EAG: Edge Adaptive Grid Data Hiding for Binary Image Authentication

Hong Cao[†] and Alex C. Kot[*]
[†]Institute for Infocomm Research, A*STAR, Singapore
E-mail: hcao@i2r.a-star.edu.sg Tel: +65-64082157
[*]Nanyang Technological University, Singapore
E-mail: eackot@ntu.edu.sg Tel: +65-67904946

***Abstract*— This paper proposes a novel data hiding method for authenticating binary images through establishing dense edge adaptive grids (EAG) for invariantly selecting good data carrying pixel locations (DCPL). Our method employs dynamic system structure with carefully designed local content adaptive processes (CAP) to iteratively trace new contour segments and to search for new DCPLs. By maintaining and updating a location status map, we re-design the fundamental content adaptive switch and a protection mechanism is proposed to preserve the local CAPs' contexts as well as their corresponding outcomes. Different from existing contour-based methods, our method addresses a key interference issue and has unprecedentedly demonstrated to invariantly select a same sequence of DCPLs for an arbitrary binary host image and its marked versions for our contour-tracing based hiding method. Comparison also shows that our method achieves better trade-off between large capacity and good perceptional quality as compared with several prior works for representative binary text and cartoon images.**

I. INTRODUCTION

Data hiding is the art of concealing a secret message into an innocent-looking host media by incurring the least perceptional distortion. With fast proliferation of digital multimedia, numerous data hiding schemes have been proposed for various media entities including images [1, 3-15], audio [1], video [1] and electronic inks [2].

This paper discusses data hiding for authenticating binary images. A binary image requires only 1 bit per pixel as compared 24 bits per color pixel. The small storage requirement makes binary images ideal for digitizing, processing, transmitting and archiving large amount of daily documents whose contents are typically black and white in nature. As digital images can be easily altered electronically nowadays with no visible traces left, a security concern arises on how to verify their integrity and how to detect malicious tampering on all types of images including binary images.

Several early works [8-9, 11-14] have described similar hybrid authentication systems by integrating data hiding and public-key cryptographic techniques. The idea is to divide a host image into two regions: the quasi-image region R1 and a region R2 of scattered embeddable locations using a data hiding technique as shown in Fig. 1. A cryptographic image hash computed based on the region R1 (sometimes together with certain user payload data, e.g. a logo) is embedded by modifying the R2 region to generate the secure marked image.

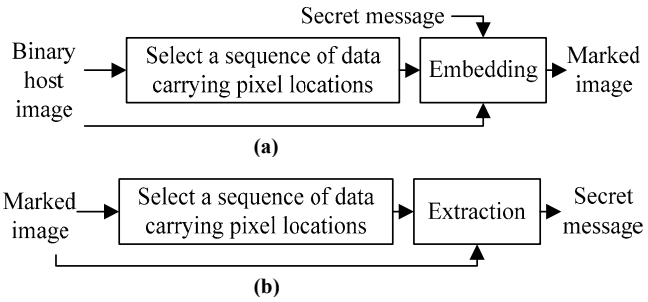

Figure 1. Block diagram of the embedding (a) and extraction (b)

The marked image visually resembles the host image with little noticeable distortions. Upon verification, the marked image is re-divided into the identical R1 and R2 regions. The secret message is then extracted from R2, decrypted and compared with re-computed image hash on R1 for image integrity verification or alteration localization. Compared with several other authentication schemes, e.g. [4], the hybrid scheme is advantageous for its high level of security, which is assured by the modern cryptography. However, a remarkable challenge to realize the hybrid authentication system for binary images lies in the data hiding technique, i.e. how to invariantly locate a large number of good data carrying pixel locations (DCPL) in the R2 region and, at the same time, ensuring the least visual discrepancies on the marked images. This paper addresses this challenge by proposing a new edge-based data hiding framework, which is demonstrated to select more the best-quality embeddable locations corresponding to ℓ-shape patterns than the conventional pattern-based methods. Moreover as a contour-tracing based method, our embedding and extraction processes are robust to edge noises and are applicable to secure arbitrary binary host images.

