Reading Chess

HENRY S. BAIRD, MEMBER, IEEE, AND KEN THOMPSON

***Abstract*—In an application of semantic analysis to images of extended passages of text, several volumes of a chess encyclopedia have been read with high accuracy. Although carefully proofread, the books were poorly printed and posed a severe challenge to conventional page layout analysis and character-recognition methods. An experimental page reader system carried out strictly top-down layout analysis for identification of columns, lines, words, and characters. This proceeded rapidly and reliably thanks to a recently-developed skew-estimation technique. Resegmentation of broken, touching, and dirty characters was handled in an efficient and integrated manner by a heuristic search operating on isolated words. By analyzing the syntax of game descriptions and applying the rules of chess, the error rate was reduced by a factor of 30 from what was achievable through shape analysis alone. Of the games with no typographical errors, 98% have been assigned a legal interpretation, for an effective success rate of 99.995% on approximately one million characters (2850 games, 945 pages). We discuss several computer vision systems-integration issues suggested by this experience.**

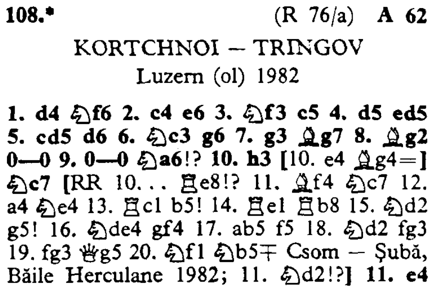
***Index Terms*—Character recognition, chess, document image analysis, layout analysis, semantics.**

I. Introduction

We present an engineering case study of a complete computer vision system exhibiting unusually high competency in a complex application, in part through the use of computable semantics. The system is a page reader and the application is the extraction of chess games from the *Chess Informant* [5], a series of volumes describing games of theoretical interest.

The books are poorly printed, and pose a severe challenge to conventional layout analysis and character recognition methods. Novel features of our experimental page reader include a strictly top-down layout analysis approach that works rapidly and reliably, largely owing to a recently-developed technique for estimating the alignment angle of blocks of text. Preliminary classification is performed before attempting to locate baselines or partition lines into words. Broken, touching, and dirty characters are handled in an integrated manner by a shape-guided heuristic operating on isolated words.

The games are extracted using a combination of layout and syntactic rules. Each game is segmented into moves (50 on average), and commentary is recognized and stripped off. Each move consists of a move number and two ply (half-moves), and each ply is described in three



108.* (R 76/a) **A 62**

KORTCHNOI – TRINGOV

Luzern (ol) 1982

1. d4 ♘f6 2. c4 e6 3. ♘f3 c5 4. d5 ed5 5. cd5 d6 6. ♘c3 g6 7. g3 ♗g7 8. ♗g2 0—0 9. 0—0 ♘a6!? 10. h3 [10. e4 ♗g4=] **♘c7** [RR 10... ♖e8!? 11. ♗f4 ♘c7 12. a4 ♘e4 13. ♖c1 b5! 14. ♖e1 ♖b8 15. ♘d2 g5! 16. ♘de4 gf4 17. ab5 f5 18. ♘d2 fg3 19. fg3 ♕g5 20. ♘f1 ♘b5∓ Csom – Şubă, Băile Herculane 1982; 11. ♘d2!?] **11. e4**

Fig. 1. A sample of text from the *Chess Informant* (shown 1.55 × actual size). This is a game's header and opening moves, including two parenthetical comments in move number 10.

characters on average (see Fig. 1). By applying knowledge of the rules of chess (the "semantics"), each move is checked for legality directly in the context built up by prior moves and indirectly through the consistency of later moves. If the interpretations of moves suggested by shape analysis are inadequate, alternatives are generated, again by invoking the rules of chess. Even though this semantic analysis is fully-backtracking, intolerable runtimes are prevented by the high quality of the shape recognition overall and a roughly uniform distribution of serious mistakes.

