the query image and the rank of the retrieved target image i is recorded. The rank of image i is defined as the position of image i in the first 100 retrieved images. Clearly, a "good" system should return the original image at the top of the list (the best rank is 1). If image i does not show up in the top 100 retrieved images, it is considered a missed image.

We tested the system against the image alterations shown in Table 1. For each alteration, the average rank of all target images (excluding missed images) is computed; the results are given in Table 2. The experimental results indicate that image analysis techniques can significantly benefit digital forensic investigations. Of course, further study is warranted. Also, system-level issues such as performance, scalability and security must be addressed before a working prototype can be tested in a forensics laboratory. The following sections discuss the system design and the ongoing implementation effort.

Table 1. Alterations applied to query images.

ID	Alteration
JPEG30	Reducing JPEG quality to 10%
JPEG10	Reducing JPEG quality to 30%
Resize1	Resizing the image such that the largest of the width and height is 512 pixels
Resize2	Resizing the image such that the largest of the width and height is 256 pixels
Resize3	Resizing the image such that the largest of the width and height is 96 pixels
Rotation	Rotating the image by 90 degrees
Flip	Creating a mirror image
Flop	Creating a mirror image

Table 2. Experimental results for queries based on altered images.

Alteration ID	Missed Images (Miss Rate)	Average Rank
JPEG30	43(0.76%)	1.16
JPEG10	43(0.76%)	1.16
Resize1	43(0.76%)	1.16
Resize2	43(0.76%)	1.16
Resize3	43(0.76%)	1.16
Rotation	27(0.48%)	1.08
Flip	43(0.76%)	1.16
Flop	43(0.76%)	1.16

4. Design Goals

- *Service-Oriented Architecture*: The stored image feature sets are not contraband. However, even if they correspond to contraband images, we anticipate that the database will be administered by law enforcement agencies. Therefore, most software products will not be able to bundle such a database. Furthermore, the database will be a highly dynamic entity once a large number of law enforcement agencies become contributors.

- *Performance*: The system should be able to handle individual requests at rates that will allow investigations to proceed interactively. Current open source imaging software can generate thumbnail images at approximately 1000 images per minute using a single CPU. A working system should be able to perform at a similar rate or better (while providing a higher value service).

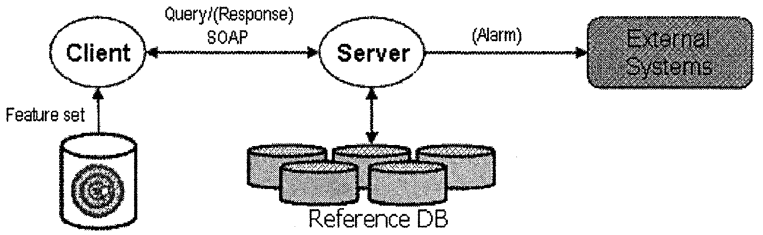

Figure 3. Architecture of a client/server image matching service. A client computes the feature set for one or more target images and issues queries to a server, which maintains a reference database of features for images of interest. The server indicates which images match the reference database and may alert external systems when matches occur.

- *Scalability*: The system should eventually be able to handle millions of images without a serious degradation in performance. This clearly implies that the system will have to incorporate replication and distributed processing as part of its original design.

- *Security*: The standard requirements for privacy, authentication and secure system administration apply. Recall that copies of the actual images are not stored. This makes the system legal and greatly mitigates the consequences of any security breach.

- *Flexible Deployment*: It should be possible to use the same architecture for forensics investigations and for preventive monitoring.

5. System Architecture

The architecture employs a client/server model (Figure 3). A client computes the feature sets of target images and submits them to the server. The server compares the submitted feature sets with those in a reference database. The client must keep track of outstanding queries and alert the user when image matches occur. Note that a match means that the feature set of the target image is close enough to a feature set in the reference database. Thus, false positives are a possibility, and these will have to be dealt with by the investigator.

The server has two basic functions:

1 Maintain the reference database of known feature sets. This includes adding and removing feature sets of images, as well as maintaining the integrity of the data and coordinating access to it. The latter two become non-trivial issues if the data and/or the processing are distributed for performance reasons.

2 Accept queries from clients and react accordingly. The server must first determine if the received feature set is a match, and then must either reply to the client, or take some other action, such as raising an alarm if the system is used for on-line monitoring.

The server is presented as a web service and uses a SOAP-based protocol to communicate with clients. The rationale is that the reference database is likely to be managed by a few law enforcement agencies and will have to be available over the Internet. The use of the public Internet is not a serious issue as a feature set is merely an array of numbers that are interpreted by the server. Standard mechanisms for authentication should still be in place to protect the database from attacks, e.g., denial of service. However, no unique security issues are raised by our design and, due to the nature of the database content, even a full-scale security breach will not yield any usable information.