Among the existing binary data hiding methods, Low et al. [3] manipulated the space between lines and words for copyright protection on binary text images. Mei et al. [10] matched 100 pairs of 5-pixel long interchangeable contour patterns to hide data in text images. Wu and Liu [4] ranked the priority for flipping each pixel by measuring its "flippability" score based on smoothness and connectivity criterion in 3×3 patterns. Random shuffling was suggested to evenly redistribute the flippable pixels and to enhance the

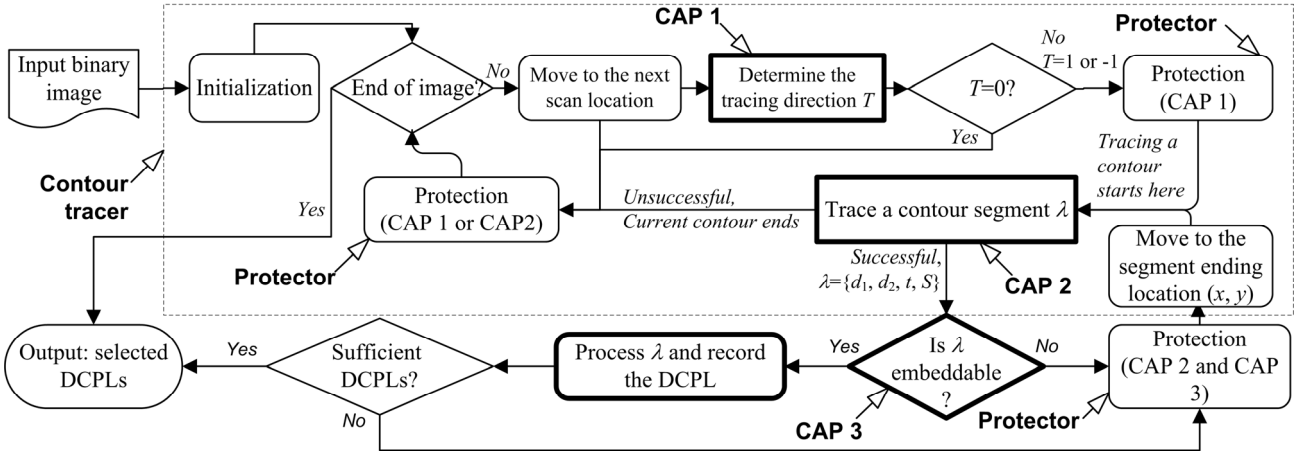

Figure 2. Flow graph of our selection engine for Data Carrying Pixel Locations (DCPL), where the content adaptive processing (CAP) blocks are highlighted with bold lines

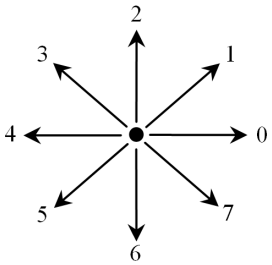

Figure 3. Chain code definition [16]

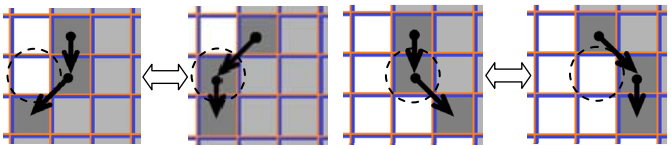

Figure 4. Two pairs of interchangeable contour segment patterns (CSP). CSPs that differ by rotation and tracing direction are not shown

security. By dividing the shuffled image into blocks, each block embeds a secret bit by enforcing its odd/even feature of the block. Similarly, Pan et al. [5] selected the 4×4 embeddable superblocks, each for hiding a secret bit, by analyzing patterns of their 3×3 sub-blocks. Tseng et al. [6] proposed a matrix embedding based stegnographic scheme, which achieves large capacity at the expense of relatively poor visual quality on the flipped pixels. Yang and Kot [8] suggested three connectivity preserving criteria for selecting good embeddable blocks. Various block division schemes including the different block sizes, interlacing or non-interlacing were investigated. In [9], Yang et al. further used interlaced morphological wavelet transform for tracking the shifted edges. Based on 2×2 blocks, double processing and orthogonal embedding techniques are employed to yield large hiding capacity without substantially degrading visual quality of marked images as compared with [8].

