

Image Processing for Internet Applications

Ping Wah Wong
Hewlett Packard Company
11000 Wolfe Road, M/S 42UO
Cupertino, CA 95014
e-mail: pwong@cup.hp.com

Abstract

In recent years, we have observed an explosion in the growth of the internet. One major advantage of the world wide web is that it provides a platform for people to easily share documents, images, and other forms of contents. We focus in this paper on imaging applications for the internet. In particular, we will describe technologies that are designed for internet imaging applications. These include systems and algorithms for the efficient transmission and sharing of images over the network. Important image processing problems such as multiresolution imaging, remote printing, image security and watermarking are also considered.

1 Introduction

The growth of the internet in recent years has spawned many applications that use high quality images and other multimedia contents. Applications include online catalogs, online maps, image database, image trading, news reporting, online magazines, online fine art, dissemination of engineering drawings and documents, personal publishing, photo sharing, and many others. This has led to an increase in the level of research efforts as well as in the implementation of practical systems in the internet imaging area.

If we examine the overall system for delivering images over the internet, there are many important factors that need to be considered. Efficient transmission of contents over the internet is very important because many users access the internet through a limited bandwidth channel (modem). End-to-end image quality is very important both from the content providers' and from the users' point of views. In many cases, digital images are eventually going to be printed. This brings up the formulation of a remote printing problem, where one can jointly consider the optimization of several components to maximize performance in internet based remote printing scenarios. As images are made available over the internet for down-loading, there is a great deal

of concern over image security and copyrighting issues. As a result, technologies that can be used for enforcing copyright are both timely and important. There are also overall system oriented issues that plays a very important role in practical internet imaging systems. These include platform and browser compatibilities, reliability in printing, and interactivity with images such as panning and zooming.

In this paper, we consider an internet imaging system that is based on the FlashPix image file format [1] and the Internet Imaging Protocol (IIP) [2]. FlashPix is a redundant multiresolution image file format using tiles at each resolution level, that is designed for fast image data access. IIP is a very simple protocol that is designed for the efficient transmission of images using the tile structure of FlashPix, taking into consideration bandwidth and resolution trade-offs. We also consider in some detail a remote printing problem, and examine several approaches including the optimization of JPEG quantization tables for printing, and generation of embedded halftone images. Finally we consider image security issues, and describe some watermarking solutions that are useful for internet imaging applications.

2 Image Transmission

Consider the problem of placing an image on the world wide web that is to be viewed through client browsers over the internet. Most existing browsers support two native image formats—GIF and JPEG. Both are single resolution image formats. In GIF, palletized image data are losslessly compressed using a flavor of Lempel-Ziv compression algorithm [3, 4]. JPEG is a lossy compression algorithm based on an international standard [5].

Suppose a user wants to view an image through a browser. In one case we have an image of size 2000 by 3000 pixels at a depth of 24 bits per pixel. Such an image size is commonly considered to be roughly equivalent to the resolution of a frame of 35 mm film. In another case we have an image at the VGA resolution with 640 by 480 pixels at 16 bits per pixel. Consider as a

rough estimate that we are compressing the images at a compression ratio of 10:1, and that a user connects to the internet using a modem of 33.5 Kbaud per second. Under these conditions, the duration it takes to transmit the 2000 by 3000 image and the 640 by 480 image are 10 minutes and 20 seconds, respectively. It is perhaps agreeable that 10 minutes of transmission time for an image is unacceptable for many users, while 20 seconds of transmission time is acceptable. Hence transmission time alone suggests that we favor using low resolution images on the web. If we consider printing, however, then it is also known that hardcopies that are generated from the 640 by 480 image, even at a postcard size, is generally not of good enough photographic quality. Hence print quality consideration suggests that we favor using high resolution images.

To answer the question regarding an appropriate image size that we can use for putting images over the world wide web, we need to know the application and the purposes that we use the images for. It is clear from the aforementioned example that if we use single resolution images, then we either have to tolerate a long transmission time or accept suboptimal print quality.

