DOCUMENT IMAGE WATERMARKING ALGORITHM BASED ON NEIGHBORHOOD PIXEL RATIO

Shiyan Hu

Department of Computer and Information Science
Polytechnic University
Brooklyn, NY 11201
shu@cis.poly.edu

ABSTRACT

In this paper, a new method for document image watermarking is proposed. The method first divides the document into weight-invariant partitions in the spatial domain using a novel image feature - the neighborhood pixel ratio. Such a ratio effectively summarizes the local information and is robust to common attacks. The document image is then partitioned in a key-dependant way for high capacity and security. Our experimental results demonstrate the effectiveness of the new method.

1. INTRODUCTION

Digital image watermarking has become a very active research topic recently. There are a lot of techniques for gray-scale and color image watermarking in the literature such as [1, 2]. In contrast, there is no much work on document image watermarking. This is mainly due to the fact that most general image watermarking methods are based on transform-domain techniques and are less useful for document image watermarking because their modifications in documents tend to be visible and are easily removed by binarization [3]. A number of methods specific for document image watermarking have been proposed in the literature such as [4, 5, 6, 7].

This paper presents a new method for document image watermarking. The method first divides the document into some weight-invariant partitions [8] in the spatial domain using the neighborhood pixel ratio. Such a ratio effectively summarizes local information and provides robustness to many common attacks. Following [1, 8], the image partition problem involved in the method is then reduced the problem to the bottleneck hamiltonian path problem, which identifies a hamiltonian path (in a complete weighted graph) that minimizes the length of the longest travelled edge. We solve this combinatorial problem in the key-dependant way by an ant colony algorithm. The experimental results demonstrate the effectiveness of the new method.

The rest of the paper is organized as follows: Section 2 describes the method for watermarking a uniform partition using neighborhood pixel ratio. Section 3 describes the process of partitioning a document image. Section 4 presents the experimental results. A summary of work in given in Section 5.

2. EMBEDDING WATERMARK INTO A UNIFORM PARTITION USING NEIGHBORHOOD PIXEL RATIO

2.1. Uniform Partition

Suppose that we want to embed the binary sequence τ into a document. To *partition* a document is to divide the document into pair-wise disjoint parts. In the following, we abuse the term "partition" a little: we also let the partition denote the disjoint part in the above definition. Let $S(P)$ of cardinality n (i.e., $|S(P)| = n$) denote the set of all text lines in a partition P. The *weight* $w(l_i)$ of a text line l_i is defined as follows. For each pixel p in l_i, we check its eight neighbors: if more than three neighbors of p are (black) pixels, a counter (corresponding to l_i) is incremented. Then $w(l_i)$ is defined to be $\frac{counter}{h_i \cdot w_i}$ where h_i and w_i are height and width of l_i, respectively. We call this ratio (line weight) the *neighborhood pixel ratio*. The *average weight* $A(P)$ equals to the sum of the weight of all text lines in $S(P)$ divided by n. We also call $A(P)$ the *weight of the partition* P. Let S_{in} of P denote the set of text lines with weight between $A(P) \pm \delta$, δ being a positive value to be discussed later. Let S_{out} be $S - S_{in}$. As the major contribution in this paper, the neighborhood pixel ratio of a text line is not likely to be significantly changed due to noise, and the ratio can be easily computed for each text line. In the following, we combine the neighborhood pixel ratio with weight-invariant partition and key-dependant decomposition techniques presented in [8, 1] to give a new document image watermarking method.

The basic idea of weight-invariant partition is that some text lines in a partition are first modified to have special weights, watermark are then embedded into these text lines, finally other text lines are modified to maintain the weight of the partition. Informally, a partition P is a *uniform partition* if half of lines in $S(P)$ have the weight close to $A(P)$. From this, we expect that the median and the mean of weights of lines in $S(P)$ are roughly equivalent. Therefore, if δ is appropriately chosen, $|S_{in}|$ will not be too small and we can embed enough bits into a partition as follows. We first modify the partition (see below) so that *all lines in* S_{in} have weights of exactly $A + \delta$ or $A - \delta$. This process will not lead to perceptible modifications if δ is small enough. After it, τ is sequentially embedded to each line in S_{in}. We will further modify $l_i \in S_{in}$ only when a 0 is to be embedded (to l_i) such that after modification, l_i has a weight of exactly $A + \delta/2$ (if $w(l_i) = A + \delta$ before embedding 0) or $A - \delta/2$ (otherwise). We now show how to determine δ. Since half of lines have weights close to $A(P)$, we first increasingly sort all lines in the partition according to their neighborhood pixel ratios and select the middle $n/2$ lines starting with l_i and ending with l_j to form S_{in}. So $\delta = \min\{A - w(l_i), w(l_j) - A\}$. Suppose that after embedding the watermark, $A(P)$ is increased. In order to maintain $A(P)$, we decrease some line weights in the lower weight lines in S_{out} such that their weight become even lower. We similarly treat the case when $A(P)$ is decreased after embedding τ. Note that we embed "01" before τ to ensure that at least one 0 and 1 are embedded to a partition, which is useful for the watermarking extractor.