In this paper, we propose a pixel-wise data hiding method for the hybrid authentication system. As shown Fig. 1, its core engine invariantly selects a large number of good DCPLs in both embedding and extraction. This is achieved through establishing dense edge-adaptive grid (EAG) along the object contours, i.e. through tracing 3-pixel long contour segments. Our study shows that the EAG more efficiently selects the good DCPLs associated with the best rated "ℓ-shape" patterns [4, 8, 15] than several existing block-based methods [4, 5, 8]. It is worthwhile to note that our proposal uniquely addresses a key challenge that the embedding changes can easily interfere with the local content adaptive processes (CAP) and result in errors in re-discovering the DCPLs and errors in the extraction. Using our redesigned content adaptive switches and a novel protection mechanism with carefully designed CAPs, comparison shows our method works well for arbitrary binary images with a good trade-off between capacity and perceptional quality.

II. Proposed Algorithm

Fig. 2 shows the flow graph of the core engine, consisting of three main components, a contour tracer, content-adaptive processes (CAP) and a protector. Contour tracer traverses the entire image to establish the EAG, i.e. to search for new contour starting locations and to trace a new contour segment (CS) iteratively. Three CAPs are designed to make local content-adaptive processing, where **CAP 1** and **CAP 2** are associated with the contour tracer and **CAP 3** determines the embeddability of a current processing CS. The protector protects the context of a CAP for preserving the same outcomes being achieved in the marked image.

A. Contour Tracer and CAPs

As detailed in Fig. 2, during the initialization, we initialize an accessorial status map for all locations and a stack. The location status map is used to keep track of the current processing statuses while searching for new DCPLs. Initially, we set the status for each location as "0" indicating that the pixel has not been processed and is in an "unprotected" status.

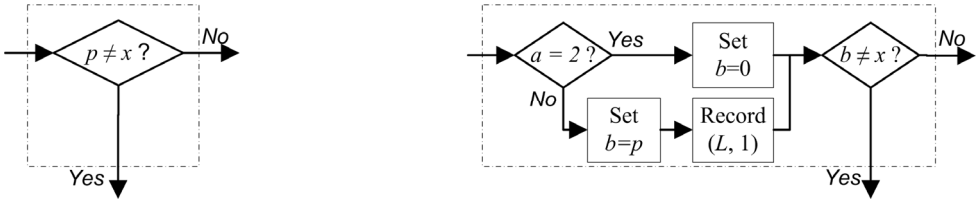

Figure 5. A content adaptive processing (CAP) unit in (a) and our re-design in (b) for protecting each CP unit from being affected by the embedding changes, where $p \in \{0,1\}$ is the pixel's value at location L of the input image, a is current status of the pixel and b is temporary variable

The stack is used to keep track of the context to be protected due to the CAPs. We initialize an empty stack.

After initialization, our contour tracer starts with scanning for a starting contour pixel in a pre-defined sequence. For each pixel, **CAP 1** is applied to check its validity of being a starting contour pixel with three possible outcomes $T \in \{0,+1,-1\}$, where "0", "+1" and "-1" indicates "not valid", "valid with counter clockwise (CCW) tracing" and "valid with clockwise (CW) tracing", respectively. We use the radio sweep tracing algorithm and only a black pixel location from which, a second contour pixel in its 3×3 neighborhood can be traced based on CCW or CW, is determined as a valid contour pixel. Once such a pixel is found, we temporarily suspend the scanning at this pixel location L and use **CAP 2** (radio sweep algorithm) to trace a new contour segment originated from L. Using a fixed-length segmentation, once three consecutive black pixels are traced, we output them as one current processing contour segment (CPCS). This CPCS can be represented as $\lambda = \{d_0, d_1, t, S\}$, where $d_0, d_1 \in \{0, 1, \ldots, 7\}$ denote the transitions, in terms of chain codes (Fig. 3), from the first contour pixel to the second, and from the second to the third, respectively. $t \in \{+1,-1\}$ denotes either CCW or CW tracing direction and S is the ending pixel location of λ. For each CPCS, we apply **CAP 3** to determine its embeddability, where an embeddable CPCS contributes one DCPL. Our **CAP 3** requires an embeddable CPCS to belong to one of the four pattern types in Fig. 4, with the following equation

$$(d_1 - d_2) \bmod 8 = 1 \text{ or } 7 \qquad (1)$$

being satisfied. Also, flipping the embeddable pixel (as highlighted with dotted circles in Fig. 4) would not violate our protector's rules (see Sec. 2.3) and cause its dual CPCS not traceable using the same contour tracing algorithm.

