Working Paper Series
ISSN 1170-487X

Compression-based template matching

by Stuart Inglis & Ian H. Witten

Working Paper 93/8

October, 1993

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Compression-based Template Matching

Stuart Inglis
Computer Science, University of Waikato, Hamilton, New Zealand.
email singlis@waikato.ac.nz

Ian H. Witten
Computer Science, University of Waikato, Hamilton, New Zealand.
email ihw@waikato.ac.nz

1 Introduction

Textual image compression is a method of both lossy and lossless image compression that is particularly effective for images containing repeated sub-images, notably pages of text (Mohiuddin *et al.*, 1984; Witten *et al.*, 1992). The process comprises three main steps:

- Extracting all the characters from an image;
- Building a library that contains one representative for each character class;
- Compressing the image with respect to the library.

The first step is to extract the "marks," or sets of connected black pixels, from the image. Marks correspond approximately to individual characters, except that characters with disconnected parts like "i" and "j" are represented by pairs of marks, one for the body and the other for the dot. (Due to the adaptive compression process that follows, this incurs a negligible penalty.) To build the library in the second step, the marks are clustered into self-similar categories. The intention is that all marks that represent a particular character fall into the same class. This effect is achieved by checking each new mark against the library and adding it as a new symbol if it is sufficiently different from all previous clusters. The final step compresses the image by coding the library symbols and then coding the sequence of symbols that correspond to the marks on the page. Because many marks in the image map to any one character, the mark that is stored in the library is an average of the original marks in the category. The use of a single representative introduces a small amount of noise in the reconstructed image, and an extra step can be used to make the compression lossless if desired. The result is an effective lossless/lossy method for compressing textual images.

A critical requirement in the second step is the ability to match marks accurately. This is really a process of pattern comparison, and is often called "template matching." Template matching in textual image compression differs from the matching required in an optical character recognition (OCR) system because there is no need to identify the actual character to which any particular mark corresponds. Nevertheless, it is important to match each new occurrence of a mark consistently with the cluster to which it belongs, because compression will deteriorate if the matching operation introduces errors.

Matching procedures generally work by aligning the two images in the position where we think the best match will be made, and then examining an *error map*, which is the bitwise exclusive-OR of the images. The most effective methods employ heuristics that are tailored to the particular resolution of the input image. For example, the "combined size-independent strategy" defines measures that depend on the size and resolution of the marks and uses them to reject matches based on local criteria. For example, if any pixel in the error map has four black neighbors then the match is rejected.

This paper addresses the problem of pattern comparison by using an information or compression based approach. Following Mohiuddin *et al.* (1984), we use the amount of uncertainty or entropy between marks as the criterion for the matching process. Given a mark M consisting of binary pixels

M_{ij}, the conditional information in M given the library pattern L is defined as

$$I(M|L) \;=\; -\log_2 \Pr[M|L] \;=\; -\sum_i \sum_j \log_2 \Pr[M_{ij}|L]$$

The pattern M is accepted as a match to that class L which minimizes $I(M|L)$, so long as

$$I(M|L) < T_L,$$

where T_L is a pre-defined threshold for acceptance to pattern class L.

In order to employ this method of classification it is necessary to estimate the probabilities $\Pr[m_{ij}|L]$. The idea is to find, for each unknown mark M, the library image L_k that best models it—in the sense that the amount of information required to specify M given L_k is minimized. This distance metric is called the *cross-entropy* between pairs of images, and can be approximated by compressing one mark relative to another. The entropy model we use is the context-based compression model proposed by Langdon & Rissanen (1981) and further developed by Moffat (1991).

There are two principal issues to investigate when studying template matching methods: their susceptibility to different kinds of noise, and how they respond to errors in the initial registration. Because of the computation-intensive nature of the comparison operation, many schemes have been devised to pre-filter or *screen* the marks in advance to determine those that will surely fail the match. We present a novel method of screening which uses a quad-tree decomposition and finds local centroids at each tree level.

2 Background

Previous models for template matching are divided into two categories: local and global matching. Local methods reject or accept matches based on pixels in a small portion of the image, while global methods generally return a value representing the overall goodness of the match. Local and global matching methods fulfill two different requirements. In the former the match either succeeds or fails, while the latter methods compute a score that can be used to rank the comparisons from best to worst. Of course, local methods can revert to a global matching scheme if a mark is not rejected, and global methods can use a threshold to accept or reject matches.