This line of reasoning leads us to consider a multiresolution approach where the appropriate image data can be transmitted to the destination depending on the need of the user. For example, if a user requests an image to be viewed on screen, then we transmit image data appropriate for the screen resolution of about 72 pixels per inch. On the other hand, if a user wants to generate high quality hardcopy, then we want to transmit data at the print resolution. As a rule of thumb, a continuous tone image resolution of 150 pixels per inch¹ is generally considered to be sufficient for generating very high quality photographic hardcopies.

There are several multiresolution imaging approaches that can be considered. Among them are Gaussian pyramid [6], Laplacian pyramid [6], and wavelet decomposition [7]. Each of these is a multiresolution imaging structure that satisfies a perfect reconstruction property. The main differences between them are in the varying degrees of storage and computational requirements for accessing and extracting image data of various resolutions. In particular, the Gaussian pyramid is a redundant image format where each multiresolution layer is generated from the next higher resolution layer by filtering and down-sampling. The various multiresolution layers are collected to form a pyramid as shown in Fig. 1.

We have chosen the Gaussian pyramid because of its high access speed; there is no computation involved in extracting the image data from any resolution level.

¹It is important to note that we are considering resolution for continuous tone images here. This is not the same as printer or print resolution.

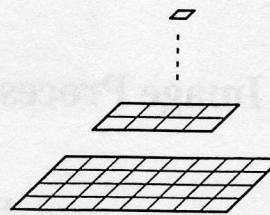


Figure 1: A Gaussian pyramid suggested by Burt and Adelson [6]. Each lower resolution layers are obtained from the next higher resolution layer by low pass filtering and down-sampling. Each multiresolution image layer is partitioned into tiles of size 64 by 64 in the FlashPix image file format [1].

This is an extremely important consideration for image servers because a server is expected to receive many concurrent "hits" (requests for data from clients) and hence it needs to be very efficient in serving image data to each client. To further facilitate high speed image data access, we partition each resolution level into tiles and compress each tile independently. Depending on the choice of the system administrator, tiles can be served in JPEG compressed or uncompressed form. In such a way, we can access and transmit a portion of the pixels within a certain resolution level by getting to the desired tiles at a very high speed. This forms the basic components of a redundant multiresolution image file format called FlashPix [1].

Using the multiresolution organization and the tile structure within FlashPix, we have built a very simple protocol called Internet Imaging Protocol (IIP) [2] so that an imaging server and client can communicate with each other for transmitting tile based multiresolution image data very efficiently. In such a way, a client can request tile based data from the server depending on its need. Furthermore, such a multiresolution structure also enables new imaging capabilities such as panning and zooming. For example, as one zooms into various areas within a FlashPix image, then the data from the higher resolution levels can be transmitted from the server to the client. Hence, it enables zooming in (up to the resolution limit of the source image) without loss of resolution.

3 Image Processing and Remote Printing

It is important to consider the end-to-end image quality throughout such a networked and distributed imaging system. The problem of maintaining a superior end-to-end image quality is a complicated matter because of a number of reasons including the variety of the sys-

tem components involved, the variation in the condition where the image was captured, and the different characteristics of the display or printing devices. To this end, we have considered many image processing algorithms and their interactions with each other where the emphasis is on building an automatic imaging pipeline. By an automatic imaging pipeline we mean that the image processing steps such as contrast enhancement, artifact removal and image sharpening should be self adjusting. Taking contrast enhancement as an example, we want an adaptive algorithm [8] that can determine the extent of adjustment from the characteristics of an image, and then adjusts the image accordingly to improve the contrast. One important requirement is that if this adaptive algorithm is applied to an image that is already of good contrast, the algorithm should not worsen the quality of the image. We considered and reported an imaging pipeline for a digital photography application [9]. Similar pipeline with some modifications can be applied in a general internet imaging system.