Initial investigations of performance have confirmed that the processing of a forensic target will have to be distributed for it to be completed in a timely fashion. The dominant factor is the computation of image feature sets. Depending on source image size, this computation can take from a fraction of a second to a couple of minutes. In our work, images were scaled to not exceed 384×384 pixels, and the processing time was about 0.5 seconds per image. Since it would take about 14 hours to complete the feature extraction sequentially in a realistic scenario (100,000 images), we are attempting to integrate feature extraction into a distributed digital forensics infrastructure [13]. This infrastructure supports investigations on a cluster, providing vast improvements in performance over traditional "single investigator machine" approaches. The performance improvements arise from using multiple CPUs to tackle CPU-intensive operations and extensive caching to reduce disk I/O.

Another benefit of the distributed forensics infrastructure is that it could support case-specific searches on a target. Specifically, the system would build a reference database of all the images on the target and allow searches for images similar to the ones submitted interactively by an investigator, e.g., images containing a particular person or building.

6. Conclusions

This paper introduces a new approach for forensic investigations of visual images using content-based image retrieval (CBIR). The approach extracts an image "fingerprint" (feature set) and uses it to perform comparisons to find the best match among a set of images. It is necessary to store only the fingerprint (not the original image) to perform comparisons. The main advantage of the approach is that it allows the

construction of a reference database of fingerprints of contraband images. A secondary benefit is that it dramatically reduces the storage requirements for the reference database making it easier to achieve good performance at a reasonable cost.

Experiments indicate that CBIR techniques are well-suited for forensic purposes. In particular, the tests of robustness of query results for versions of the original images obtained through common transformations (e.g., resizing) are very promising.

Two main applications are proposed: a reference database for contraband images and case-specific image search tools. In the first case, law enforcement agencies will be able to collectively build and access the database to automatically search targets for known contraband images. In the second case, a database of all images found on a target is constructed and investigators can submit queries for images similar to specific images of interest. To accommodate these applications, a service-oriented architecture and a distributed forensic tool are proposed.

The main contribution of this work is that it presents a sound and practical approach to automating the forensic examination of images. Unlike other approaches, such as hashing, the image analysis approach is very stable in that it can locate not only the original image but also common variations of the image.

Acknowledgments

The research of Yixin Chen was supported in part by NASA EPSCoR Grant NASA/LEQSF(2004)-DART-12 and by the Research Institute for Children, Children's Hospital, New Orleans, Louisiana. The authors also wish to thank James Z. Wang and Jia Li for providing executable code of the SIMPLIcity System.

References

[1] C. Carson, S. Belongie, H. Greenspan and J. Malik, Blobworld: Image segmentation using expectation-maximization and its application to image querying, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24(8), pp. 1026-1038, 2002.

[2] I. Cox, M. Miller, T. Minka, T. Papathomas and P. Yianilos, The Bayesian image retrieval system PicHunter: Theory, implementation and psychophysical experiments, *IEEE Transactions on Image Processing*, vol. 9(1), pp. 20-37, 2000.

[3] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, Philadelphia, Pennsylvania, 1992.

[4] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, Efficient and effective querying by image content, *Journal of Intelligent Information Systems*, vol. 3(3-4), pp. 231-262, 1994.

[5] A. Gersho, Asymptotically optimum block quantization, *IEEE Transactions on Information Theory*, vol. 25(4), pp. 373-380, 1979.

[6] T. Gevers and A. Smeulders, PicToSeek: Combining color and shape invariant features for image retrieval, *IEEE Transactions on Image Processing*, vol. 9(1), pp. 102-119, 2000.

[7] A. Gupta and R. Jain, Visual information retrieval, *Communications of the ACM*, vol. 40(5), pp. 70-79, 1997.

[8] J. Li, J. Wang and G. Wiederhold, IRM: Integrated region matching for image retrieval, *Proceedings of the ACM International Conference on Multimedia*, pp. 147-156, 2000.

[9] W. Ma and B. Manjunath, NeTra: A toolbox for navigating large image databases, *Proceedings of the IEEE International Conference on Image Processing*, pp. 568-571, 1997.

[10] S. Mehrotra, Y. Rui, M. Ortega-Binderberger and T. Huang, Supporting content-based queries over images in MARS, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp. 632-633, 1997.

[11] V. Ogle and M. Stonebraker. Chabot: Retrieval from a relational database of images, *IEEE Computer*, vol. 28(9), pp. 40-48, 1995.

[12] A. Pentland, R. Picard and S. Sclaroff, Photobook: Content-based manipulation for image databases, *International Journal of Computer Vision*, vol. 18(3), pp. 233-254, 1996.

[13] V. Roussev and G. Richard III, Breaking the performance wall: The case for distributed digital forensics, *Proceedings of the Digital Forensics Research Workshop*, 2004.

[14] J. Smith and S. Chang, VisualSEEK: A fully automated content-based query system, *Proceedings of the ACM International Conference on Multimedia*, pp. 87-98, 1996.

[15] J. Wang, J. Li and G. Wiederhold, SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(9), pp. 947-963, 2001.

[16] J. Wang, G. Wiederhold, O. Firschein and X. Sha, Content-based image indexing and searching using Daubechies' wavelets, *International Journal on Digital Libraries*, vol. 1(4), pp. 311-328, 1998.