The evolution of previous methods of pattern-matching is as follows. The "combined symbol matching" (CSM) method of Pratt *et al.* (1980) computes a weighted sum of pixels in the error map, weighting error pixels more highly if they occur in clusters. A match is rejected if this exceeds a specified threshold. The "weighted and-not" approach (WAN) of Holt and Xydeas (1986) improves on this by distinguishing black-to-white errors from white-to-black ones. The "pattern matching and substitution" (PMS) method of Johnsen *et al.* (1983) rejects a match if any position in the error map is found to have four or more neighbors that are set. In order to detect mismatches due to the presence of a thin stroke or gap in one image but not the other, another heuristic is used. In an attempt to normalize for size, the "combined size independent strategy" (CSIS) of Holt *et al.* (1988) uses two different criteria to detect thin strokes or gaps—one for characters that exceed certain preset width and height thresholds, and the other for the remaining ones.

Langdon and Rissanen introduced the idea of context-based compression of binary images, in which the probability of a pixel being black or white is estimated and used to encode the pixel via arithmetic coding. The probabilities are conditioned on a certain context of preceding pixels, and they suggested that a good choice is the 10-bit context shown in Figure 1(a). (The word "template" is generally used in the literature to describe the particular size and shape of context that is to be used for prediction; however we use "context" instead to avoid confusion with the templates that are being matched.) A black dot marks each pixel included in the context and a bullseye marks the position of the pixel about to be coded. The light gray pixels are ones whose values are not yet known by the decoder and so cannot be included

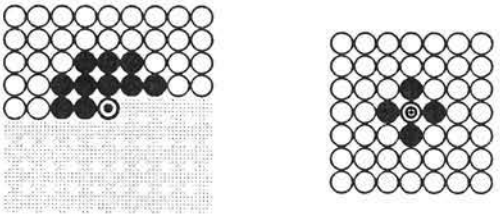
in the context, and the open circles indicate pixels whose values are known but are not included in the context.

For each different context, counts are recorded and used to predict black (c_b) and white (c_w) pixels in that context. The number of bits required to encode a particular black pixel is

$$-\log_2 \frac{c_b}{c_b + c_w} \text{ bits.}$$

Context-based image compression methods outperform well-known image compression methods for binary images such as CCITT Group 4.

Moffat (1991) pursued the idea further and investigated a number of variants. One was the notion of using a set of *surrounding* pixels as the context of the pixel being coded, rather than a set of *preceding* ones. An example is the template illustrated in Figure 1(b). This is of course a physically unrealizable configuration, but was introduced as a hypothetical "absolute best" compression scheme in order to estimate a bound on compressibility. We call these templates *clairvoyant*.



(a) Conventional (b) Clairvoyant

Figure 1: Two different types of templates

3 Compression-based template matching: method and results

Compression-based template matching has been implemented and tested against several other approaches to template matching. This section describes the implementation, presents results obtained with the new method, and compares it with the previous state of the art. Before any template matching method can be applied the patterns to be matched must be aligned together, and we first discuss some experiments to determine the best method of alignment. Next we explain exactly how cross-entropy is calculated and compare its performance on a set of test patterns with that of other methods. In order to investigate the effect of noise on classification accuracy, we applied artificial noise models to corrupt the test patterns, and the results of this are examined next. Finally we look at the question of screening, describe our method for screening using quad-tree decomposition, and report on the results obtained.

In order to test the template matching methods, a large collection of marks was required, along with their correct classification. These were obtained by extracting connected groups of pixels from several test images. The test images were digitized pages of a library catalog from Trinity College, Dublin. They had been scanned at 400 dpi with no noticeable skewing, and with careful attention paid to contrast settings so that there was little fragmentation of characters or bleeding of one into another.

The marks were were labeled by hand, and ones that could not be identified easily were discarded. This left a collection of around 2500 marks, each with a well-defined, unambiguous classification. Most were in roman font but some were italicized, and were labeled as such so that they were assigned to a completely different class from the corresponding roman letter. Six marks were ligatures formed by two characters bleeding together—for example, "rt" occurred twice—and they were labeled as such. There was some minor variation in type size—for example, letters in a small capitals face were assigned to the same class as regular capital letters. In total, 80 different classifications were used with an average of just over 30 marks in each class. Twelve classes had just one member, while six contained over a hundred members.

