Content-Based Image Retrieval for Digital Forensics

Yixin Chen, Vassil Roussev, Golden G. Richard III, Yun Gao

Department of Computer Science
University of New Orleans
New Orleans, LA 70148

{yixin, vassil, golden, ygao}@cs.uno.edu

ABSTRACT

Digital forensic investigators are often faced with the task of manually examining a large number of (photographic) images in order to identify potential evidence. The task can be especially 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 in facilitating this process and are largely confined to generating pages of thumbnail images and identifying known files through cryptographic hashes.

We present a new approach that significantly automates the examination process by relying on image analysis techniques. The general approach 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, we build a feature set database that allows us to automatically scan a target machine for images that are similar to the ones in the database.

An important property of our approach is that it is not possible to recover the original image from the feature set. Therefore, it becomes possible to build a (potentially very large) database targeting known contraband images that investigators may be barred from collecting directly. The same approach can be used to automatically search for case-specific images, contraband or otherwise, or to provide online monitoring of shared storage for early detection of certain images.

In this paper, we motivate our work through several real-world scenarios, outline the mathematical foundations of the image analysis tools that we used, and describe the results of a set of comprehensive tests through which we validated the use of these tools for forensics purposes. We also discuss architectural and performance issues related to the implementation and practical use of a working system based on our prototype.

Keywords: digital forensics, image retrieval, distributed digital forensics

1. Introduction

Digital forensic investigations often require the examination of pictures found on the target media. Two typical tasks in that respect 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 ill-equipped to help the investigator given 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 lab. Note that this was a relatively old system with a very modest 6GB hard drive and the images were mostly in the browser's cache. Even if the investigator spends on average a fraction of a second on each image, it will still require several hours of routine, tedious work to browse through all of them. The dramatic drop in prices of storage devices, coupled with the leap in capacity (current street price for a 200GB hard drive is 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 the average home user will contain hundreds of thousands of images. If we consider a target such as a web hosting service that can have tens of millions of images, the problem of examining all images becomes virtually intractable and investigators will need some means to narrow down the search space.

The driving problem behind our 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 Lab (GCCFL). They have a clear and pressing need for a forensic tool that would allow 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 found 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 on a routine basis by numerous agencies would be a challenging task. From a technical point of view, the storage and bandwidth requirements would be staggering. Scalability would be difficult to achieve as replication and distribution of such highly sensitive material would have to be limited. Finally, a potential security breach at such a storage facility or misuse by authorized personnel can only be compared to a nuclear accident as far as the public outcry is concerned.

In summary, any realistic system design should not rely on having access to the original images during the lookup process. Rather, it would 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 be sufficient to allow a high-probability match but it should also be impossible to reconstitute any recognizable version of the original image. In our search for a solution, we have come to the conclusion that analytical methods for content-based image retrieval can go a long way towards addressing image analysis needs in digital forensics. This paper is a first effort to evaluate the suitability of this approach as well as to present an architectural framework that would allow the deployment of a working system.

The rest of the paper is organized as follows. First, we describe previous work in content-based image retrieval, which forms the basis of our own work. Next, we present a set of experimental results that validate the use of content-based image retrieval for digital forensics investigations. Finally, we outline an architectural design (currently under implementation) that will allow the building of a scalable system that can be deployed and used in the real world.

2. Content-Based Image Retrieval

2.1. 2.1 Overview

Depending on the query formats, image retrieval algorithms roughly belong to two categories: text-based approaches and content-based methods (see Figure 1). The 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, such as 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. Further, it may be difficult to describe image content with a small set of keywords. This motivates 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).

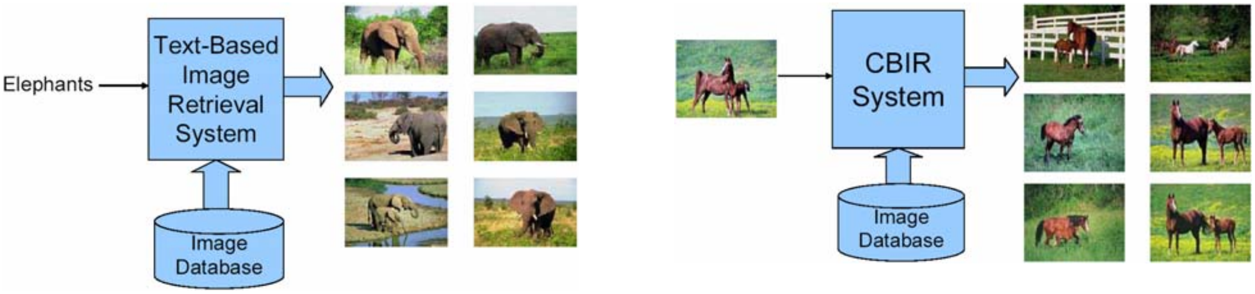

Figure 1: Scheme diagrams of a text-based image retrieval system (left) and a content-based image retrieval system (right).