After the CPCS is processed, our contour tracer moves to its ending location and starts there tracing a new contour segment. This process iterates until the current contour ends. At this moment, the previously suspended scanning is restarted from the next scanning location to search for new contour starting pixel locations and new contour segments. The entire process ends either when the required number of DCPLs is recorded or when the entire input image is scanned.

B. Design of CAP Switch

One key challenge for contour-based binary data hiding is that contour tracing involves tremendous amount of local content-dependant processing. Outcomes of these processes depend on image content and can be easily interfered by the embedding changes so that the sequence of DCPLs selected from a host image in the embedding cannot be rediscovered exactly in its marked version, particularly for binary host images with noises present near the edge pixels. This can cause undesired extraction errors and failures of the hybrid authentication. Current contour-based hiding methods, e.g. the work in [10], demands relatively clean object contour of the text and may not work for practical binary images, whose edges or part of the image region is mixed with substantial amount of noises.

To address this challenge, we observe that each CAP process is actually comprised of many similar fundamental CAP switches. As shown in Fig. 5(a), this switch checks whether p (the current pixel value at the location L of the input image) is unequal to a target value x with two different output paths corresponding to *Yes* and *No*. This CAP switch is the most fundamental and it can be shown that other more complex switch can be implemented as a combination of this basic type of switch. We have redesigned this CAP switch as shown in Fig. 5 (b). The new switch first checks whether the current status a of location L is equal to 2 (being quarantined). If *yes*, regardless of the actual p, b is assigned as 0 (black pixel) for the subsequent comparison with x. Otherwise, b is assigned the actual value of p for the comparison. In the later case, we also record (L, 1) into the protector's stack, for updating location L to a target status of 1 at a later time. The redesigned switch allows us quarantining a newly selected DCPL by **CAP 3** so that change of its current pixel value does not affect other local CAPs. Also, it protects the context of this CAP switch, i.e. the location whose pixel value has been checked for making a flow-control decision.

C. Protector

We design a protector to preserve all previous CAPs' context and to facilitate the processing efficiency. As mentioned earlier, this is achieved by keeping track of an accessorial status map for all locations and a stack. For each location L, we define three possible statues $A(L) \in \{0, 1, 2\}$. Locations with Status 0 are currently unprotected and all pixel locations are assigned as "unprotected" during the initialization (Fig. 2). Locations with Status 1 are currently protected; therefore, they are not allowed to be selected as new DCPLs to incur possible embedding changes in the future processing. But we still allow them being traced as new contour pixels. Locations with Status 2 are currently

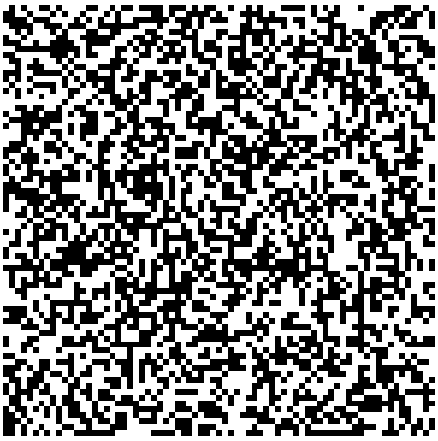

Figure 6. An image of random noises. The image is enlarged so that each pixel can be seen clearly.

quarantined meaning these locations can be neither selected as a new DCPL nor traced as a new contour pixel. The locations subject to possible future pixel changes, e.g. a newly selected DCPL, shall be assigned to a "quarantined" status. We use the stack to dynamically keep track of the context of a CAP and their targeted protection statuses. An empty stack is initialized at the beginning.

In the two "Protection" blocks in Fig. 2, the accumulated records in the stack (due to our switch design in Fig. 5(b) and due to the DCPL selection by **CAP 3** in Fig. 2.) are read out. For each record (Location: L, Targeted status: n) in the stack, we perform a status update by setting $A(L) = \max(n, \widehat{A}(L))$, where $\widehat{A}(L)$ is the status of L before this update. Note that $A(L) \geq \widehat{A}(L)$ is always satisfied implying that status of a location cannot be lowered. We empty the stack after each "protection" block for the subsequent CAP.