3.1 Registration

Before marks can be compared, they must be aligned so that each pixel on one can be mapped to the other: this is the process of *registration*. The result of any matching method depends critically on the registration, and in particular, for those matching methods that use the error map, it is clear that this map is very sensitive to the precise alignment of the images. There are several possible methods of registration. Some previously-reported work aligns the upper left corner of the bounding boxes, while other work uses the lower left corner instead. A third possibility is to align the geometric center of the marks' bounding boxes, while a fourth is to use the marks' centroids—that is, the average position of the black pixels they contain.

We undertook a series of experiments to determine the best relative position of registration. Each of the marks was overlaid with a 7×7 grid, and each of the 49 grid points was considered to be a possible reference point for registration. A 50th possibility, namely the mark's centroid (which of course varies from mark to mark), was also included. For any particular choice of reference point, every pair of marks was aligned on the corresponding intersection.

Each pair of marks that were the same according to the hand-assigned labels was registered on each of the 50 positions, and the two templates were matched. (In fact the CSIS template-matching method was used because of its speed.) The result in each case was a binary decision, "match" or "no match," and the correct result in each case was "match." There were 487,000 pairs of marks that had the same label, and repeating the operation over all these gave an overall correctness classification for each of the 50 reference points.

The resulting scores were largest for centroid alignment. In fact, for our data the centroid was, on average, very close to the geometric center. Registering based on the upper left and lower left corners gave classification accuracies that were both 6% lower than when the centroid was used.

3.2 Cross-entropy

To calculate the cross-entropy between two marks, the context-based coding model of Section 2 was used. As it is not necessary to be able to compress and decompress the mark, a clairvoyant context can be used to estimate the information content of each pixel. Various different contexts were tested, and the 5-pixel clairvoyant mask of Figure 1(b) was found to give best results. Larger masks may be more suitable for larger marks, or for greater digitization resolutions.

As was explained earlier, the coding probabilities are based on counts associated with each context. In compression it is normal to use an adaptive model, accumulating the counts as compression proceeds, because otherwise the model must somehow be communicated to the decoder. However, when obtaining an entropy figure there is no need to communicate the model and so a static one can be used. Like the clairvoyant context, this provides a lower bound on the amount of compression that could possibly be achieved in practice. Preliminary experiments indicated that the static model gives a slight performance increase over an adaptive model.

The compression-based template matching procedure begins by building the model of one mark *with respect to* the other. It would be much faster to calculate a single model and use it for all matches. However, this was found to degrade performance very seriously. Suppose the two marks to be matched are L and M. For any particular pixel $L_{i,j}$ in the first, there is a corresponding pixel $M_{i+\Delta x, j+\Delta y}$ in the second, where $(\Delta x, \Delta y)$ is the displacement due to registration. Figure 2 shows two different marks with the 5-pixel clairvoyant mask superimposed. The values of the five pixels under the mask in the first mark L, taken together, determine a particular context. The value of the central mask pixel in the other mark M gives a color, black or white, and the corresponding count for that context value is incremented. The process is repeated for every pixel, to build a complete model. In fact the marks are assumed to be surrounded by white space, and, once they have been registered, the maximum extent in both the horizontal and vertical directions is used as the area over which the entropy is calculated.

Once the model has been created, a second pass is taken over the marks and the counts in the models

