

AN ITERATIVE DECODING APPROACH TO DOCUMENT IMAGE ANALYSIS

Taku A. Tokuyasu[†] and Philip A. Chou[‡]

[†]Computer Science Division, University of California, Berkeley, CA

[‡]Microsoft Research, Microsoft Corporation, Redmond, WA

ABSTRACT

We introduce an iterative approach to recognizing two-dimensional grammatical structure within digital images, which we term “turbo recognition.” Inspired by the success of turbo decoding for channel coding of one-dimensional sequences, we develop a recognition scheme for images based on two independent views of the same underlying message. These correspond to two independent image sources, one in the horizontal direction and the other in the vertical direction, which are driven by a single input message. The recognition process proceeds iteratively, first along one direction and then the other, applying the Forward/Backward algorithm to derive a new prior probability distribution on the input message for the orthogonal recognition step. This holds promise as a principled approach within the Document Image Decoding (DID) framework for the recognition of nontrivial 2D layout structure such as tables.

1. INTRODUCTION

Document Image Decoding (DID)[1] is a communication theory approach to document image recognition. It specifies a statistical model, shown schematically in Figure 1a, for the generation and recognition of digital images such as scanned documents. Image generation begins with an input message U ,

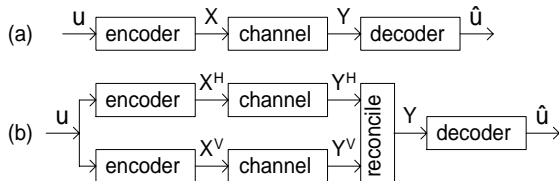


Figure 1: DID approach to document image recognition. (a) Traditional model. (b) Turbo model.

which is encoded as an ideal digital image $X = \text{encode}(U)$. This in turn is transmitted through a noisy channel $P(Y|X)$, resulting in an observed

image Y . Document recognition is then precisely defined as the recovery from Y of the message $\hat{U} = \text{decode}(Y)$ that minimizes the probability of error $P(\hat{U} \neq U)$. This message maximizes the posterior probability on input messages given the observed evidence, i.e. $\text{decode}(Y) = \arg \max_{U'} P(U'|Y)$.

In DID, the encoder is typically a finite-state transducer. When driven by the input message U , it emits the output symbols that form the ideal image[1]. The noisy channel can be taken to be simply bit-flip noise in the case of binary images. Algorithms familiar in the communication theory context, such as the Viterbi and Forward/Backward algorithm, are then used to recover the best estimate \hat{U} .

The most prominent success of DID to date has been in the area of optical character recognition (OCR). DID has exhibited an order of magnitude decrease in error rate, relative to commercial OCR systems, when applied to documents several hundred pages in length[2]. DID is competitive with conventional systems in speed as well, when optimized for the largely one-dimensional layout structure of text documents[2].

DID has had relatively less success when applied to problems of two-dimensional document layout. While it can perform (with the help of expert knowledge) complex layout analyses[1], an efficient general method within DID for decoding documents with even a simple two-column layout has yet to be developed. The difficulty stems ultimately from two problems. First, with a finite-state model of document production, it is difficult to model both horizontal and vertical constraints without a dramatic increase in the state space.¹ The second problem is that many two-dimensional structures have multiple interpretations. For example, a table can be read horizontally, as a sequence of columns, or vertically, as a sequence of rows. No single grammatical structure can provide both interpretations.

In this paper, we introduce a method that ex-

¹Context-free grammars can help in this regard[3, 4], but are still prohibitively expensive in practice.

tends the DID framework to deal effectively with certain kinds of two-dimensional layout. The method uses two finite-state transducers simultaneously — one horizontally and one vertically — to impose constraints in both directions. Unlike complicated two-dimensional models such as Markov random fields, the method largely retains the computational advantages of one-dimensional methods. However, decoding must now be performed by iterating between the horizontal and vertical constraints. Inspired by the analogy to turbo decoding (discussed below), we term the method “turbo recognition.”