Of the first 142 games, two were flawed by typographical errors (editorial or typesetting mistakes), and the semantic analysis failed, for assorted reasons, on three more. The resulting *98% of games correct* implies that 99.99% of the moves, and 99.995% of the characters, were interpreted correctly. Without semantic analysis, only 40% of the games would have had a legal interpretation, even though the character recognition rate due to shape analysis alone is reasonably high (99.5%).

Prior approaches to the detection and correction of errors in text images have operated on *isolated words* from natural languages, typically by means of dictionary lookups (e.g., [2] [11] [13]) and character *n*-gram frequency analysis (e.g., [14] [16] [7]). The lack of effective and efficient algorithms for the analysis of natural language has long frustrated the natural desire to extend this to sentences. The comparative tractability of the semantics of chess offered an opportunity to experiment on *extended sequences of text*, often consisting of hundreds of characters, that make up complete games.

The exercise had two cooperating purposes: the first author's principal motive was to experiment with contextual reasoning in image analysis and the second author wished

Manuscript received May 23, 1988; revised October 12, 1989. Recommended for acceptance by C. Y. Suen.

The authors are with AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974.

IEEE Log Number 9034827.

to bring a large fraction of the *Informant* on line to support a variety of studies in computer chess, among them the enrichment of the opening book of Belle, a chess-playing machine [4].

Section II describes the books in the *Chess Informant* series. Page layout analysis is discussed in Section III. Classification, word segmentation, and baseline-finding are discussed together in Section IV. A method to cope with broken, touching, and dirty shapes is discussed in Section V. Extracting games and moves using layout geometry and syntax is described in Section VI. Use of the rules of chess to detect and correct classification errors is described in Section VII. Performance statistics are summarized in Section VIII. Systems engineering aspects are discussed, finally, in Section IX.

II. The Chess Informant

The *Chess Informant* [5] is a series of volumes describing chess games selected by an international editorial board for their theoretical interest. Over 40 volumes have appeared, and continue to appear at the rate of about two a year. Each volume contains a description of about 800 games, with detailed commentary (along with indexes, rankings lists, etc). The game descriptions are not available in computer-legible form.

An excerpt from a typical game description is shown in Fig. 1. Each game is introduced by a header giving the game number, two codes classifying the opening play, the names of the players, the location of the tournament, and the year.

A game description consists of a list of numbered moves with commentary. Some comments are brief remarks using special symbols, such as !?, meaning "a move deserving attention," or $\pm$, meaning "white has the upper hand." Other comments take the form of lists of alternative moves, enclosed in square brackets [· · ·].

The series is published in Beograd, Yugoslavia. The text was set using letterpress techniques (handset lead type) using over 200 distinct letters (see Fig. 2), of 10 point type size.

These include complete alphanumeric fonts in both roman and boldface, chesspiece characters, letters modified with diacritical marks, and a set of special symbols used for commentary. Lines of text are set solid: that is, spaced vertically as closely as possible for the point size. The column-break is narrow, and lines are not horizontally aligned across it.

The quality of the paper and ink impression is variable from volume to volume and from page to page. Uneven inking has produced a wide range of character densities and slipping type has produced gross shape distortions and proximity effects (Fig. 3).

Other defects, including surface dirt, ink smearing, and nonrectilinear layout of columns and lines, are discussed later. Although the *Informant*'s print quality is often worse than that seen in most modern books and journals, it differs from them more in degree than in kind of defects.

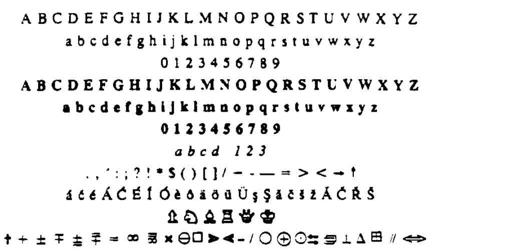

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9
a b c d 1 2 3

Fig. 2. Most of the distinct letter shapes occurring in the game descriptions.

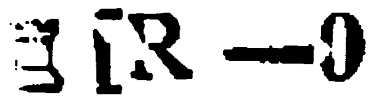


Fig. 3. Distortions due to uneven inking, poor impressions, and broken or tilted type.