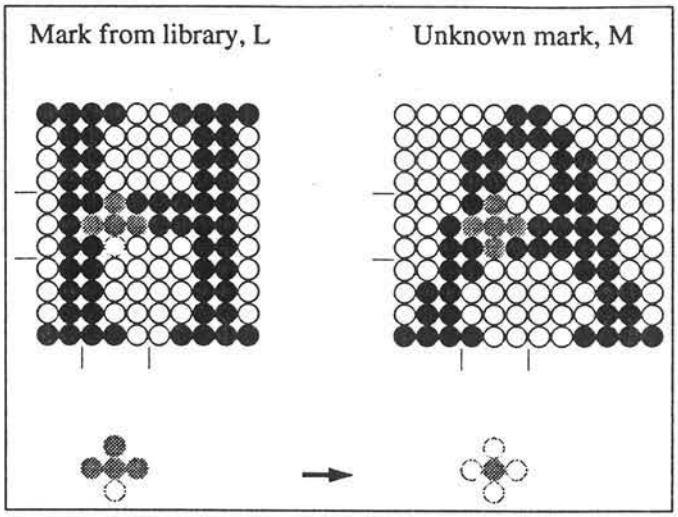


Figure 2: Using a five-pixel mask to build a model with respect to another mark

are used to calculate the entropy of the second mark M with respect to the first one L. The process is not symmetric, and both model and entropy will depend on whether the calculation is done for mark M with respect to L, or for L with respect to M. Both are calculated and the maximum is taken as the final cross-entropy measure.

In order to to cluster marks into self-similar categories, a threshold is used to decide whether a match should be accepted or rejected. After some experimentation it was decided to use a two-part criterion based on both the entropy per pixel and the absolute cross-entropy value. A match was rejected if either the number of bits per pixel exceeded B, or the total number of bits exceeded T, where B and T are constants that were determined experimentally. The values used were $B = 0.4$ and $T = 300$.

It would be interesting to consider setting an absolute threshold based on whether one mark could be coded more effectively with respect to the other, or whether it would be coded more effectively if the other were ignored. This would avoids the arbitrariness of numeric thresholds, and has yet to be explored.

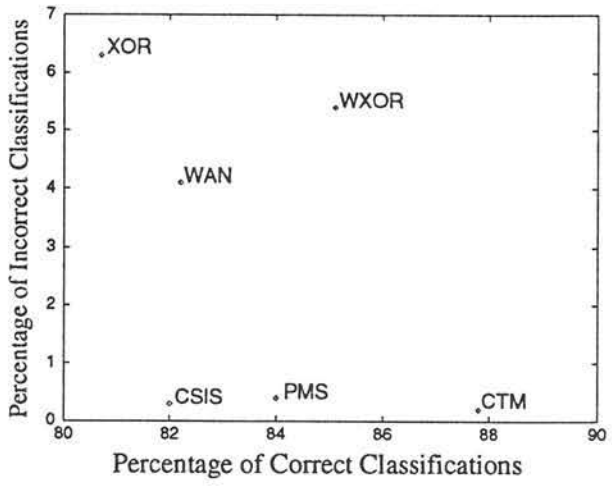


Figure 3: Results of each template matcher

3.3 Comparing CTM with other methods

The set of hand-labeled marks was used to compare the performance of compression-based template matching (CTM) with the other methods mentioned in Section 2. In order to test a particular template

matching method, each mark was compared with every other mark using that matching method. The result of every comparison is a binary decision: "match" or "no match." In order to produce overall figures, only positive identifications are considered. If the true classification of the two marks according to the hand-assigned labels is the same, and the outcome is "match," this is counted as a correct positive identification. If the true classification is different and the outcome is "match," this is counted as an incorrect positive identification. The percentage of correct and incorrect positive identifications is used as a measure of the matching method's performance.

All the template matching methods have parameters that need to be chosen. Each of the methods was assessed under the most favorable possible conditions. Parameters were determined by searching within the parameter space to maximize the number of correct minus the number of incorrect identifications. For example, for the WXOR method the range of the rejection threshold was varied to get the best result, which was in fact 80.7% correct and 6.3% incorrect identifications. In both the PMS and CSIS methods, each localized criterion for rejecting matches was treated as a parameter which was then subjected to optimization. For example, the number of times an error pixel has more than a specified weight became a variable parameter.

Figure 3 shows the results. Compression-based template matching clearly outperforms the other methods. It obtains the best results in both of the classification dimensions. The percentage of correct classifications exceeds that of the closest competitor, WXOR, and the number of incorrect classifications is lower than the lowest of the other methods, PMS and CSIS.

3.4 Noise deterioration

Each method of template matching responds differently to different kinds of noise. Three types of noise were added to the marks to emulate results under noisy conditions. The first was salt-and-pepper noise, in which the polarity of a certain percentage (10%) of pixels within a mark's bounding box was reversed. The second was the kind of edge noise caused by variations in paper quality or evenness of inking. This was modeled by adding black pixels around the perimeter of the mark, with a certain probability (again chosen to be 10%) that each white pixel around the perimeter would be changed to black. The third type of noise was a more extreme kind of edge noise, intended to emulate the blurring that occurs when images are scanned at too high a contrast, and was obtained by repeating the previously-described edge-noise process four times.