A very interesting problem in internet imaging is in the remote printing of images. In both local and remote image printing, it is necessary to perform image scaling and halftoning for an image to be printed properly. In the internet environment where images are to be transmitted to a remote location, we frequently transmit the image data in a compressed format for communication efficiency considerations. Our motivation in the remote printing problem is to examine the interplay among compression, scaling and halftoning so that we can achieve better performance compared to applying each of these steps in isolation of the others.

Fig. 2 shows two possible imaging pipelines suitable for remote printing. In Fig. 2 (a), the continuous tone image is encoded and the compressed bit stream is sent to the remote location. Popular image compression devices include JPEG [5] and wavelet coders [10, 11]. At the client side, the bit stream is decoded, and finally the image is scaled and halftoned before sending to the printer. Fig. 2 (b) shows a procedure similar to that in facsimile. In this case, the image is first halftoned and then compressed. The resulting bit stream is sent to the client where it will be decoded and printed. We will consider in the following two problems that are suitable for the pipelines in Fig. 2 (a) and (b), respectively.

3.1 Optimized Compression for Printing

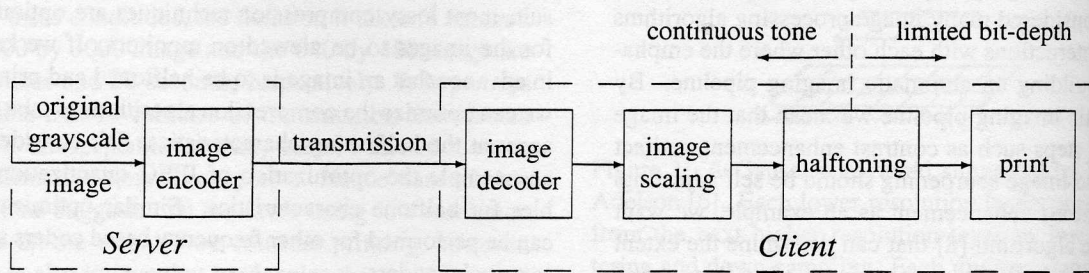
Consider the remote printing pipeline of Fig. 2 (a) where the continuous tone image is compressed before it is transmitted to the client. At the receiving end, the bit stream is decoded, and then the reconstructed image is halftoned during the printing process. Frequently, lossy image compression algorithms are designed to minimize

a visual distortion criterion. The visual distortion is usually based on a human visual model such as Mannos and Sakrison's contrast sensitivity function [12]. As a result, most lossy compression techniques are optimized for the images to be viewed on monitors. If we know in advance that an image is to be halftoned and printed, we can optimize the compression algorithms to take into account the halftoning characteristics. We consider as an example the optimization of JPEG quantization tables for halftone characteristics. Similar optimization can be performed for other frequency based coders such as wavelet coders.

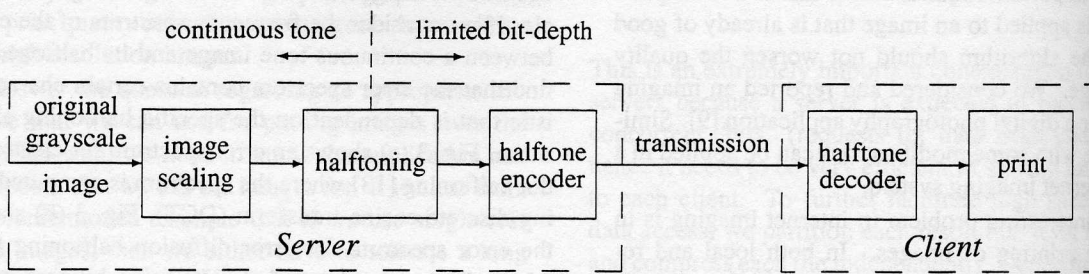
A halftoning algorithm converts a continuous tone image into an image with limited bit-depth, e.g., two levels. If we consider the frequency spectrum of the error between a continuous tone image and its halftone, we find that the error spectrum contains certain characteristic that is dependent on the specific halftoning algorithm. Fig. 3 (a) shows an error spectrum for clustered-dot halftoning [13] where the spectrum is computed using discrete cosine transform (DCT). Fig. 3 (b) shows the error spectrum for error diffusion halftoning [14] where the image was scaled (2X) using bilinear interpolation. We can see that halftoning algorithms cast a certain "signature" in the frequency spectra of images. The important observation is that we should optimize compression algorithms by applying a relatively low bit rate to those frequency components where the halftone error spectrum is strong, and a relatively high bit rate to those frequency components where the halftone error spectrum is weak. The intuition is that we do not want to apply high bit rate to those frequency components that will be corrupted by strong halftone noise. Details of such a scheme for JPEG compression is reported in [15, 16]. Using such an optimized JPEG quantization table, we found that we can typically improve the compression performance by about 30 to 50%.