The watermark extracting phase is much simpler. Since the weight of a partition is not changed after the above process, we can simply search in the embedded partition, compute $A(P)$, and then compute δ as follows: find the line l_i such that $|A - w(l_i)| = \min_k |A - w(l_k)|$, then $\delta = 2|A - w(l_i)|$. With δ, the watermark extraction is straightforward. Since there always has small noise, we have to set an error-tolerant constant ξ for practical purpose, i.e., we treat $\delta \approx \delta \pm \xi$.

It remains to present the principle for modifying a text line. We try to add (resp. remove) pixels to (resp. from) the characters which contain many pixels, and for a single character, the principle is to modify its boundary. When we add (or remove) a pixel to a boundary, we need to update the counter by inspecting its eight neighbors. The process is repeated until the counter reaches the goal within an error of $4/h_i w_i$.

2.2. Recursive Partition and Robustness

As discussed in [8], a natural question arises: how to compute uniform partitions from the original document? The simplest way is to group a fixed number of consecutive pages as a partition and then check for the uniform condition. If the partition is not uniform, it will be further divided. This process is repeated until certain partitions are uniform. The main assumption of the above strategy is that a reasonable number of physically consecutive lines can form a uniform partition, however, this is not always possible. Even if it is the case, partitions formed in this way are usually small while there are not many uniform partitions, which greatly limits the capacity of the method. Therefore, we introduce an improved partition method in Section 3.

For robustness, we note that the average weight of a partition is hard to be changed due to noise, assuming that the partition itself is large enough. Even for a single text line, noise can be often "filtered out", e.g., small noise separate from characters' boundaries can be "filtered out" from incrementing the counter (in computing the line ratio) since we consider a pixel as a text pixel only when at least four of its neighbors are (black) pixels. We illustrate this fact through an example, refer to Figure 1. The original line and the noisy line are visually different, however, their neighborhood pixel ratios are 0.1028 and 0.1041, respectively. Two ratios only differ by 1.3%. Refer to Section 4 for further experiments on robustness of the watermark to common attacks.

3. PARTITION BY APPROXIMATING BOTTLENECK HAMILTONIAN PATH

If the document is not partitioned in a regular way, the watermarking scheme will be more secure, since one must know the partitions before extracting the watermark from them. In addition, we expect that most lines in a partition are similar or *logically close* (here, two lines are similar if their weights are close) such that (1) only slight modification needs to be carried out in a single partition; (2) more lines can be selected to embed watermark and thus watermarking capacity may be increased. However, it is not easy to achieve in general if we always consecutively select lines. To compute a partition composed of logically close lines, we first reduce the image partition problem to a combinatorial problem and then an efficient metaheuristic algorithm is applied to solve it in a key-dependant way. Such a method has been successfully explored in [1, 8]. For completeness, we include the procedure as follows.

Consider an undirected weighted complete graph $G = (V, E)$ where every line l_i in the document is mapped to a node v_i and the weight of the edge linking v_i and v_j is the absolute value of the difference between the weights of two corresponding lines l_i and l_j. Since at most one bit can be embedded to a line, we can denote embedding a line by visiting its corresponding node in G. Then we want to compute a path visiting every node exactly once (i.e., a *hamiltonian path*), and select some consecutive lines along the path to form partitions. Such a path may not be arbitrary. Because it is more likely to embed enough bits if the lines within a partition are similar to each other, the edges linking

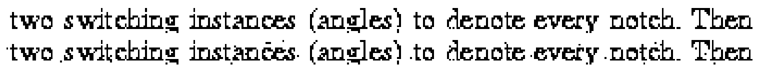

two switching instances (angles) to denote every notch. Then
two switching instances (angles) to denote every notch. Then

Fig. 1. The original line (top) and the line corrupted with A.W.G.N. followed by thresholding to a binary image (bottom)

the corresponding nodes are expected to be short. Therefore, the problem is reduced to the *Bottleneck Hamiltonian Path Problem*, where one is interested in finding a hamiltonian path that minimizes the length of the *longest travelled edge*. Formally, given an $n \times n$ matrix $C = (c_{ij})$, we are to find an acyclic permutation π of $\{1, 2, \ldots, n\}$ such that $\max_{1 \le i \le n}\{c_{i\pi(i)}\}$ is minimized.