III. Layout Analysis

Each page was scanned on a Ricoh IS-30 document scanner at a resolution of 300 lines per inch (12 lines/mm), producing a binary image of about 8.5 Mbits, of which typically 800 000 are black. Maximal subsets of 8-connected black pixels ("blobs") are found, typically 3000 per page. Blobs that are manifestly too large to be a 10-point character (or a string of connected characters) are ignored.

We have found it is not necessary to examine blobs in detail in order to analyze page layout into columns and lines (for other approaches, see [10], [9]). We simply view each blob as a bounding rectangle (Fig. 4). Throughout this analysis, we proceed strictly top-down, from coarser partitions of the page area to finer ones (for a survey and analysis of other methods, see [15]). The motive for this is that at each step we can make decisions based on a maximum of statistical support and with a minimum of *a priori* assumptions.

The first step is to determine the skew angle of the page as a whole. This is done by a recently-developed technique [1] that is fast and accurate to a resolution of 2 minutes of arc. The page is corrected for skew by "pseudo-rotating" the boxes; each is translated so that the set of their centers rotate, but the bitmap within each box is not rotated. This works well in practice since with ordinary care a person can place a document on a scanner within 3° of vertical, and such a small rotation has little effect on character bitmaps; however, it has large effects on page layout analysis, as we will see.

In order to find the location of the two columns, each box is widened slightly by a fixed multiple to encourage overlapping of adjacent boxes, then projected vertically to give a one-dimensional distribution (Fig. 5); the widening is governed by the assumption that a column-break is at least as wide as three em-spaces. This distribution exhibits a small number of prominent plateaus which are easily located by thresholding. The threshold is chosen to

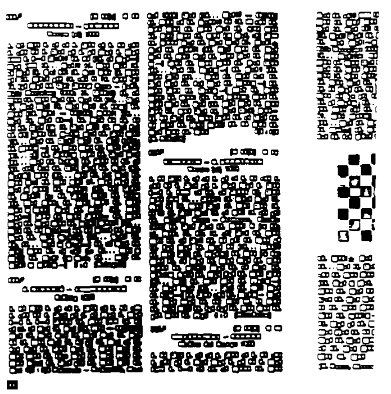


Fig. 4. An image of a page, before skew correction. Two columns of a full page are on the left, and part of the next page appears on the right. Connected components are represented by their bounding rectangles.

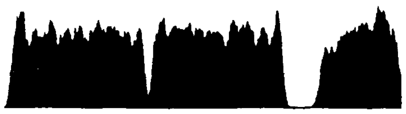


Fig. 5. The vertical projection profile of the bounding rectangles shown in Fig. 4, after skew correction.

be the 8th percentile value of the histogram of values in the distribution. Due to the narrowness of the column-break, this technique would fail if the skew estimation were in error by as little as 1.0° of arc. This problem is compounded by a printing defect: it is possible for the columns to be distorted by an affine shear of as much as 0.5° (illustrated schematically in Fig. 6). Shear correction using the best alignment angle discovered among roughly-vertical angles effectively solves this problem.

Before proceeding to find lines in each column, it is necessary to recompute the skew angle for each, due to a second unexpected printing defect: even when the columns show no affine shear, it is possible for each to sit at a different angle, as much as 0.6° apart (illustrated schematically in Fig. 7). After recorrecting each column for skew, it is possible to locate lines of text by analyzing the horizontal projection (Fig. 8). Here a fixed threshold is used, on the assumption that a line of text consists of at least two characters. Break decisions are less critical than for columns, and this technique has worked well on a wide variety of text. However, where line-spacing is tight, it could fail if the estimate of skew angle were off by as little as 0.3° of arc.

The contents of each line is then sorted left to right, and, still manipulating only the boxes of blobs, characters are tentatively identified, as collections of one or more blobs. The rule used is to combine any pair of blobs (or, recursively, characters) that overlap vertically by more than 70% of the width of either. This test is performed also at a 12° slant, the typical italic inclination angle. This works for all common fragmented characters and diacritical marks in the text, but need not be perfect, since it can be overruled later. In fact it succeeds on all but one of the 172 special characters in the Latin alphabet enumerated in the *Chicago Manual of Style* [6].