3.2 Embedded Halftoning

Now we consider a scheme that will be applicable to the printing pipeline in Fig. 2 (b). Here an image is halftoned before it is transmitted to the destination. Now suppose that we are in an internet environment where we do not necessarily know how the image is to be used at the destination. For example, the client may decide to view the image with a color display, or view with a limited bit-depth monochrome display, or the client may decide to print with either a 300 dots per inch (dpi) or a 600 dpi printer. Under this situation, it is advantageous to generate a multi-tone image that will be suitable for several purposes. For example, we like to generate a 16-tone image for a color display with a palette size of 16, where a binary monochrome image is embedded so that we can

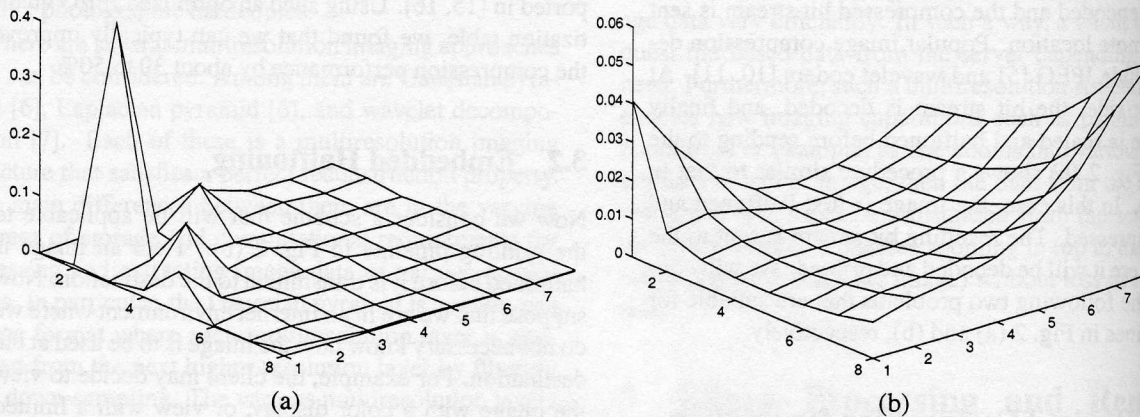


(a)



(b)

Figure 2: Two possible pipelines for remote printing systems. The difference between the two pipelines is in whether compression is performed on the continuous tone images or on the halftoned images.



(a)

(b)

Figure 3: Combined halftoning and scaling error profiles shown in the DCT domain, showing the signature of halftoning algorithms. Part (a) shows the DCT error spectrum of clustered-dot halftoning with no image scaling, while part (b) shows the DCT error spectrum for error diffusion halftoning with image scaling (bilinear interpolation, 2X).

extract the appropriate bits (1 bit per pixel) from the 16-tone image for printing purposes. The key is that we want the two tone monochrome image to be extractable from the 16-tone color image without any further computation. We call this an embedding property, which can be done in size [17], or in the bit-depth of the output [18]. We will consider each one in the following.

