Document Image Decoding by Heuristic Search

Anthony C. Kam and Gary E. Kopec, *Member, IEEE*

Abstract

This correspondence describes an approach to reducing the computational cost of document image decoding by viewing it as a heuristic search problem. The kernel of the approach is a modified dynamic programming (DP) algorithm, called the *iterated complete path* (ICP) algorithm, that is intended for use with separable source models. A set of heuristic functions are presented for decoding formatted text with ICP. Speedups of 3–25 over DP have been observed when decoding text columns and telephone yellow pages using ICP and the proposed heuristics.

I. Introduction

Document image decoding (DID) is an approach to document recognition that is based on an explicit communication theory view of the processes of document creation, transmission and recognition [2]. In the DID model, a stochastic *message source* selects a finite string M from a set of candidate strings according to a prior probability distribution. An *imager* converts the message into an ideal binary image Q. A *channel* maps the ideal image into an observed image Z by introducing distortions due to printing and scanning, such as skew, blur and additive noise. Finally, a *decoder* receives image Z and produces an estimate M of the original message according to a maximum a posteriori (MAP) decision criterion.

Much of the recent work in DID has focussed on a class of combined source/imager models called separable Markov sources [4]. Loosely, a separable source is one that may be factored into a product of one-dimensional models that represent horizontal and vertical structure, respectively. MAP decoding of an image with respect to a separable model can be implemented using a nested dynamic programming (DP) algorithm called the *separable Viterbi algorithm* [4].

The time complexity of separable Viterbi decoding is $\mathbf{O}(B_h \times H \times W)$, where B_h is the number of branches in the horizontal models and H and W are the image height and width, respectively, in pixels. The factor $B_h \times W$ represents the cost of using the horizontal models to decode a single image row while the factor H arises because decoding is repeated at every row. Although the computation grows only linearly with the number of image pixels, in absolute terms it can be prohibitive. For example, decoding an 8.5in $\times$ 11in image scanned at 300 ppi using a simple text column model requires about 40 *minutes*. Thus, methods for decreasing the required computation are essential if DID is to become a widely-used approach to document image analysis.

The basic source of complexity in image decoding is the need to decode every image row. If the horizontal models were run only along the actual text baselines, the cost would decrease by the factor $\frac{H}{L}$, where L is the number of text lines. For example, with $H = 3300$ (11in $\times$ 300 ppi) and $L = 50$, the time to decode a simple text page would drop to about 36 *seconds*.

Conventional line segmentation algorithms could be used to locate text lines prior to image decoding, in a manner analogous to their use with recent HMM-based approaches to text recognition [5], [1]. However, pre-decoding segmentation can be unreliable, particularly for highly degraded images. Moreover, since conventional segmentation algorithms are not based on a rigorous probabilistic formulation their use would negate many of the theoretical advantages of DID.

This correspondence describes a heuristic search approach to image decoding that aims to reduce computational cost without sacrificing the optimality of DP. The kernel of the approach is a modified DP algorithm, called the *iterated complete path* (ICP) algorithm that replaces full decoding of most image rows with computation of a simple upper bound, or heuristic, on the decoding score. Three types of heuristics are presented that are useful for separable models of text-like images. One type corresponds to the common use of horizontal pixel projection to locate text lines. The second heuristic bounds the score for a given row in terms of the score of an adjacent row. The third heuristic is a bound computed by decoding a reduced resolution image.

The rest of this correspondence is organized as follows. Section II briefly reviews image decoding using separable models. Section III describes the iterated complete path algorithm. Section IV develops the concept of a decoding heuristic and defines the horizontal projection, adjacent row and reduction heuristics. Section V describes a set of experiments using ICP and the proposed heuristics to decode columns of text and telephone yellow pages.

II. Decoding Using Separable Models

Fig. 1 shows the structure of a simple class of separable sources. The sources in Fig. 1 are very similar to the pseudo 2-d hidden Markov models (PHMMs) described in [5]. Thus, ICP and the heuristics presented below are applicable to PHMMs as well. Extension to the general class of separable models defined in [4] is straightforward.

A separable source is a collection of named subsources $G_0, G_1 \ldots G_K$. Each subsource is a directed graph consisting of a finite set of states (nodes, vertices) and a set of directed transitions (branches, edges). Two distinguished states of a subsource are the initial state n_I and the final state n_F. Each transition t connects a pair of states, L_t and R_t, that are called, respectively, the *predecessor* (left) state and the *successor* (right) state of t. A *complete path* π through

a Markov source is a sequence of transitions $t_1 \ldots t_P$ where $L_{t_1} = n_I$, $R_{t_P} = n_F$ and $R_{t_i} = L_{t_{i+1}}$ for $i = 1 \ldots P-1$.[1] One of the subsources is designated as the top-level subsource and is labeled G_0. The top-level subsource represents the overall vertical structure of a class of document images. Each of the remaining subsources represents a particular type of embedded horizontal structure.

Each transition of a horizontal subsource is labeled with a 4-tuple of attributes, $(Q_t, m_t, a_t, \Delta x_t)$, where Q_t is a binary bitmap *template*, m_t is a *message* character string, a_t is the *transition probability* and Δx_t is the horizontal *displacement* of t. The template or message string may be null, i.e. effectively omitted. A complete path through a horizontal subsource defines a composite message, $M_\pi = m_{t_1} \cdots m_{t_P}$, formed by concatenating the message strings of the transitions of the path. Also associated with each path π is a sequence of horizontal positions $x_0 \ldots x_P$ recursively defined by $x_i = x_{i-1} + \Delta x_{t_i}$, where x_0 is an initial position, normally 0.[2] In a similar way, each transition of the vertical subsource G_0 is labeled with a transition probability a_t, a vertical displacement Δy_t and an optional horizontal subsource name S_t. A path π_0 through G_0 defines a sequence of vertical positions $y_0 \ldots y_P$ and specifies a sequence of horizontal subsources $S_{t_1} \ldots S_{t_P}$.