The bottleneck hamiltonian path problem is a well-known NP-hard problem and we compute an approximation by applying an *ant colony algorithm*, which is very effective and efficient in solving tough combinatorial optimization problems. As a stochastic algorithm, ant colony algorithm involves using pseudo-random numbers. To make it sensitive to our key, we generate a pool of pseudo-random numbers seeded with the secret key. The algorithm uses these numbers sequentially and terminates when all numbers have been used up. It is worth noting that the partitions (the line indices) need to be recorded as a part of the key for extracting the watermark, which makes the watermark more secure than the one without the key-dependant partition phase. A basic ant colony algorithm in [9, 1] reads as follows.

For an instance of bottleneck hamiltonian path problem, in addition to the edge weight, each edge has a desirability measure $\tau(r, s)$, called *pheromone*. In the initialization step, m ants are randomly positioned on nodes of the graph. Each ant generates a complete hamiltonian path by choosing nodes according to a *probabilistic state transition rule*. While constructing its path, an ant also modifies the amount of pheromone on the visited edges by applying the *local updating rule*. Once all ants have completed their paths, a *global updating rule* is applied to modify the pheromone on edges. This process is iterated until all pseudo-random numbers have been used up as mentioned before. In general, ants are guided in building their paths by both heuristic information (preference of short edges) and by pheromone information (preference of high amount of pheromone). The further details are omitted here due to space limitation.

4. EXPERIMENTAL RESULTS

We first test the proposed scheme on embedding a random 10-bit string into a single uniform partition[1]. The original partition and watermarked partition are shown in Figure 2. The modification is imperceptible. A detailed portion of Figure 2 is shown on Figure 3, where one sees the modification by the embedding process.

[1]Recall that we also need to embed a "01" before the watermark.

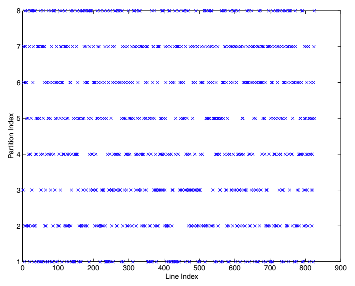
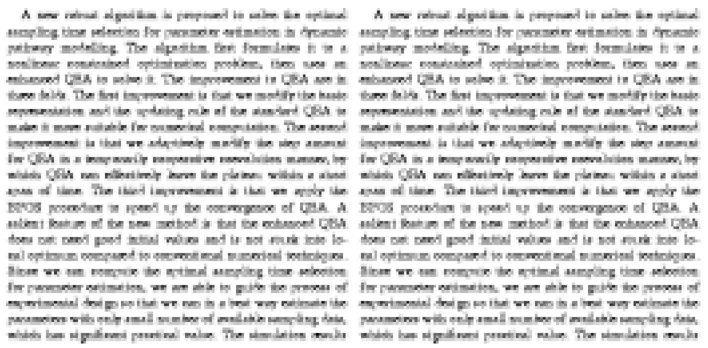


Fig. 4. Distribution of lines in partitions

To evaluate the robustness of the proposed watermarking scheme, experiments have been conducted on common attacks. We apply the JPEG compression of Quality Factor 10% to the gray-scale version of the binary image, then the resulting image is converted back to a binary image. The geometric distortion attacks include translation, rotation by $5°$ and $10°$ and scaling by the factor of 0.5 and 2. Note that for rotation, a geometric skew detection algorithm [10] for document images is applied. The experiment is carried out on 50 document images: they are first watermarked followed by the above attacks and the marks are successful extracted. This indicates that the proposed document image watermarking withstands all of the above attacks.

We next apply our method to embed a 200-bit sequence into a twenty-three-page single-column paper. We first preprocess the document such that it contains only text by the standard block extraction method. We then apply the ant colony algorithm to compute a key-dependant approximation of bottleneck hamiltonian path. In the experiment, we select every consecutive 100 lines (except for the last partition) along the bottleneck hamiltonian path to form one partition and we have 8 partitions in a total. The results are summarized in Table 1. From it, one sees that the average line weight (average neighborhood pixel ratio) of a partition before and after embedding is very close, while the standard deviation differs a little, which is due to our embedding phase.