In the past decade, many CBIR systems have been developed. Examples include the IBM QBIC System [FALO94], the MIT Photobook System [PENT96], the Berkeley Chabot

[OGLE95] and Blobworld Systems [CARS02], the Virage System [GUPT97], Columbia's VisualSEEK and WebSEEK Systems [SMIT96], the PicHunter System [COX00], UCSB's NeTra System [MA97], UIUC's MARS System [MEHR97], the PicToSeek System [GEVE00], and Stanford's WBIIS [WANG98] and SIMPLIcity Systems [WANG01], to name just a few.

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. These 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 CBIR. 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 such as IBM QBIC, MIT Photobook, Virage System, and Columbia VisualSEEK and WebSEEK. A major drawback of the global histogram search lies in 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 will review a CBIR system, SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) [WANG01], which will be tested for forensics purposes.

2.2. 2.2 SIMPLIcity System

In the SIMPLIcity system, the query image and all images in the database are first segmented into regions. To segment an image, the system first partitions the image into non-overlapping blocks of size 4x4. A feature vector is then extracted for each block. The block size is chosen to compromise between texture effectiveness and computation time. Smaller block sizes may preserve more texture details but increase the computation time. Conversely, increasing the block size can reduce the computation time but lose texture information and increase the segmentation coarseness.

Each feature vector consists of six features. Three of them are the average color components in a 4x4 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 [DAUB92], that is, 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 4x4 block is decomposed into four frequency bands: the LL (low low), LH (low high), HL, and HH bands. Each band contains 2x2 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 $f = \left(\frac{1}{4}\sum_{i=1}^{1}\sum_{j=0}^{1} c_{k+i,l+j}{}^{2}\right)^{\frac{1}{2}}$. 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. 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 with 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 adaptively select 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 in accordance with the adjustment of the stopping 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 are calculated for each region to describe shape properties. They are normalized inertia [GERS00] of order 1 to 3. 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 in the figure to make illustration easier. In the segmentation result, each region is represented by a distinct color.

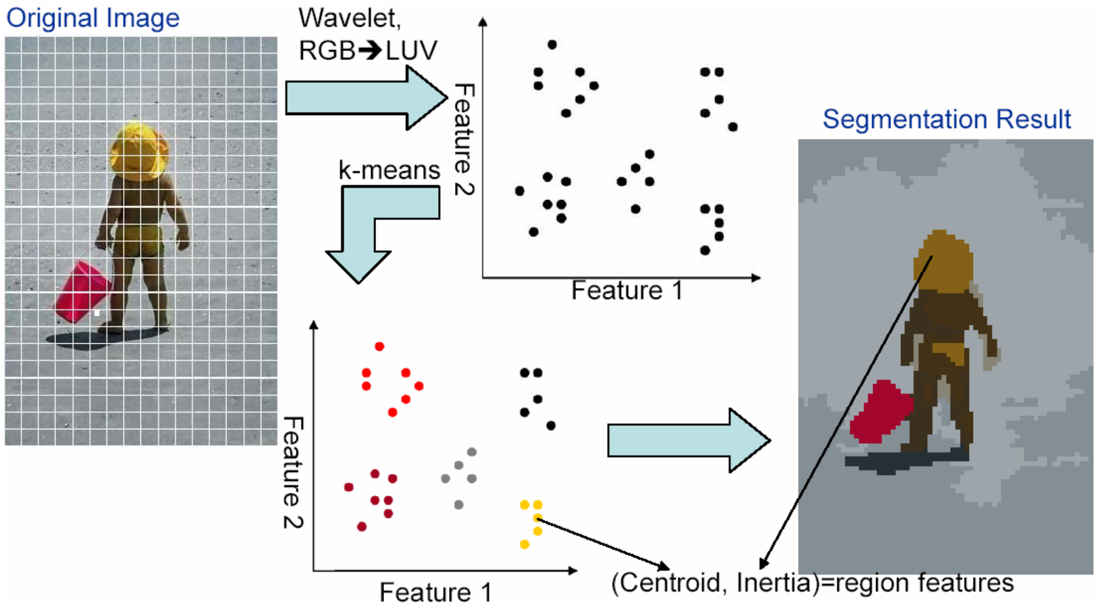

Figure 2: Scheme diagram of the feature extraction process.