MAP decoding of an observed image Z with respect to a separable source involves finding a complete path $\hat{\pi}_0$ in G_0 that maximizes the likelihood function

$$\mathcal{F}(\pi_0) \equiv \sum_{i=1}^{P} [\mathcal{F}(t_i; y_i) + \log a_{t_i}] \tag{1}$$

where $\mathcal{F}(t; y)$ is the result of decoding Z along row y using subsource S_t. Formally,

$$\mathcal{F}(t; y) = \max_{\pi} \sum_{i=1}^{P} [\mathcal{L}(Z \mid Q_{t_i}[x_{i-1}, y]) + \log a_{t_i}] \tag{2}$$

where the maximization is taken over complete paths through S_t. The term of the form $\mathcal{L}(Z \mid Q_t[x, y])$ in (2) is the template match score between Z and template Q_t aligned with its origin at (x, y) [2]. As discussed in [4], decoding can be implemented by using DP to compute $\mathcal{F}(t; y)$ for each y and subsource S_t and then using DP to find $\hat{\Pi}_0$. The cost of finding $\hat{\Pi}_0$ is typically insignificant compared to the cost of computing all of the values of $\mathcal{F}(t; y)$. Thus, as noted above, the overall computational complexity is $\mathbf{O}(B_h \times H \times W)$, where B_h is the total number of branches in the horizontal subsources.

III. Iterated Complete Path Algorithm

Viewing image decoding as a heuristic search problem is motivated by the observation that only the values of $\mathcal{F}(t; y)$ corresponding to transitions on the best path $\hat{\Pi}_0$ are actually relevant to decoding; the remaining values are computed essentially to verify that $\hat{\Pi}_0$ is indeed the best path. The basic strategy of heuristic image decoding is to replace most evaluations of $\mathcal{F}(t; y)$ with computations of a simple upper bound function, called a *heuristic*. While any of the well-known heuristic search algorithms, such as A* [7] might be applied to this problem, we will discuss heuristic decoding using a simple modification to DP, called the *iterated complete path* (ICP) algorithm. The structure of ICP makes it well-suited for separable models and the simplicity of the algorithm makes it convenient for exposition.

The basic ICP algorithm is founded on the following lemma. Suppose that $\mathcal{U}(t; y)$ is a function that upper bounds $\mathcal{F}(t; y)$, i.e., $\mathcal{U}(t; y) \geq \mathcal{F}(t; y)$ for each t and y. For each path Π_0 let $\mathcal{U}(\Pi_0)$ be defined by $\mathcal{U}(\Pi_0) = \sum_{i=1}^{P} [\mathcal{U}(t_i; y_i) + \log a_{t_i}]$ by analogy with (1). Suppose that $\hat{\Pi}_0$ is a complete path in G_0 that maximizes $\mathcal{U}(\Pi_0)$, i.e. $\mathcal{U}(\hat{\Pi}_0) \geq \mathcal{U}(\Pi_0)$ for every Π_0. Then, if $\mathcal{U}(\hat{t}_i; \hat{y}_i) = \mathcal{F}(\hat{t}_i; \hat{y}_i)$ for the transitions of $\hat{\Pi}_0$ it is simple to show that $\hat{\Pi}_0$ maximizes $\mathcal{F}(\Pi_0)$.

Proof: Note that if $\mathcal{U}(t; y) \geq \mathcal{F}(t; y)$ for the transitions of some path Π_0 then $\mathcal{U}(\Pi_0) \geq \mathcal{F}(\Pi_0)$. Similarly, $\mathcal{U}(\Pi_0) = \mathcal{F}(\Pi_0)$ if $\mathcal{U}(t; y) = \mathcal{F}(t; y)$. Thus, given the assumptions, $\mathcal{F}(\hat{\Pi}_0) = \mathcal{U}(\hat{\Pi}_0) \geq \mathcal{U}(\Pi_0) \geq \mathcal{F}(\Pi_0)$. □

ICP finds a sequence of complete paths that maximize a sequence of $\mathcal{U}$ functions. Initially, $\mathcal{U}(t; y)$ is given by an upper bound function $\mathcal{H}(t; y)$, the heuristic, that is assumed to be computationally much less expensive to evaluate than $\mathcal{F}(t; y)$. As ICP proceeds, $\mathcal{U}$ is refined by replacing some of the $\mathcal{H}(t; y)$ values by values of $\mathcal{F}(t; y)$ that are computed by actually decoding image rows. ICP terminates when $\mathcal{U}(\hat{t}_i; \hat{y}_i) = \mathcal{F}(\hat{t}_i; \hat{y}_i)$ for each transition of $\hat{\Pi}_0$.

The basic ICP procedure is shown in Fig. 2. The inputs to ICP are the top-level subsource G_0, a procedure that can be invoked to compute $\mathcal{F}(t; y)$ for any t and y, and a routine that computes $\mathcal{H}(t; y)$. The ICP procedure maintains two internal data arrays indexed by $(t; y)$. The elements of array $\mathcal{U}$ are initialized with the values of $\mathcal{H}$ before the iteration begins. As noted above, some of the elements of $\mathcal{U}$ are updated with actual values of $\mathcal{F}$ during the course of the iteration.

[1] All complete paths through a source do not necessarily have the same length. However, for notational simplicity the dependence of P on Π will not be indicated.

[2] This correspondence uses an image coordinate system in which x increases to the right, y increases downward, and the upper left corner is at $x = y = 0$.

Each element of array $\mathcal{A}$ is a boolean flag that indicates whether the corresponding element of $\mathcal{U}$ is an upperbound or actual score; that is, $\mathcal{A}(t;y) = true$ if $\mathcal{U}(t;y) = \mathcal{F}(t;y)$ and $\mathcal{A}(t;y) = false$ otherwise.