Suppose we want to print a continuous tone image of size M by N at a remote printer. Further suppose that there are two printers at the remote location with resolutions R dots per inch (dpi) and R/K dpi, where K is assumed to be a rational number. If we know exactly the printer that the image will be printed on, then we can generate a halftone appropriate for the specified printer. If we do not know in advance which printer the user will choose, however, then we can generate a halftone of size, say, M by N pixels for the printer at R dpi, where this halftone will have the property that if it is down-sampled by a factor of K to give another halftone of size M/K by N/K pixels, the resulting smaller size halftone is a high quality representation of the original grayscale image at the resolution R/K dpi. This halftone is said to have an *embedding in size* property. Here we can interpret K as the scaling factor between the two halftone images. A detailed formulation of this problem can be found in [17]. A non-iterative algorithm for generating halftones with embedding in size property is also described in [17]. Fig. 4 shows three halftone images of successively smaller size, where the smaller size halftones can be obtained from the larger size halftones by simple down-sampling.

A second problem in embedded halftoning is *embedding in bit-depth*. Suppose we have two image display or printing devices of different characteristics. For example, one may be a color display that can handle 16 different colors, and another one may be a monochrome printer that can accept only binary (either black or white) image data. To this end, we want to design a multi-toning algorithm that can generate a 16 color image, where we can extract the binary halftone for the monochrome printer by, for example, taking one bit out of the 4 bits meant for the 16 color display. This problem can be solved using a set partitioning algorithm in vector quantization in conjunction with an error diffusion algorithm. Details and examples can be found in [18].

It is certainly possible to combine the two embedded halftoning approach to generate a multi-toned image with both the embedding in size and bit-depth properties. This will provide a very flexible solution for the remote printing problem over the internet.

4 Image Security

The proliferation of the world wide web has allowed digital contents to be shared very easily. Because of the ease of access to these contents, serious questions arise regarding security and copyrighting issues. There are two general mechanisms that one can consider when it comes to image security over the internet. They are access control and watermarking.

There are several ways to achieve access control in an internet based image server. One approach is through the use of cryptographic methods [19]. For example, if the administrator of an image server decides that all the images are to be transmitted in encrypted form, then all the clients and users will need a decryption key in order to be able to view the image. Here the decryption key (could be different for different users) serves as the element of access control. On a system level, we can implement a password scheme to authenticate users such as those used in a shared access computer systems. In such case, only authorized user who has an account with a certain access privilege can gain access to the system. This is different from cryptographic control of images in the first case because it actually blocks access to the entire system, while in the first case we only block access to images on the system.

With an image server that supports transmission of tiles within a multiresolution structure as in FlashPix and IIP, we can implement a scheme that provides resolution based access control. Consider as an example an application where digital images are to be sold through the web. We can allow free and unlimited access to certain low resolution levels of images, while the access of the high resolution levels requires special authorizations. For instance, authorization can be granted base on an agreed payment from the user to the server owner. We have considered a secure printing system [20] that allows images and documents to be sold through the world wide web on a pay-per-print model. We use resolution as well as visible watermarking (to be discussed later) to provide a preview mechanism so that users can choose the documents and images that they want to purchase.

Another area of image security is in the copyrighting of digital contents. Recently there has been a great deal of research activities in digital watermarking that address various aspects of copyrighting. There are many applications of watermarking, and various watermarking schemes are designed to satisfy these purposes [21]. In the following we will briefly describe the applications of several categories of watermarking schemes.

One important application of watermarking is in ownership assertion. In this case, an owner of an image inserts a watermark into an image and then he can make this image available on the network. If another person