The similarity between two images is computed according to an integrated region matching (IRM) scheme [LI00]. In order to reduce the influence of inaccurate segmentation, the IRM measure allows for matching a region of one image to several regions of another image. That is, 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 for forensic purposes, 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 percent, 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 numbers 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 the design of 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 90º rotations and mirroring (vertical and horizontal) of images that can be expected during the processing of the raw images.

The target database consists of 5,631 photo images in JPEG format. The goal is to demonstrate the ability of the system to recognize an image when its altered version is submitted as the query. We apply image alteration to an image (called target image *i*) in the database. The resulting image *i’* is then used as the query image and the rank of the retrieved target image *i* is recorded. Here 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, i.e., a lower value in rank. The lowest (or 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 for all target images (excluding the missed images) is computed and these results are given in Table 2.

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 90º.
Flip	Creating a mirror image.
Flop	Creating a mirror image.

Table 2. Experimental results for queries based on altered images. The rank is the position of the target image in the first 100 retrieved images.

Alteration ID	**The Number of 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

The experimental results clearly indicate that the image analysis techniques employed in our pilot study are an excellent match for the problems faced in digital forensic investigations and warrant further development. One of problems we face is that the original research upon which our work is based is directed at a slightly different problem and the research system we used is far from being directly applicable to forensic investigations. Specifically, system-level issues such as performance, scalability, and security need to be addressed before a working prototype can be tested in a forensics lab. In the following sections, we discuss our system design and ongoing implementation effort.

4. Design Overview

4.1. Goals

Our design rationale is based on the following goals:

- *Service-oriented architecture.* Despite the fact that the stored image feature sets are not contraband, even if they do 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 database. Furthermore, the database will be a highly dynamic entity once a large number of federal, state, and local agencies become contributors.
- *Performance.* The system should be able to handle individual requests at rates that will allow investigations to proceed interactively. For reference, current open source imaging software can generate thumbnail images at approximately 1000 images per minute—a working system should be able to perform at a similar rate or better (while providing a much higher value service).
- *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 here. Recall that we do not store copies of the actual images, which not only makes the system legal but it greatly mitigates the consequences of any security breach.
- *Flexible deployment.* It should be possible to use the same basic architecture for both forensics investigations and for preventive monitoring.

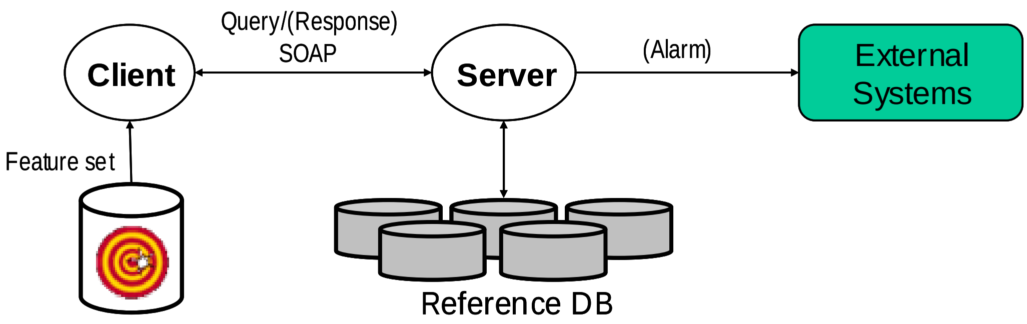

Figure 3: Architecture for a client/server image matching service. A client computes the feature set for one or more target images and issues queries against a server, which maintains a reference database of features for images of interest. The server indicates which images match the reference database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

4.2. Architecture

Architecturally, our design is based on a classical client/server approach, as shown in Figure 3. The function of the client is to take one or more target images, compute the feature sets, and submit them to the server. The server then compares the submitted feature sets with those in a reference database. The client must keep track of outstanding queries so that it can alert a user when image matches occur. Note that a match here 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 need to be dealt with by the investigator.

The server has two basic functions:

- 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.

- 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 the client. The rationale here 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 does not raise any new security concerns because a feature set is merely an array of numbers to be interpreted by the server. Standard mechanisms for authentication should still be in place to protect the database from attacks, e.g., denial of service attacks. 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 information that is usable outside the system.