During each pass of the iteration, a path $\hat{\Pi}_0$ that maximizes $\mathcal{U}(\Pi_0)$ is computed by DP. For each transition in $\hat{\Pi}_0$, if array element $\mathcal{U}(\hat{t}_i;\hat{y}_i)$ is an upperbound score it is replaced by $\mathcal{F}(\hat{t}_i;\hat{y}_i)$ and $\mathcal{A}(\hat{t}_i;\hat{y}_i)$ is updated. The iteration continues until $\mathcal{U}(\hat{t}_i;\hat{y}_i)$ is an actual score for each $\hat{t}_i$ of $\hat{\Pi}_0$.

The previous lemma guarantees that the path returned by ICP maximizes the likelihood function if $\mathcal{H}(t;y) \geq \mathcal{F}(t;y)$ for all t and y. In that case, ICP is an example of an *admissible* [7] search algorithm. Admissibility is also guaranteed if the heuristic satisfies a significantly weaker condition, that $\mathcal{H}(\hat{t}_i;\hat{y}_i) \geq \mathcal{F}(\hat{t}_i;\hat{y}_i)$ for the transitions of *some* best path $\hat{\Pi}_0$. The significance of this condition, called the *weak A** criterion [3], is that if we assume that $\hat{\Pi}_0$ is the path that actually generated the image (i.e. that the decoder gives the correct answer) then it is only necessary to consider the behaviour of the heuristic on image regions generated by the corresponding subsource. The utility of weak A* will become apparent in the next section.

IV. Document Decoding Heuristics

A heuristic for a horizontal subsource is a scalar function $\mathcal{H}(t;y)$ that upperbounds the actual score $\mathcal{F}(t;y)$. Typically, $\mathcal{H}(t;y)$ is defined in terms of some measurement performed on the observed image Z in the vicinity of row y. In discussing heuristics, it is convenient to distinguish between the upper bound $\mathcal{H}(t;y)$ and the measurement on which it is based, which we will denote $\mathcal{M}(t;y)$. The functions $\mathcal{M}(t;y)$ will be called *heuristic measurements* to distinguish them from the heuristics themselves.

We have explored three types of heuristic measurements that appear useful for decoding images of formatted text. The *weighted projection* heuristic $\mathcal{H}_{wp}$ is an upper bound on $\mathcal{F}(t;y)$ in terms of the horizontal projection profile $\vec{z}$, where z_i is the number of 1's in row i of the observed image Z. Formally, $\mathcal{M}_{wp}(t;y) = \sum_i h_i z_{y+i}$ where $\vec{h}$ is a subsource-dependent vector of non-negative constants whose sum is unity. We have found that a good choice for $\vec{h}$ is the mean horizontal projection for images generated by the subsource. With this choice of $\vec{h}$, the weighted projection can be interpretted as the output of a matched filter for the subsource profile.

The *adjacent row* heuristic $\mathcal{H}_{ar}$ is an upper bound on $\mathcal{F}(t;y)$ in terms of $\mathcal{M}_{ar}(t;y) = \mathcal{F}(t;y+j)$ where $j = \pm 1$. The adjacent row heuristic formalizes the observation that $\mathcal{F}(t;y)$ normally does not change much from one row to the next. It is applied at the end of each pass in ICP to update entries in $\mathcal{U}$ that are adjacent to newly-computed values of $\mathcal{F}(\hat{t}_i;\hat{y}_i)$. Note that use of $\mathcal{M}_{ar}(t;y)$ entails a minor modification to the basic ICP algorithm of Fig. 2.

The *reduction* heuristic $\mathcal{H}_{red}$ is an upper bound on $\mathcal{F}(t;y)$ in terms of $\mathcal{M}_{red}(t;y) = \max\{\mathcal{F}'(t';y'+j) \mid j = -1, 0, 1\}$ where $y' = \text{round}(y/4)$ and $\mathcal{F}'(t';y')$ is computed by decoding a 4×4 reduction of the input image with a scaled version of S_t. The reduction measurement maximizes $\mathcal{F}'(t';y')$ over a three row neighborhood of the nominal location of row y in the reduced image in order to desensitize the measurement to variations in the alignment of the image with respect to the downsampling grid. The scaled model is derived by reducing each template image and scaling all displacements and font metrics.

The central issue in constructing a heuristic from a heuristic measurement is the relationship between $\mathcal{M}(t;y)$ and $\mathcal{F}(t;y)$. This relationship has been studied in depth for the types of image source and channel models currently used in DID and a general class of heuristics that includes the three defined above [3]. Since the cited reference is not widely available, we will informally summarize the results in the context of a specific example— the weighted projection heuristic for one of the subsources of the telephone yellow page model described in [2], [4], called the ***standard name line***. As will be seen, the main practical consequence of the analysis is a simple procedure for constructing piecewise linear heuristics from scatter plots of $\mathcal{F}(t;y)$ and $\mathcal{M}(t;y)$ values.

Fig. 3 shows a small portion of a yellow page column image [6]. The lines "Communications 2001" and "Davis Communication Services" are examples of standard name lines. Fig. 4 shows the projection weight vector for the ***standard name line***. This profile was computed from a set of 605 lines extracted from a dataset of 48 yellow page columns [2]. Fig. 5 shows a scatter plot of $\mathcal{F}(t;y)$ versus $\mathcal{M}_{wp}(t;y)$ computed at the locations of the ***standard name line*** samples. The tight clustering around an approximate straight line is typical of the observed relationship between heuristic measurements and actual scores. A theoretical explanation for this linearity is one of the main contributions of the analysis in [3].

The analysis begins with the observation that heuristic measurements $\mathcal{M}(t;y)$ and actual scores $\mathcal{F}(t;y)$ can be modeled as sums of random variables associated with the transitions of complete paths through the image source. The randomness arises from the stochastic nature of the source, which induces a probability distribution on complete paths, as well as the channel, which induces a joint distribution on $\mathcal{M}(t;y)$ and $\mathcal{F}(t;y)$ given a particular complete path. The problem is to understand the conditional distribution of $\mathcal{F}(t;y)$ given an observed value of $\mathcal{M}(t;y)$.