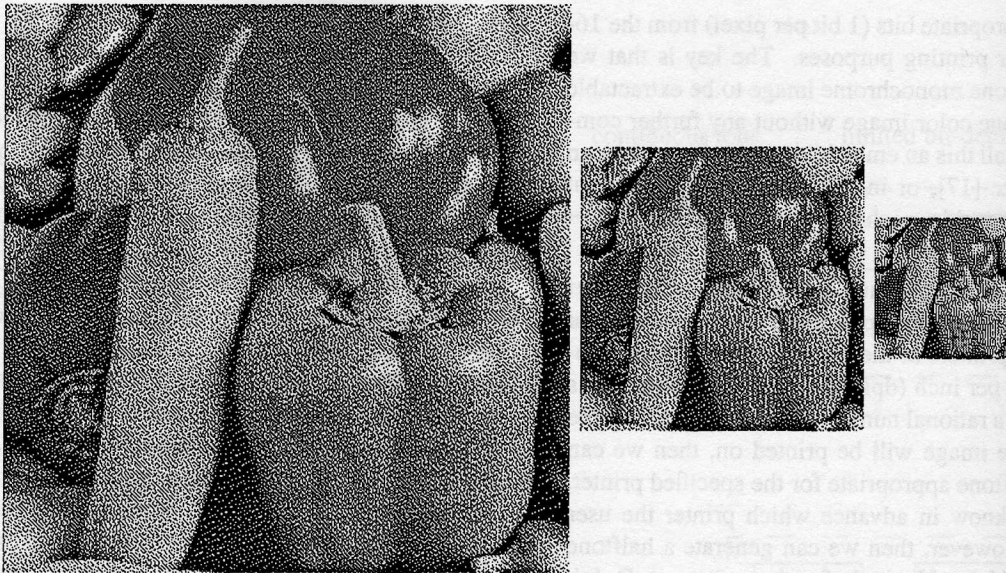


Figure 4: An example of embedded halftoning where a smaller size halftone can be obtained from a larger size halftone by simple down-sampling.

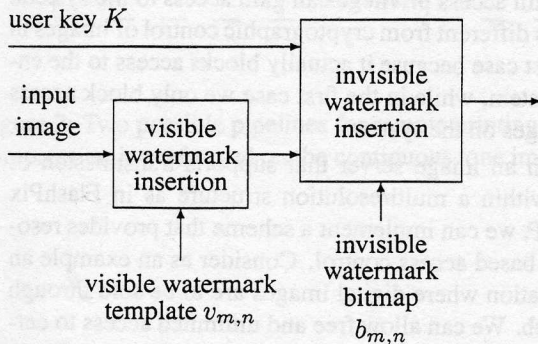


Figure 5: A method in using an invisible authentication watermark to protect a visible watermark.

uses the same image or a derived image, we need a way to show that the image actually belongs to the original owner. This can be done through the detection of a watermark, provided the watermark is robust [22, 23] so that it can survive the common image processing operations to images.

Another important application of watermarking is in image verification. In this case, the objective is to be able to detect any change that is made to an image. This has obvious applications in legal usages, news reporting, medical archiving, and many others. The objective can be achieved by inserting a verification watermark [24, 25] to an image. When verification is required, then we can extract the watermark (in most cases requiring a user key) from the image so that any change in the image is reflected and localized by the integrity of the watermark.

Finally, visible watermark is important in some internet applications. For example, if we offer an image

for sale, we can provide a sample that is visibly watermarked. Upon receiving an appropriate payment from the customer, the owner can either transmit an unmarked image, or provide a mechanism for the customer to remove the visible watermark. One important issue in visible watermarking is to prevent an unauthorized person B to insert a visible logo of A and claim that the image was watermarked by A . Since it is easy for B to insert a logo of A into an image, this is a serious problem. We suggest a way to solve this problem [25] using an invisible verification watermark to protect the visible watermark. A block diagram of the scheme is shown in Fig. 5. In this case, the visible watermark is tied to the invisible watermark. Since only the owner of the image A would have the appropriate key to insert the invisible verification watermark of A , any other person B will not be able to falsify ownership of an image by simply inserting a logo of A .

5 Conclusions

We have discussed in this paper some image processing and system issues on communicating with images over the world wide web. We considered FlashPix and Internet Imaging Protocol that allow tile based multiresolution image data to be efficiently transmitted over the internet. In particular, this scheme enables image data of suitable resolution to be transmitted from the server to the clients based on client need. Not only does it use bandwidth effectively, it also provides capabilities such as zooming and panning that enhances the inter-

active experience of the users. We have also considered some interesting problems for remote printing applications, that includes optimization of compression for halftones, and the idea of embedded halftoning. Finally, we discussed image security and watermarking for addressing copyrighting issues arising from putting digital content publicly over the internet. Watermarking is currently still a relatively young technology, and there are plenty of opportunities for further research in the basic algorithms as well as in innovative applications.