2. TURBO RECOGNITION

Turbo coding for the communication problem was discovered in 1993 by Berrou et al.[5, 6]. Compared to state-of-the-art convolutional codes, turbo codes can achieve a far lower bit error rate on channels with a given signal-to-noise ratio (10^{-5} vs. 10^{-2} , at 1.7 dB), or conversely, reliable coding on channels with a far worse signal-to-noise ratio (bit error rate 10^{-5} at 1.7 dB — within 0.5 dB of the Shannon limit of 1.2 dB — vs. 4.0 dB). Turbo codes consist of two parallel convolutional codes (see e.g. [7]). The first convolutional code encodes the bit sequence as usual, while the second encodes a permutation of the original sequence. In a sense, the two convolutional codes take “orthogonal” views of the same data. In this way, error patterns which are difficult for one of the codes to correct may be easy for the other code to correct. For example, burst errors with respect to one code appear as isolated errors to another.

We apply this insight and the general methodology of turbo coding to the recognition of rectangular layout structures in document images. For this purpose, we assume that the images are approximately aligned on a rectangular grid.²

In turbo recognition of document images, the generation of an observed image is modeled by the process shown in Figure 1b, in which a single input (two-dimensional) message U is encoded by two one-dimensional finite-state transducers. The first transducer operates (independently) on each row of U , producing an overall ideal image X^H , while the second operates on each column of U , producing an overall ideal image X^V . The transposition of the image, horizontal to vertical, plays the role of the permutation in turbo coding. The ideal images X^H and X^V are passed through independent channels to produce the corrupted images Y^H and Y^V .

²Existing techniques should be sufficient to accomplish the necessary alignment.

These images are then deterministically reconciled to produce the single observed image Y , which is equal to Y^H (or Y^V) if $Y^H = Y^V$ and is otherwise equal to some null image.

Note that a valid input message U must now satisfy (i.e. be accepted by) two transducers instead of the usual one. Let L^H (L^V) denote the set of images whose rows (columns) all drive the horizontal (vertical) transducer into its accepting state(s). The prior distribution on U , $P(U|L^H, L^V)$, can then be considered as a distribution $P(U)$ restricted to the intersection of L^H and L^V .

The decoding problem is to find the message image \hat{U} maximizing the posterior distribution $P(U|Y, L^H, L^V)$. This is a hard problem in general, but if Y is not the null image (which of course is always the case in practice) then we have effectively observed Y^H and Y^V (since $Y^H = Y^V = Y$), so that $Y^H \rightarrow X^H \rightarrow U \rightarrow X^V \rightarrow Y^V$ is a Markov chain. Then the problem is to find the message image \hat{U} maximizing the posterior distribution $P(U|Y^H, Y^V, L^H, L^V)$ in

$$\begin{aligned}
& P(U|Y^H, Y^V, L^H, L^V)P(Y^V, L^V|Y^H, L^H) \\
&= P(U, Y^V, L^V|Y^H, L^H) \\
&= P(Y^V, L^V|U)P(U|Y^H, L^H) \\
&= \prod_j P(Y_j^V, L_j^V|U_j) \\
&\quad \times P(U_j|U_1, \dots, U_{j-1}, Y^H, L^H) \\
&= \prod_j P(Y_j^V, L_j^V|U_j) \\
&\quad \times \prod_i P(U_{i,j}|U_{i,1}, \dots, U_{i,j-1}, Y_i^H, L_i^H) \\
&\approx \prod_j P(Y_j^V, L_j^V|U_j) \\
&\quad \times \prod_i P(U_{i,j}|Y_i^H, L_i^H). \tag{1}
\end{aligned}$$

Here, j is a column index, U_j is the j th column of the image U , L_j^V is the event that U_j drives the vertical transducer into an accepting state, and Y_j^V is the j th column of the observed image Y^V . Likewise, i is a row index, U_i is the i th row of the image U , L_i^H is the event that U_i drives the horizontal transducer into an accepting state, and Y_i^H is the i th row of the observed image Y^H . Finally, $U_{i,j}$ is the (i, j) th pixel of the image U . Now, (1) is maximized over U by maximizing

$$P(Y_j^V, L_j^V|U_j) \prod_i P(U_{i,j}|Y_i^H, L_i^H)$$

independently for each column U_j of U . For this purpose, the product distribution $\prod_i P(U_{i,j}|Y_i^H, L_i^H)$

provides a prior on the column U_j . Each factor in this product is the marginal posterior distribution $P(U_{i,j}|Y_i^H, L_i^H)$, which can be found by the Forward/Backward algorithm through the horizontal trellis for row i .