The analysis is simplified by assuming asymptotic behaviour, in which the paths are long and $\mathcal{M}(t;y)$ and $\mathcal{F}(t;y)$ are sums of large numbers of variables. In that case, $\mathcal{M}(t;y)$ and $\mathcal{F}(t;y)$ are approximately Gaussian and the problem reduces to finding the mean and variance of $\mathcal{F}(t;y)$ given an observed value of $\mathcal{M}(t;y)$. The main result is that, with

these assumptions (plus a few others) $\mathcal{E}[\mathcal{F} \mid \mathcal{M} = M] \approx \beta M + \kappa_\beta$ and $Var[\mathcal{F} \mid \mathcal{M} = M] \approx \gamma M + \kappa_\gamma$, where β, κ_β, γ and κ_γ are constants that can be computed from the model and channel parameters. The linear relationship between $\mathcal{M}(t;y)$ and the mean of $\mathcal{F}(t;y)$ is the source of the linear trend in Fig. 5.

The asymptotic conditional Gaussian distribution of $\mathcal{F}(t;y)$ given $\mathcal{M}(t;y)$ implies that any function of $\mathcal{M}(t;y)$ that is guaranteed to be an upper bound on $\mathcal{F}(t;y)$ is likely to be so loose a bound as to be ineffective as a heuristic. Thus, it is necessary to consider *probabilistic* upper bounds, i.e. functions of $\mathcal{M}(t;y)$ that bound $\mathcal{F}(t;y)$ with some probability η (called the *bounding factor*), accepting the consequence that ICP based on such bounds will be admissible only with some probability. One simple probabilistic bound is to take the conditional mean of $\mathcal{F}(t;y)$ plus some number of standard deviations. Since the conditional mean of $\mathcal{F}(t;y)$ is linear in $\mathcal{M}(t;y)$ and the standard deviation grows with the square root of $\mathcal{M}(t;y)$ a related bound is to make $\mathcal{H}(t;y)$ a linear function of $\mathcal{M}(t;y)$. We have it found it effective to generalize this to a piecewise linear relationship and to determine the parameters empiricially from the scatter plots.

The dashed polyline in Fig. 5 is a five-segment piecewise linear definition of $\mathcal{H}_{wp}(t;y)$ in terms of $\mathcal{M}_{wp}(t;y)$. The central diagonal segment is designed to lie above 95% of the scatter points (i.e. $\eta = .95$). The exact slope of the central segment is not critical, although experience suggests that the heuristic is more effective if the underbounded values of $\mathcal{F}(t;y)$ fall in the lower range. As indicated in Fig. 5, this can be accomplished by making the slope of the heuristic greater than the "slope" of the data cluster. The horizontal segments near scores of -1000 and $+10000$ cause $\mathcal{H}_{wp}(t;y)$ to saturate at values that bracket the minimum and maximum values of $\mathcal{F}(t;y)$ observed in the sample data. Finally, the value of $\mathcal{H}_{wp}(t;y)$ is $-\infty$ when $\mathcal{M}_{wp}(t;y)$ falls outside the range of heuristic measurements observed in the data. Note that defining $\mathcal{H}_{wp}(t;y)$ using only data collected at actual ***standard name line*** locations is justified by the weak A* criterion.

Fig. 6 compares $\mathcal{H}_{wp}(t;y)$ and $\mathcal{F}(t;y)$ for the image of Fig. 3. The weighted projection measurement $\mathcal{M}_{wp}(t;y)$ in the region of the reverse-video subject heading is much larger than the maximum projection for standard name lines defined in Fig. 5. As a result, over most of the heading $\mathcal{H}_{wp}(t;y) = -\infty$ and the heuristic does not upperbound the actual score.[3] However, $\mathcal{H}_{wp}(t;y) \geq \mathcal{F}(t;y)$ at the locations of the actual standard name lines ("Communications 2001" at $y = 348$ and "Davis Communication Services" at $y = 448$). The previous discusion of the weak A* condition implies that $\mathcal{H}_{wp}(t;y)$ will support admissible search in this example.

V. Experimental Results

We have conducted a set of experiments to compare the decoding times and recognition accuracies of ICP with dynamic programming. The three heuristics defined above were used, combined into a single heuristic by taking their minimum. The parameters of the piecewise-linear heuristic functions were obtained from scatter plots as described above. The bounding factor was $\eta = 0.95$ except where noted.

Four image models were used. Two of the models were the simple text column and the telephone yellow page column models described in [2], [4]. The remaining two models were derived from these two by replacing each transition associated with a non-null template with the subnetwork shown in Fig. 7. This transformation is motivated by the observation that the individual characters in a scanned text line frequently lie above or below the nominal baseline by one or two pixels. The three template branches in Fig. 7 correspond to three vertical displacements with respect to the nominal text baseline. This modification leads to a significant improvement in recognition accuracy and is used routinely in DID. We will refer to the modified models as "jittered" or having jitter 1; the original models have jitter 0.

Jittered models represent an interesting case for ICP because computing $\mathcal{F}(t;y)$ for a given y requires computation of template match scores $\mathcal{L}(Z \mid Q)$ for the three image rows at $y-1$, y, and $y+1$. Note that two of the three rows of $\mathcal{L}(Z \mid Q)$ values required to decode row y are also required to decode rows $y+1$ and $y-1$. In the case of the separable Viterbi algorithm, if computation of $\mathcal{F}(t;y)$ proceeds row-wise from top to bottom, it is straightforward to buffer the $\mathcal{L}(Z \mid Q)$ values so that only one row of new values is computed for each row decoded. Moreover, only a bounded amount of storage (three rows) is required to fully exploit the potential sharing. On the other hand, ICP typically decodes a set of isolated lines during each pass. Moreover, lines decoded during successive passes need not be adjacent. Thus, either a potentially large amount of storage will be required to retain $\mathcal{L}(Z \mid Q)$ values or some values might be computed more than once.

We have obtained satisfactory results using a simple eager evaluation strategy in which $\mathcal{F}(t;y-1)$, $\mathcal{F}(t;y)$ and $\mathcal{F}(t;y+1)$ are computed whenever $\mathcal{F}(t;y)$ is required but no template match scores are retained between ICP passes. With this approach, 5 rows of $\mathcal{L}(Z \mid Q)$ are computed for every three rows decoded.