The usage of the world wide web will undoubtedly grow even further in the foreseeable future. Images are commonly recognized to be a very effective component for communication. Consequently, we feel that internet imaging will undoubtedly remain to be a very active area for research and practical development. There are many problems such as watermarking and effective transmission protocols that will not only be of great intellectual interest, but also of great value in helping solve practical problems.

References

- [1] "Flashpix format specification, version 1.0." Eastman Kodak Company, Hewlett Packard Company, Live Picture, Inc., and Microsoft Corporation, September 1996.
- [2] "Internet imaging protocol specification, version 1.0.3." Hewlett Packard Company, Live Picture, Inc., and Eastman Kodak Company, August 1997.
- [3] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 337-343, May 1977.
- [4] T. A. Welch, "A technique for high-performance data compression," *IEEE Computer*, pp. 8-19, June 1984.
- [5] ITU-T Rec. T.81 — ISO/IEC No. 10918-1, "Information technology — Digital compression and coding of continuous-tone still images, Part I: Requirements and guidelines," 1993.
- [6] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. COM-31, pp. 532-540, April 1983.
- [7] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-11, pp. 674-693, July 1989.
- [8] D. Tretter, "System and method for histogram-based image contrast enhancement." United States Patent application, submitted in October 1995.
- [9] P. W. Wong, D. Tretter, C. Herley, N. Moayeri, and R. Lee, "Image processing considerations for digital cameras," in *Proceedings of IEEE CompCon*, (San Francisco, CA), pp. 280-285, February 1997.
- [10] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445-3462, December 1993.
- [11] P. W. Wong and S. Noyes, "Space-frequency localized image compression," *IEEE Transactions on Image Processing*, vol. 3, pp. 302-307, May 1994.
- [12] J. L. Mannos and D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 525-536, July 1974.
- [13] R. A. Ulichney, *Digital Halftoning*. Cambridge MA: MIT Press, 1987.
- [14] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *SID International Symposium, Digest of Technical Papers*, pp. 36-37, 1975.
- [15] R. A. Vander Kam and P. W. Wong, "Customized JPEG compression for grayscale printing," in *Proceedings of Data Compression Conference*, (Snowbird, UT), pp. 156-165, March 1994.
- [16] R. A. Vander Kam, P. W. Wong, and R. M. Gray, "JPEG-compliant perceptual coding for a grayscale image printing pipeline," *IEEE Transactions on Image Processing*. To appear.
- [17] P. W. Wong, "Adaptive error diffusion and its application in multiresolution rendering," *IEEE Transactions on Image Processing*, vol. 5, pp. 1184-1196, July 1996.
- [18] J. R. Goldschneider, E. A. Riskin, and P. W. Wong, "Embedded multilevel error diffusion," *IEEE Transactions on Image Processing*, vol. 6, pp. 956-964, July 1997.
- [19] B. Schneier, *Applied Cryptography*. New York, NY: John Wiley & Sons, 1995.
- [20] P. W. Wong, D. Tretter, T. Kite, Q. Lin, and H. Nguyen, "A web-based secure system for the distributed printing of document and images," *Journal of Visual Communication and Image Representation*. Submitted for publication.

- [21] N. Memon and P. W. Wong, "Protecting digital media content: Watermarks for copyrighting and authentication," *Communications of ACM*. To appear.
- [22] N. Nikolaidis and I. Pitas, "Copyright protection of images using robust digital signatures," in *Proceedings of ICASSP*, May 1996.
- [23] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," Tech. Rep. 95-10, NEC Research Institute, 1995.
- [24] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of ICIP*, (Santa Barbara, CA), October 1997.
- [25] P. W. Wong, "A watermark for image integrity and ownership verification," in *Proceedings of IS&T PIC Conference*, (Portland, OR), May 1998.