The approximation in (1) is exact when $U_{i,j}$ is conditionally independent of $U_{i,1}, \dots, U_{i,j-1}$ given Y_i^H, L_i^H . This will be true if paths through the trellis beginning $U_{i,1}, \dots, U_{i,j-1}$ have probability one, i.e., if one path takes all the probability. Thus the approximation will be good if the posterior distribution of U given Y^H, L^H is sharply peaked. Intuitively, by feeding in the posterior distribution of the horizontal decoding as the prior distribution of the vertical decoding, the posterior distribution becomes more peaked. This leads to the iterative algorithm, in which the posterior distribution of the vertical decoding is then fed in as the prior distribution of the horizontal decoding, whereupon the process is repeated until convergence.

It should be noted that the convergence properties of the turbo decoding algorithm itself have yet to be fully understood[9, 10]. We find that three or four iterations suffice for small images (see below), and we are presently studying the behavior on larger images. The empirical success of turbo codes, however, encourages us to believe that similarly high performance can be achieved in general in the image recognition context.

3. EXAMPLE

Consider the set of all binary images X containing exactly one black rectangle. In the turbo recognition framework of Figure 1b, where the input message U is fed into two independent finite-state transducers, the rectangle can be described as the intersection of two orthogonal black stripes, represented by the images U^H and U^V shown in Figure 2a. Each row of U^H satisfies a simple one-dimensional grammar, consisting of a run of white pixels, followed by a run of black, followed by a run of white (and the same holds true for each column in the image U^V). Note that the black runs in different rows are not explicitly forced to be aligned. Instead, alignment is achieved implicitly by the fact that (valid) input messages U satisfy both the horizontal and vertical grammars simultaneously.

To effect this integration of horizontal and vertical constraints, take each input message pixel U_{ij} to be composed of the *pair* of binary components, (U_{ij}^H, U_{ij}^V) , where U_{ij}^H is 1 if and only if the i, j th pixel lies in the vertical stripe, and U_{ij}^V is 1 if and only if the i, j th pixel lies in the horizontal stripe.

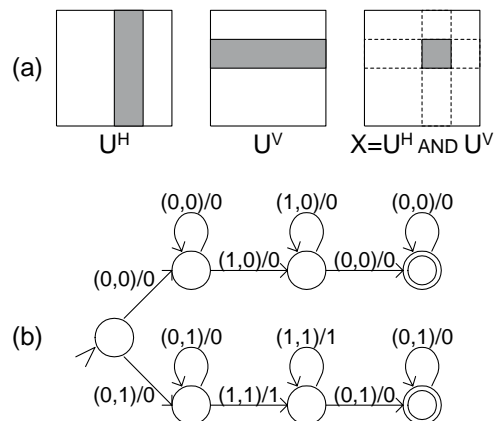


Figure 2: One rectangle: (a) input and ideal output images, and (b) horizontal transcoder.

The horizontal transcoder, e.g., which encodes such messages into ideal images by independently processing each row of U , is shown in Figure 2b.³ Each transition is labeled by the input symbol (pair) which drives the transition, and by the corresponding output message bit, where 0 is “non-printing” or white, and 1 is “printing” or black. As far as the U^H component of U is concerned, the two branches in Figure 2b are in fact identical.⁴ The top branch, corresponding to a row which misses the rectangle, is followed when the U^V component of U is 0. The bottom branch, corresponding to a row which hits the rectangle, is followed when the U^V component of U is 1. The transition outputs a 1 if and only if both input components are 1, which reflects the fact that the output image X is the intersection of the input component images U^H and U^V . To summarize, a valid input message U drives both the horizontal and vertical transcoders into accepting states, producing an ideal output image equal to a single black rectangle somewhere on the page. Given a noisy version Y of such an image, we can apply DID based on transcoders such as Figure 2b to recover the message U (and hence underlying black rectangle) that best explains the observed image.