The text column data set consisted of ten 8.5in $\times$ 11in pages of random text in 12 pt Adobe Times-Roman. Each page contained 45 lines of 70 characters each; the entire dataset contained 31000 characters. The pages were printed and scanned at 300 ppi. Following scanning, the images were cropped slightly to remove some of the margin whitespace. The yellow page column data set consisted of 48 columns extracted from 10 pages scanned at 300 ppi [2]. The dimensions of a typical column were $W = 600$ and $H = 3000$. The yellow page dataset contained 1057 constituents (i.e. ***standard***,

[3] $-\infty$ is represented as -100 in Fig. 6.

bold, ***subject***, etc.).

Table I summarizes the decoding times and accuracies of ICP and the separable Viterbi algorithm for the four image models. The indicated CPU times are the mean decoding times per image; the images were decoded on a 50 MHz SUN Sparc-10. Text recognition accuracy ("Message % correct") was computed by aligning the decoded and true text strings using dynamic programming and dividing the number of matches by the length of the correct string [2]. Yellow page decoding accuracy was computed similarly, except that constituents were counted rather than individual characters [2].

The "Path admiss." column gives a measure of the admissibility of ICP as the fraction of transitions in the top-level Viterbi best path $\hat{\Pi}_0$ that also occur in the ICP best path. Only transitions labeled with a horizontal subsource are included. This statistic is a conservative measure of the match between ICP and Viterbi algorithm outputs since it includes transitions that do not contribute to the output message.

The column labeled "Rows decoded" contains the average number of values of $\mathcal{F}(t;y)$ computed in decoding each image. These counts include only transitions through printing subsources [4] (those with templates on one or more branches), since the computation of $\mathcal{F}(t;y)$ for non-printing subsources is relatively costless. In the case of the separable Viterbi algorithm, the number of rows decoded is always the image height times the number of printing subsources.

The four models are listed in Table I in order of increasing complexity. As can be seen, the speedup advantage of ICP ranges from a factor of 25 for the unjittered text model to a factor of 3.22 for the jittered yellow page column. In all cases, the decoded message accuracy of ICP is almost indistinguishable from that of the Viterbi algorithm. Furthermore, the path admissibility of ICP is about .99 for three of the four models, even though the heuristics were designed with $\eta = .95$.

The large speedup in the case of the text column reflects the fact that only about three values of $\mathcal{F}(t;y)$ are computed for each actual text line. For example, ICP decodes 144 image rows to recognize 45 text lines using the jittered model. The text model contains only a single printing subsource plus a carriage return. Thus, the function of the heuristics is essentially to perform a two-way discrimination of text from whitespace. As the data suggest, this can be done very efficiently.

The yellow page model, on the other hand, contains 19 different printing subsources. Many of these are similar and it is unrealistic to expect simple heuristics to effectively discriminate among them. Thus, it is not surprising that, for a given row y, $\mathcal{F}(t;y)$ will be computed for several subsources if it is computed for one. Indeed, the best path for a yellow page column contains 264 printing transitions on average. From Table I it can be seen that the number of rows decoded is about 18 times this (4641) for the unjittered model and 26 times (6799) for the jittered model.

Table II shows the effect of varying the bounding factor η on the timing and performance of ICP in decoding yellow pages using the jittered model. As indicated, decoding time decreases by about a factor of two as η is lowered from a very conservative value (.99) to one for which the heuristics underbound 20% of the time. Decoder accuracy is essentially the same for all $\eta \geq .95$ and falls off for smaller values of η. This motivates our choice of $\eta = .95$ as the default value.

VI. Final Remarks

This correspondence has presented a formulation of document image decoding as a heuristic search problem. An important element of the proposed decoding method is the use of simple heuristic measurements to provide upper bounds on decoding scores. One of the heuristics that we have investigated, the weighted projection, is patterned after a common technique for segmenting text lines from background whitespace. While ICP does not perform explicit segmentation, our use of projections may be viewed as corresponding to a "soft" segmentation in which image regions are quantitatively rated as candidate text lines. From this perspective, the projection heuristic is simply a reformulation of common practice into the rigorous framework of the DID paradigm. It is an interesting question whether other aspects of conventional document recognition art can be similarly recast.

References

[1] F. Chen, D. Bloomberg and L. Wilcox, "Spotting phrases in lines of imaged text", in *Document Recognition II*, L. Vincent and H. Baird, editors, Proc. SPIE vol. 2422, pp. 256–269, 1995.

[2] G. Kopec and P. Chou, "Document image decoding using Markov source models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, June, 1994, pp. 602–617.

[3] A. Kam, *Heuristic document image decoding using separable Markov models*", M.I.T. Master of Science thesis, June, 1993.

[4] A. Kam and G. Kopec, "Separable source models for document image decoding", in *Document Recognition II*, L. Vincent and H. Baird, editors, Proc. SPIE vol. 2422, pp. 84–97, 1995.

[5] S.-S. Kuo and O. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-d hidden Markov models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, Aug., 1994, pp. 842–848.

[6] Pacific Bell, *Smart Yellow Pages, Palo Alto, Redwood City and Menlo Park*, 1992.

[7] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, 1984, Reading, MA: Addison-Wesley Publishing Co.