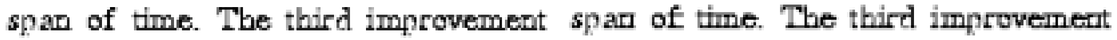
Figure 4 shows which partition each indexed line of the image is classified into (line index vs. partition index). Clearly within each partition, the lines are not always physically close but logically close (see also Table 1). Therefore, we can embed enough number of bits into the partitions. In addition, it is in general hard for others to tell which lines belong to a certain partition and this information is necessary for extracting the watermark. The new scheme is therefore



A new robust algorithm is proposed to solve the optimal sampling time selection for parameter estimation in dynamic pathway modelling. The algorithm first formulates it to a nonlinear constrained optimization problem, then uses an enhanced QGA to solve it. The improvement to QGA are in three folds. The first improvement is that we modify the basic representation and the updating rule of the standard QGA to make it more suitable for numerical computation. The second improvement is that we adaptively modify the step amount for QGA in a temporarily cooperative coevolution manner, by which QGA can effectively learn the plateau within a short span of time. The third improvement is that we apply the BFGS procedure to speed up the convergence of QGA. A salient feature of the new method is that the enhanced QGA does not need good initial values and is not stuck into local optimum compared to conventional numerical techniques. Since we can compute the optimal sampling time selection for parameter estimation, we are able to guide the process of experimental design so that we can in a best way estimate the parameters with only small number of available sampling data, which has significant practical value. The simulation results

A new robust algorithm is proposed to solve the optimal sampling time selection for parameter estimation in dynamic pathway modelling. The algorithm first formulates it to a nonlinear constrained optimization problem, then uses an enhanced QGA to solve it. The improvement to QGA are in three folds. The first improvement is that we modify the basic representation and the updating rule of the standard QGA to make it more suitable for numerical computation. The second improvement is that we adaptively modify the step amount for QGA in a temporarily cooperative coevolution manner, by which QGA can effectively learn the plateau within a short span of time. The third improvement is that we apply the BFGS procedure to speed up the convergence of QGA. A salient feature of the new method is that the enhanced QGA does not need good initial values and is not stuck into local optimum compared to conventional numerical techniques. Since we can compute the optimal sampling time selection for parameter estimation, we are able to guide the process of experimental design so that we can in a best way estimate the parameters with only small number of available sampling data, which has significant practical value. The simulation results

Fig. 2. The original partition (left) and the watermarked partition (right)



span of time. The third improvement

span of time. The third improvement

Fig. 3. A detailed portion of Figure 2: the original text line (left) and the watermarked line (right).

Table 1. Average line ratio (neighborhood pixel ratio) and its standard deviation before/after the embedding phase for a document.

Original Document		Embedded Document	
Line Avg.	Stan. Dev.	Line Avg.	Stan. Dev.
0.1061	0.0458	0.1053	0.0589
0.1102	0.0390	0.1109	0.0610
0.0993	0.0382	0.0989	0.0602
0.0831	0.0293	0.0820	0.0503
0.1205	0.0462	0.1202	0.0716
0.1268	0.0318	0.1272	0.0678
0.0895	0.0301	0.0887	0.0539
0.1332	0.0491	0.1333	0.0622

secure.

5. CONCLUSIONS

The main contribution of this paper is to propose a new image feature - namely the neighborhood pixel ratio for watermarking document image. Such a new feature is robust to common attacks and when combined with the weight-invariant partition technique in [8], a new document image watermarking is presented. A key-dependant decomposition scheme is also adopted to improve the method in terms of security and capacity. Our experimental results demonstrate the effectiveness of the new method.

6. REFERENCES

[1] S. Hu, "Key-dependant decomposition based image watermarking," *Proceedings of ACM Multimedia*, 2004.

[2] X. Qi and J. Qi, "Improved affine resistant watermarking by using robust templates," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. III, pp. 405–408, 2004.

[3] S. Low and N. Maxemchuk, "Capacity of text marking channel," *IEEE Signal Processing Letters*, vol. 7, no. 12, pp. 345–347, Dec 2000.

[4] T. Amano and D. Misaki, "A feature calibration method for watermarking of document images," *Proc. 5th Int. Conf. Document Analysis and Recognition*, pp. 91–94, Sept 1999.

[5] J. Brassil, S. Low, and N. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proc. IEEE*, vol. 87, pp. 1181–1196, July 1999.

[6] J. Brassil and L. O'Gorman, "Watermarking document images with bounding box expansion," *Proc. Info Hiding'96*, pp. 227–235, 1996.

[7] S. Low, N. Maxemchuk, and A. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. Communications*, vol. 46, pp. 372–383, March 1998.

[8] S. Hu, "A new document watermarking algorithm based on hybrid multi-scale ant colony system," *ISCIS*, pp. 440–448, 2004.

[9] M. Dorigo and L. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[10] D. Le, G. Thoma, and H. Weschler, "Automated page orientation and skew angle detection for binary document images," *Pattern Recognition*, vol. 27, no. 10, pp. 1325–1344, 1994.