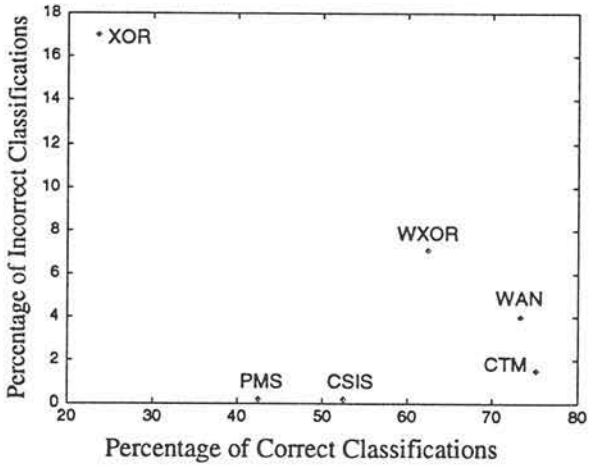


Figure 4: Average results with added noise

The testing process of Section 3.3 was repeated for each noise type. Every mark was corrupted as described above, and the entire process of comparison was re-done (using the same parameters as before).

Method	Random Noise		Edge Noise		High Edge Noise	
	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
XOR	24.4%	22.4%	38.2%	22.2%	8.3%	6.4%
WXOR	70.3%	7.3%	74.2%	5.7%	42.7%	8.4%
PMS	51.6%	0.3%	65.3%	0.3%	13.7%	0.1%
WAN	74.2%	4.7%	74.5%	4.1%	71.2%	3.3%
CSIS	64.3%	0.3%	68.7%	0.2%	24.6%	0.1%
CTM	74.9%	0.7%	76.3%	0.6%	74.1%	3.2%

Table 1: Susceptibility to different types of noise

Screening Method	Feature Space				Progressive centroids	
	Area	Perimeter	Horiz/Vertical Runs	Width/Height	Hamming	Euclidean
Screened matches	61.1%	67.3%	83.1%	84.3%	92.7%	93.4%

Table 2: Best screening percentages without affecting the classification

The results are shown in Table 1. Compression-based template matching (CTM) continues to achieve the highest number of correct matches under all conditions. However, it no longer achieves the lowest number of incorrect matches. The PMS and CSIS methods are consistently better at rejecting incorrect matches. However, their their performance with respect to correct matches is very much worse, reducing to only 25% for high edge noise, and CTM still seems to be the best choice overall.

Figure 4 shows the result of averaging the different kinds of artificial noise. CTM continues to classify more marks correctly than the other methods, and while the rate of incorrect classification is slightly higher than both PMS and CSIS, the latter methods make major sacrifices in correct classifications. The method with the closest number of correct classifications is the WAN method, but this has almost twice as many incorrect results.

3.5 Screening

Because of the high cost of the matching process, efficient methods for screening were investigated. Screening reduces the number of template matching operations that are undertaken, while simultaneously minimizing the number of matchable marks that are excluded. Many methods of screening have been investigated; including using features of the marks such as width, height, area, perimeter length, and number of internal black/white runs.

A novel approach to the screening problem is to register marks according to their centroids, and then divide them into quadrants defined by this position. At this stage a local centroid is calculated for each of the four quadrants, and for each of the four local centroids the distance between its position in the two marks is determined. Finally, these four distances are averaged.

The process could be continued recursively, defining a distance at level 0 based on the original centroids (which is 0, since centroid registration is used), a distance at level 1 as described above, and further higher-level distances. However, excellent screening results were obtained with the level-1 distance and further elaboration seemed to be unnecessary.

Each of the screening methods mentioned above was used as a pre-filtering step to the compression-based template matcher. They work as follows. The screening method determines the value of a numeric parameter. If the parameters for the two marks differ by more than a predetermined threshold the match is rejected without further consideration, while if it differs by less, a full template matching operation is undertaken.

For each method an appropriate threshold was determined in a preliminary series of experiments. It was chosen to be as selective as it possibly could be without affecting the outcome of the classification process. For example, using area as a screening function, the best cut-off value for the threshold eliminated 61.1% of the matches from further consideration.