Index Terms
document image decoding, Markov models, heuristic search, dynamic programming

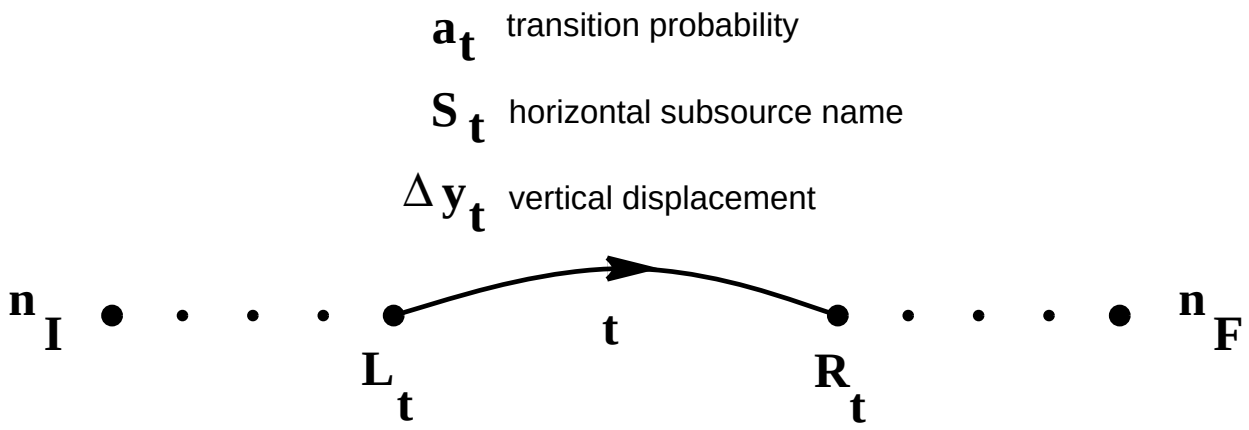
List of Figures

List of Tables

List of Footnotes

- A. C. Kam is a staff member at Caliper Corporation. Work performed while a graduate student at the Massachusetts Institute of Technology.
- G. E. Kopec is a Principal Scientist at the Xerox Palo Alto Research Center.

1. All complete paths through a source do not necessarily have the same length. However, for notational simplicity the dependence of P on Π will not be indicated.
2. This correspondence uses an image coordinate system in which x increases to the right, y increases downward, and the upper left corner is at $x = y = 0$.
3. $-\infty$ is represented as -100 in Fig. 6.



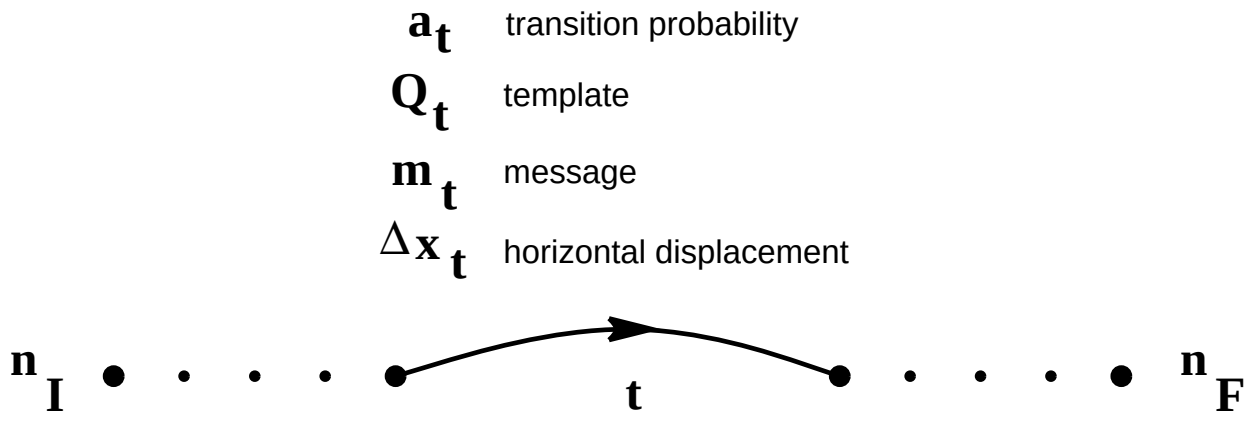


Fig. 1. Separable Markov image source. (a) Top-level (vertical) subsource G_0. (b) A horizontal subsource.

procedure $(G_0, \mathcal{F}, \mathcal{H}, H)$ ***do begin***
  for $t \in G_0,\ y \in [0, H]$ ***do begin***
    $\mathcal{U}(t; y) := \mathcal{H}(t; y)$
    $\mathcal{A}(t; y) := false$
  end
  $\hat{\Pi}_0 := \arg\max_{\Pi_0} \mathcal{U}(\Pi_0)$ by DP
  $\mathcal{Q} := \{i \mid \mathcal{A}(\hat{t}_i; \hat{y}_i) = false\}$
  while $\mathcal{Q} \neq \emptyset$ ***do begin***
    for $i \in \mathcal{Q}$ ***do begin***
      $\mathcal{U}(\hat{t}_i; \hat{y}_i) := \mathcal{F}(\hat{t}_i; \hat{y}_i)$
      $\mathcal{A}(\hat{t}_i; \hat{y}_i) := true$
    end
    $\hat{\Pi}_0 := \arg\max_{\Pi_0} \mathcal{U}(\Pi_0)$ by DP
    $\mathcal{Q} := \{i \mid \mathcal{A}(\hat{t}_i; \hat{y}_i) = false\}$
  end
  return $(\hat{\Pi}_0)$
end

Fig. 2. The basic iterated complete path (ICP) algorithm.

Model type	Model jitter	Decoder type	Iterations	Rows decoded	CPU Time mins.	CPU Time speedup	Message % correct	Path admiss.
Text	0	viterbi	—	3242	37.0	—	96.6	—
Text	0	ICP	9	114	1.48	25.0	96.5	.991
Text	1	viterbi	—	3242	43.9	—	99.1	—
Text	1	ICP	6	144	3.34	13.1	99.1	.993
YP	0	viterbi	—	53357	27.6	—	96.2	—
YP	0	ICP	37	4641	5.39	5.12	96.0	.974
YP	1	viterbi	—	53357	30.2	—	99.5	—
YP	1	ICP	21	6799	9.39	3.22	99.4	.988

TABLE I

VARIOUS MEASURES OF DECODING TIME AND ACCURACY FOR THE ICP AND VITERBI ALGORITHMS FOR FOUR IMAGE MODELS.