In Fig. 2, protection is performed immediately after a CAP to ensure that no DCPL will be selected from its context in the subsequent processing because only an unprotected location, i.e. with a status 0, can be selected as a new DCPL according to our aforementioned **CAP 3**. This ensures that each CAP can be performed invariantly in the extraction with the same outcome.

III. Experimental Results

To validate the invariance of our EAG in selecting the DPCLs for arbitrary host images and their marked versions, we synthetically generated 100 binary images of 512×512 containing random noises. Fig. 6 shows a cropped portion of such noise image. From the image, we can see that there exists no dominant object contour and the interference between nearby contours are significant. Using each of the 100 images as a host image, we discover its DCPLs with our engine in Fig. 2 and embed random secret message of about 4800 bits by enforcing the DCPLs' pixel values to be same as the secret bits. We then re-identify the DPCLs from the marked image and extract the secret message. Through comparison, we find that for all 100 cases, a same sequence of DCPLs is re-selected from the marked image as those from its host image to have correct extraction of the secret message. Our successful embedding and extraction for these extremely noisy binary host images with no clear object contours also show that our design ensures the embedding changes do not affect our re-discovery of the same sequence of DCPLs in the extraction. To the best of our knowledge, this is the first demonstration that our contour-tracing based data hiding method can work on such extremely noisy binary host images. Though practically we do not perform data hiding on such images with no meaningful visual information, the experimental outcomes suggest that our proposed method would work well for all types of contours and our embedding/extraction mechanism would not be affected by the interfering noises.

We have successfully tested our data hiding method over hundreds of practical bi-level images. A subset of about 200 such images is shared at https://sites.google.com/site/sstarcao/. Table 1 shows a comparison with several other relevant data hiding methods using three representative images, i.e. Chinese text, English text and Cartoon host images. All these methods can be applied to the hybrid authentication. The results show that our method achieves the least perceptional distortion per flipping measured in Edge Line Segment Similarity (ELSS) [15] since our method selects mostly the best rated hiding locations, i.e. centers of ℓ-shape patterns [4, 8, 15], as illustrated in Fig. 4. Note that we choose the ELSS/per flipping to measure the visual quality because the marked images using different hiding algorithms have been embedded with different amount of secret bits with different number of alterations. Though Yang's interlaced 4×4 block method (IB4), i.e. state-of-the-art block-based method, has comparable capacity with ours for generic data hiding, it experiences an average drop of 25.7% in capacity when used for hybrid authentication, as some embeddable locations must be deselected to avoid parity attacks [9]. Also our average ELSS distortion per flipping is 30% lower than IB4. Another contour-based method, Mei et al. [10], achieves comparable low ELSS distortion per flipping as ours, but their hiding capacity is only 54% of our achieved capacity. From the results, we can see that our method has achieved better trade-off between the large hiding capacity and good visual quality.

The processing of our proposed algorithm is fairly efficient. For a full-capacity embedding case, our algorithm takes $M \times N$ operations to scan through the entire host image in one pass, where M and N denote the number of rows and columns in the host image, respectively. The basic contour tracing operation will be performed for about B times, where B denotes the number of black boundary pixels. In our current implementation in C++, it takes only about 3.5×10^{-2} second to complete embedding 4060 bits in a French Text image of 512×512 pixels with a common 2.53-GHz Duo CPU. For an A4-Page of binary "English text" image scanned in 300 dot per inch (DPI) with a size of 2479×3507, our algorithm takes about 0.36 second to embed 30391 secret bits. These suggest that our algorithm is efficient for practical use.

Table 1. Comparison of binary data hiding methods when they are configured for hybrid authentication (Random message is embedded)

	Chinese Text	English Text	Cartoon
(a) Host images of size 157×73			
(b) Proposed with **106**, **220** and **226** bits embedded, respectively. Ave. ELSS distortion = **0.98**			
(c) Mei *et al.* [10] with **59**, **137** and **101** bits embedded, respectively. Ave. ELSS distortion = **1.11**			
(d) Yang *et al.* [9] IB4 method with **85**, **153** and **190** bits embedded, respectively. Ave. ELSS distortion = **1.40** (For generic data hiding (non-hybrid authentication), the above 3 capacities can increase to **122**, **212** and **235**, respectively.			