Results for each of the methods is shown in Table 2. Area was the least effective method, followed

Method	Library Size	Compression ratio Lossless	Lossy
XOR	140	34.9	87.1
WXOR	135	35.8	91.1
PMS	180	35.9	79.3
WAN	126	31.6	76.8
CSIS	200	37.6	77.3
CTM	155	37.8	88.6

Table 3: Library size and compression ratios for TIC

by perimeter. The simplest method—width and height of the mark—fared much better, as did the number of horizontal and vertical runs. The new progressive centroid method gave by far the best performance, achieving 93.4% screening. A faster version using Hamming instead of Euclidean distance performed only slightly worse.

4 Using CTM in a textual image compression system

Compression based template matching is a good choice for use in a textual image compression system. Not only does it give better results for the classification process, as detailed in Section 3, but also each library mark can be chosen to be the one that gives the greatest compression.

We have incorporated the CTM scheme in a textual image compression system and performed experiments on CCITT image 1 (as this image contains mainly type-written text). The results, shown in Table 3, show a very slight improvement in lossless compression ratio over all other methods. However, the real advantage of the new scheme is in lossy compression. Here the quality of template matching, as well as the amount of compression obtained, is of paramount importance, because matching errors produce errors in the reproduced document. The new method improves the lossy compression ratio by almost 15% over CSIS. Although other systems such as XOR and WXOR have high lossy compression figures, it is evident from Figure 3 that their accuracy is much lower.

5 Summary

Compression-based template matching uses as its basis the amount of information that one pattern contributes to the encoding of the other. This is measured by employing a context-based compression method to approximate the cross entropy between images.

The resulting template matching procedure improves significantly on previous methods. Using it as a unsupervised classifier achieved 87.8% accuracy with 0.2% incorrect classifications, which is better in both dimensions than other methods. Combining this with the progressive screening of marks yields a method that is more accurate and involves less template comparisons than earlier methods. As an element of a textual image compression system, the matcher minimizes the number of bits required to encode a mark and achieves a slight overall gain in the lossless figures, as well as a 14% improvement in the lossy compression ratio.

References

Ascher, R.N. and Nagy, G. (1974) "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans Computers* C-23(11): 1174–1179; November.

Bell, T.C., Cleary, J.G. and Witten, I.H. (1990) *Text Compression*. Prentice Hall, Englewood Cliffs, NJ.

Holt, M.J.J. and Xydeas, C.S. (1986) 'Recent developments in image data compression for digital facsimile,' *ICL Technical Journal*: 123-146; May.

Holt, M.J. (1988) 'A fast binary template matching algorithm for document image data compression,' in *Pattern Recognition*, J. Kittler (ed.) (Proc. Int. Conf., Cambridge). Springer Verlag, Berlin.

Johnsen, O., Segen, J. and Cash, G.L. (1983) 'Coding of two-level pictures by pattern matching and substitution,' *Bell System Technical J* 62(8): 2513-2545; May.

Langdon, G.G. and Rissanen, J. (1981) "Compression of black-white images with arithmetic coding," *IEEE Trans Communications* COM-29(6): 858–867; June.

Mohiuddin, K., Rissanen, J. and Arps, R. (1984) 'Lossless binary image compression based on pattern matching' *International Conference on Computers, Systems & Signal Processing* pp. 447-451, Bangalore, India; December.

Moffat, A. (1991) 'Two level context based compression of binary images,' in *Proc. DCC'91*, J.A. Storer and J.H. Reif (eds.), pp. 382-391. IEEE Computer Society Press, Los Alamitos, CA.

Pratt, W.K., Capitant, P.J., Chen, W.H., Hamilton, E.R. and Wallis, R.H. (1980) 'Combined symbol matching facsimile data compression system,' *Proc IEEE* 68(7): 786-796; July.

Witten, I.H., Bell, Harrison, James and Moffat, A. (1992) 'Textual Image Compression' in *Proc. DCC'92*, J.A. Storer and M. Cohn (eds.), pp. 42-51. IEEE Computer Society Press, Los Alamitos, CA.

Witten, I.H., Moffat, A. and Bell T.C. (forthcoming) *Managing gigabytes: Compressing and indexing large collections of documents and images*. Van Nostrand Reinhold, New York.