Telecommunications-Telephone Equipment & Systems-Service & Repair

APEX COMMUNICATIONS INC
254 San Geronimo
Sunnyvale **408 773 9600**
Communications 2001
Voice Mail System
433 Airport Bl Burl 347 1500
Data-Tel SMto 349 6010
Davis Communication Services
683 Crane Av FosterCty 341 0214
G'raffe Communications Inc 594 9196

Fig. 3. Small portion of a typical yellow page column.

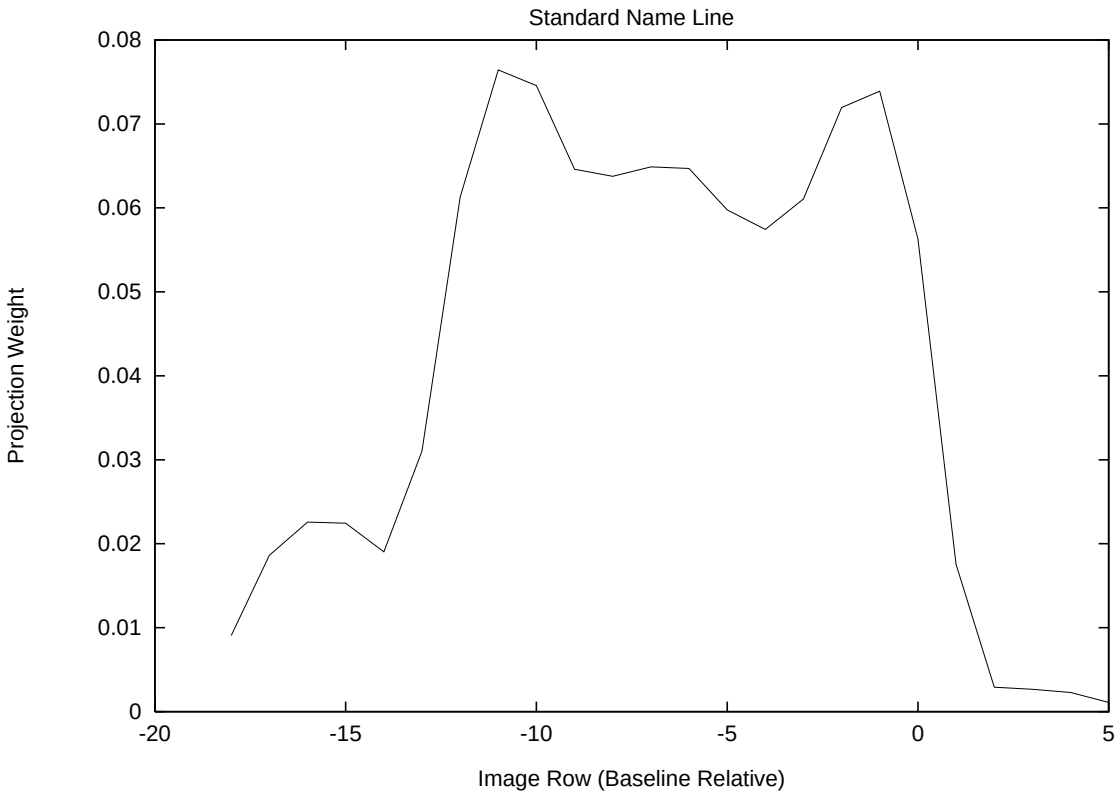


Fig. 4. Projection weight vector $\vec{h}$ for ***standard name line***.

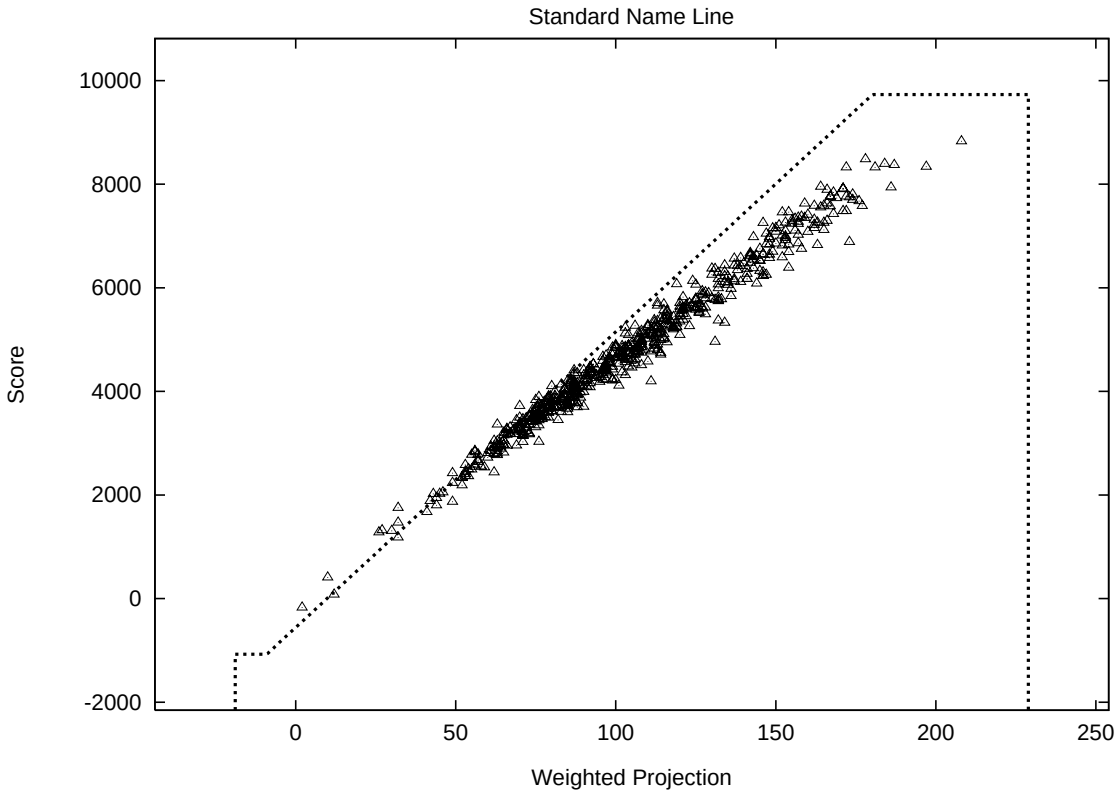


Fig. 5. Scatter plot of actual score $\mathcal{F}(t;y)$ versus weighted projection heuristic measurement $\mathcal{M}_{wp}(t;y)$. The dashed polyline defines the weighted projection heuristic $\mathcal{H}_{wp}(t;y)$.