IV. CONCLUSION

In this paper, we proposed to establish dense edge-adaptive grid to efficiently select good data carrying pixel locations (DCPL). Through designing a novel protection scheme with our carefully re-designed content-adaptive process (CAP) switch, our dynamic system addressed an interference issue between the embedding changes and the CAP outcomes. Experimentally, we demonstrated that our contour-based method invariantly selected the DCPLs from arbitrary binary host images, e.g. random noise images, and their marked versions. Our method well fits the region-separation requirement for state-of-the-art hybrid authentication systems. For such hybrid authentication, our method offered 25.7% capacity improvement and 30.0% reduction of the distortion over the existing interlaced 4×4 block (IB4) scheme. Therefore, our method achieved a better trade-off between large data hiding capacity and good perceptional quality than the prior binary data hiding schemes.

Along this avenue, our future work will prove that the proposed algorithm and framework invariantly selects DPCLs from arbitrary binary host images.

REFERENCES

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, Digital Watermarking. Sam Mateo, CA: Morgan Kaufmann, 2001.

[2] H. Cao and A. C. Kot, "Lossless Data Embedding in Electronic Inks," IEEE Trans. on Information Forensics and Security, vol. 5 (2), pp.314-323, June 2010.

[3] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document Identification for Copyright Protection using Centroid Detection," IEEE Trans. on Communications, vol. 46 (3), pp.372-383, 1998.

[4] M. Wu and B. Liu, "Data Hiding in Binary Image for Authentication and Annotation," IEEE Trans. on Multimedia, vol. 6 (4), pp. 528-538, 2004.

[5] G. Pan, Z. Wu, and Y. Pan, "A Data Hiding Method for Few-Color Images," in Proc. ICASSP. vol. 4, 2002, pp. IV-3469 – IV-3472.

[6] Y.-C. Tseng, Y.-Y. Chen, and H.-K. Pan, "A Secure Data Hiding Scheme for Binary Images," IEEE Trans. on Communications, vol. 50, pp. 1227-1231, 2002.

[7] H. Lu, A. C. Kot, and Y. Q. Shi, "Distance-Reciprocal Distortion Measure for Binary Document Images," IEEE Signal Processing Letters, vol. 11 (2), pp. 228-231, 2004.

[8] H. Yang and A. C. Kot, "Pattern-Based Data Hiding for Binary Image Authentication by Connectivity-Preserving," IEEE Trans. on Multimedia, vol. 9 (3), pp. 475-486, 2007.

[9] H. Yang, A. C. Kot, and S. Rahardja, "Orthogonal Data Embedding for Binary Images in Morphological Transform Domain - A High-Capacity Approach," IEEE Trans. on Multimedia, vol. 10 (3), pp. 339-351, 2008.

[10] Q. G. Mei, E. K. Wong, and N. D. Memon, "Data Hiding in Binary Text Documents," in Proc. SPIE. vol. 4314, 2001, pp. 369-375.

[11] P. W. Wong and N. Memon, "Secret and Public Key Image Watermarking Schemes for Image Authentication and

Ownership Verification," IEEE Trans. on Image Processing, vol. 10 (10), pp. 1593-1601, 2001.
[12] H. Y. Kim and A. Afif, "Secure Authentication Watermarking for Binary Images," in Proc. SIBGRAPI, 2003, pp. 199-206.
[13] H. Y. Kim and R. L. de Queiroz, "A Public-Key Authentication Watermarking for Binary Images," in Proc. ICIP. vol. 5, 2004, pp. 3459-3462.
[14] H. Y. Kim and R. L. de Queiroz, "Alternation-Locating Authentication Watermarking for Binary Images," in Proc. Int. Workshop on Digital Watermarking (IWDW), 2004. pp. 125-136.
[15] J. Cheng and A. C. Kot, "Objective Distortion Measure for Binary Text Image Based on Edge Line Segment Similarity " IEEE Trans. on Image Processing, vol. 16 (6), pp. 1691-1695, 2007.
[16] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," IEEE Trans. on Electronic Computers, vol. EC-10 (2), pp. 260-268, 1961.740-741, August 1987 [*Digests 9th Annual Conf. Magnetics Japan*, p. 301, 1982].
[17] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.