Some initial performance measures during our experiments confirmed our hypothesis that the processing of a forensic target will have to be distributed in order to be completed in a timely (ideally, interactive) fashion. The dominant factor is computation of the feature set for an image. Depending on the size of the source image, this computation can take anywhere from a fraction of a second to a couple of minutes. In our work, we scaled all images so that they did not exceed 384x384 pixels. Thus, the processing time was about 0.5 seconds per image. Assuming a target of 100,000 images, it would take about 14 hours to complete the feature extraction sequentially. Therefore, we are working on an implementation that integrates the feature extraction function into our distributed digital forensics infrastructure [ROU04]. This infrastructure supports digital forensics investigations on a cluster, providing vast improvements in performance over traditional “single investigator machine” approaches. These performance improvements are based on using multiple CPUs to tackle CPU-intensive operations and extensive use of caching to reduce disk I/O.

Another benefit in using our distributed forensics infrastructure is to support case-specific searches on a target. In this case, the system would build a reference database of all the images on the target and allows searches for images similar to ones submitted interactively by an investigator, e.g., images containing somebody’s face or a particular building.

5. Conclusions

In this paper, we introduce a new approach to the forensic investigation of visual images through content-based image retrieval (CBIR). We use established CBIR techniques originally developed for other application domains and apply them for digital forensic purposes. Our approach is to extract an image ‘fingerprint’ (feature set) and use it to perform subsequent comparisons to find the best match among a set of images. A notable characteristic of this method is that it does not need to store the original image (only the fingerprint) in order to perform subsequent comparisons. The main advantage is that this allows the building 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 a lot easier to achieve good performance at a reasonable cost.

We performed a set of experiments to evaluate the suitability of the CBIR techniques used for forensic purposes. In particular, we tested the robustness of the query results by searching the reference database for *versions* of the original images obtained through common transformations, such as resizing. Our results, based on a sample of 5,631 images, strongly support the suitability of the chosen techniques for forensic purposes. Specifically, we propose two main applications: 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. In the second case, a database of all images found on a target is built and the investigator can submit queries for images similar to some specific images they are interested in. To accommodate the two scenarios, we presented a design based on a service-oriented architecture (currently under implementation) and a distributed forensic tool based on our existing work.

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

6. Bibliography

[CARS02] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 8, pp. 1026-1038, 2002.

[COX00] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, “The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments,” *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 20-37, 2000.

[DAUB92] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[FALO94] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, “Effcient and Effective Querying by Image Content,” *J. Intell. Inform. Syst.*, vol. 3, no. 3-4, pp. 231-262, 1994.

[GERS00] A. Gersho, “Asymptotically Optimum Block Quantization,” *IEEE Trans. Information Theory*, vol. 25, no. 4, pp. 373-380, 1979.

[GEVE00] T. Gevers and A. W. M. Smeulders, “PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval,” *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 102-119, 2000.

[GUPT97] A. Gupta and R. Jain, “Visual Information Retrieval,” *Commun. ACM*, vol. 40, no. 5, pp. 70-79, 1997.

[LI00] J. Li, J. Z. Wang, and G. Wiederhold, “IRM: Integrated Region Matching for Image Retrieval,” *Proc. 8th ACM Int'l Conf. on Multimedia*, pp. 147-156, 2000.

[MA97] W. Y. Ma and B. Manjunath, “NeTra: A Toolbox for Navigating Large Image Databases,” *Proc. IEEE Int'l Conf. Image Processing*, pp. 568-571, 1997.

[MEHR97] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, and T. S. Huang, “Supporting Content-Based Queries over Images in MARS,” *Proc. IEEE Int'l Conf. on Multimedia Computing and Systems*, pp. 632-633, 1997.

[OGLE95] V. Ogle and M. Stonebraker, “Chabot: Retrieval from a Relational Database of Images,” *IEEE Computer*, vol. 28, no. 9, pp. 40-48, 1995.

[PENT96] A. Pentland, R. W. Picard, and S. Sclaroff, “Photobook: Content-Based Manipulation for Image Databases,” *Int'l J. Comput. Vis.*, vol. 18, no. 3, pp. 233-254, 1996.

[ROU04] V. Roussev, G. G. Richard III, "Breaking the Performance Wall: The Case for Distributed Digital Forensics," *Proceedings of the 2004 Digital Forensics Research Workshop (DFRWS 2004)*, Baltimore, MD.

[SMIT96] J. R. Smith and S.-F. Chang, “VisualSEEK: A Fully Automated Content-Based Query System,” *Proc. 4th ACM Int'l Conf. on Multimedia*, pp. 87-98, 1996.

[WANG01] J. Z. Wang, J. Li, and G. Wiederhold, “SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 9, pp. 947-963, 2001.

[WANG98] J. Z. Wang, G. Wiederhold, O. Firschein, and X. W. Sha, “Content-Based Image Indexing and Searching Using Daubechies’ Wavelets,” *Int'l J. Digital Libraries*, vol. 1, no. 4, pp. 311-328, 1998.