This example can be generalized in a straightforward way to the case of, e.g., cells in a rectangular array, where each cell is itself represented by further set of horizontal and vertical transcoders. Matrices, or tables, of text or textures can be modeled in this way.

³The vertical transcoder is identical, with the input labels transposed.

⁴This is not the case in general.

4. RESULTS

To illustrate some of our preliminary results, we discuss another example, namely a checkerboard of white and black rectangles. The grammar in this case is similar to the one above, where the input symbol pairs corresponding to black output pixels are now (1,1) and (0,0).

The behavior of the algorithm in the presence of noise is illustrated in Figure 3. The ideal (out-

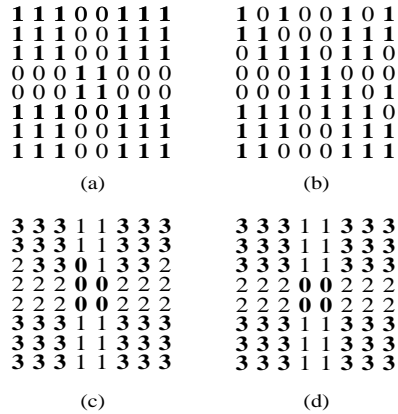


Figure 3: Turbo recognition on a checkerboard image. (a) Original image. (b) Corrupted image. (c) Decoded image after four iterations. (d) Decoded image after five iterations.

put) image in (a), corrupted by noise with bit-flip probability 0.2, is given in (b). After four iterations (with one horizontal and one vertical pass each) of the turbo recognition algorithm, the input message corresponding to the maximum marginal probability at each pixel is given in (c) (where 0 through 3 represent the input symbols (0,0) through (1,1) in the obvious way). This decoded message is in fact not a legal input message, a reflection of the fact that the maximum marginal (MM) message need not satisfy the grammar. As in turbo coding, we expect however that the MM and MAP estimates become identical as the posterior distributions become more peaked. Indeed, an additional iteration in the present example produces a perfect reconstruction (d) of the original input message. We expect further refinements of the iterative method in the future.

While the examples we have presented are admittedly quite simple, decoding on small low-resolution images may be sufficient for layout segmentation of e.g., two-column text layouts. As mentioned previously, other relevant examples, such as tables, are also readily envisioned.

5. SUMMARY

We presented a framework, a simple example, and preliminary results of document image layout analyses using iterative decoding, which we call “turbo recognition” by analogy with the breakthrough channel coding technique called turbo coding. Preliminary results indicate that the method can be used to recognize grammatical images in extreme amounts of noise. The method appears to be ideally suited to recognizing images with both horizontal and vertical structure, such as tables, arrays, matrices, and multicolumn formats.

6. REFERENCES

- [1] G. E. Kopec and P. A. Chou. Document image decoding using Markov sources. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(6):602–617, June 1994.
- [2] G. E. Kopec. Document image decoding in the UC Berkeley digital library. *Document Recognition III*, Proc. of the SPIE, 1996, vol.2660:2-13.
- [3] P. A. Chou. Recognition of equations using a two-dimensional stochastic context-free grammar. In *Visual Communications and Image Processing*, pages 852–863, Philadelphia, PA, November 1989. SPIE.
- [4] J. F. Hull. Recognition of mathematics using a two-dimensional trainable context-free grammar. Master’s thesis, MIT, Cambridge, MA, June 1996.
- [5] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo codes. In *Proc. Int’l Communications Conf.* IEEE, 1993.
- [6] C. Berrou and A. Glavieux. Near optimum error-correcting coding and decoding: Turbo codes. *IEEE Trans. Communications*, 44(10):1261–1271, October 1996.
- [7] Brendan J. Frey. *Graphical models for machine learning and digital communication* The MIT Press, Cambridge, MA, c1998.
- [8] R. McEliece, E. Rodemich, and J. Cheng. The Turbo decision algorithm. In *Proc. 33rd Allerton Conf. on Communications, Control, and Computing*, pages 366–379, Monticello, IL, 1993. IEEE.
- [9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.
- [10] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 1998. Submitted.