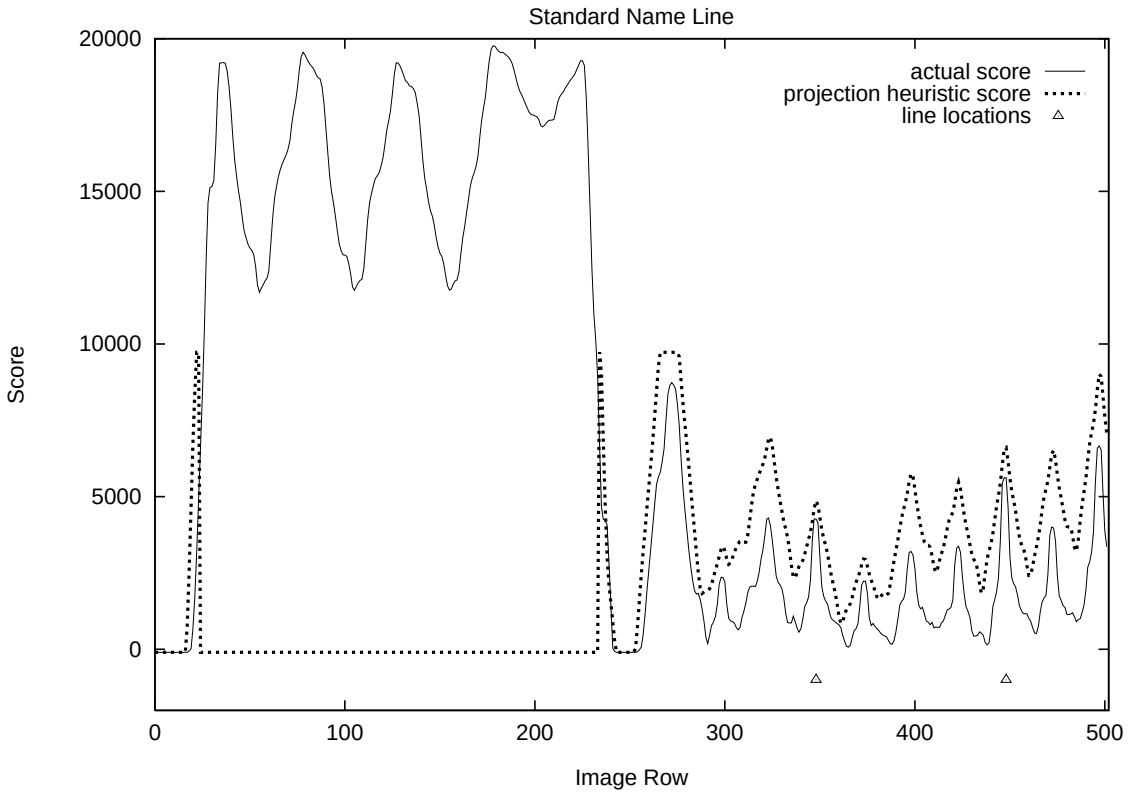


Fig. 6. Weighted projection heuristic $\mathcal{H}_{wp}$ (dash line) and actual score $\mathcal{F}$ (solid line) for image of Fig. 3.

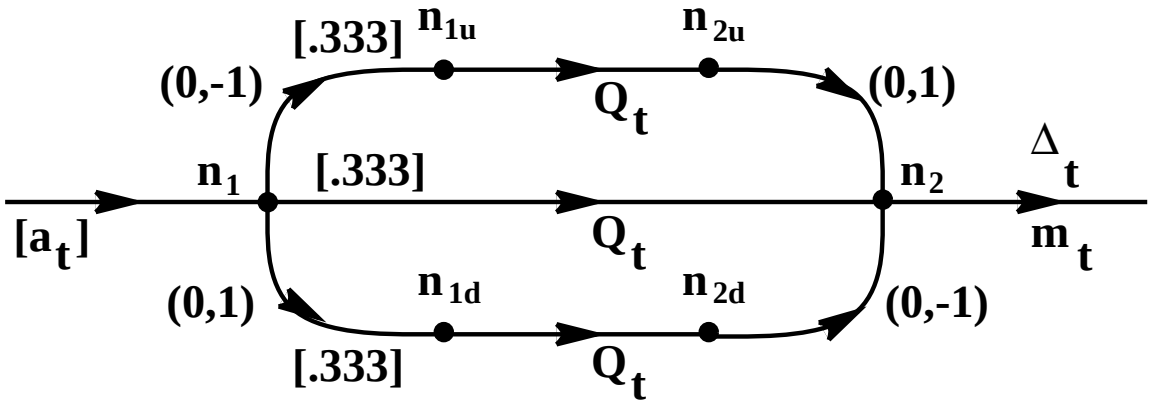


Fig. 7. Source modification to accommodate character baseline jitter. This network replaces each transition of the original model that has a non-null template.

η	Iterations	Rows decoded	CPU Time mins.	CPU Time speedup	Message % correct	Path admiss.
.99	25	8889	12.3	2.46	99.4	.998
.98	23	8119	11.3	2.67	99.4	.992
.95	21	6799	9.39	3.22	99.4	.988
.90	19	5787	8.30	3.64	98.9	.966
.80	17	4429	6.74	4.48	96.8	.898

TABLE II

EFFECT OF BOUNDING FACTOR η ON PERFORMANCE OF ICP IN YELLOW PAGE COLUMN DECODING.