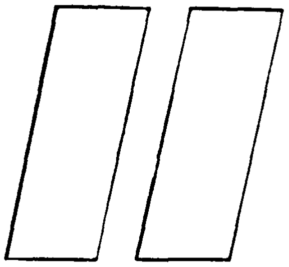


Fig. 6. Schematic, exaggerated illustration of affine shear distortion of column layout.

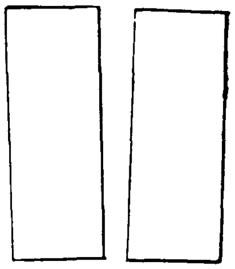


Fig. 7. Schematic, exaggerated illustrations of columns laid out at different skew angles.

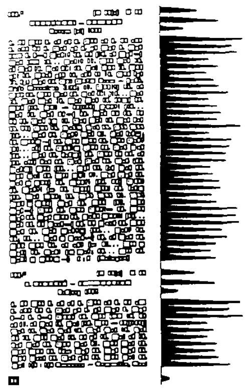


Fig. 8. The horizontal projection profile of a column of text, after skew correction.

IV. Classification, Baselines, and Words

Before attempting to locate baselines or words, we perform a preliminary classification of characters. The method used is template matching. To match a pair of characters, the centers of their bounding boxes are aligned, and a bitwise exclusive-or computed. The number of nonzero bits in the result is scaled by the total area of both patterns and subtracted from 1.0, giving a match score in the range [0,1], with 1.0 indicating a perfect match. This is a simple implementation of a long-established technique [12].

Template-matching was selected since, having fewer than 250 distinct lettershapes, there was no requirement for a multiple-font classifier (but, see [8]). Also, many characters suffered from fragmentation, to which template matching is less vulnerable than topology-based feature-analysis methods.

We manually collected over 1300 character samples, about 7 per class on average, each labeled with a correct baseline location. Classification of an unknown shape occurred by essentially exhaustive matching in this database, yielding a list of classes in decreasing order of match score, which was truncated when scores fell below 0.9 of the best. A manually-selected threshold (0.75) was used to distinguish good from bad matches.

Each match determines a baseline location, and the dominant baseline-height of each text line was chosen as the median of the set of baseline-heights for each character's first good match. This technique is fast and has proven to be accurate and insensitive to commonly-occurring problems including poor character segmentation and dirt. If skew-correction has been less accurate, a more complex two-dimensional fitting procedure would have been required.

Each character's match score is then modified by reducing it to the extent that the height-above-baseline differs from the mean for that class, assuming a Gaussian distribution with the standard error of the training samples. This final match score thus depends primarily upon similarity of shape, and secondarily on vertical placement in the line. In this application, baseline displacements as small as $1/5$ of the x-height of the font can be crucial, since this distinguishes between the square brackets '[' and ']' and similar shapes such as 'I' and 'J'. The brackets are important lexical break characters used to extract games and ignore commentary.

Text lines are partitioned into words by examining the statistics of intercharacter spacing within each column. A histogram of intercharacter distances usually possesses a bimodal distribution whose minimum makes a good threshold for distinguishing word- from character-spaces. In practice, the histograms tend to be sparse and badly-behaved. By quantizing coarsely (by $1/6$ of an em-space), and requiring the threshold to lie within a narrow range ([0.2, 0.5] em), the results are usually good.

Most residual word-segmentation problems are caused by characters with extreme kerning properties, such as periods and wide chess-piece characters. No attempt was made to exploit special knowledge of kerning. Results were sufficiently good to serve an important function during the "clean-up" processing described next. The final syntactic and semantic analysis does not depend on word-breaks at all.

Graphics such as board diagrams were either discarded earlier as too large or appear now as a line dominated by very bad matches; these lines are detected and ignored.

V. Fragments, Smearing, and Dirt

There are three ways in which the character-finding heuristic described above can fail: 1) a character can be fragmented; 2) two or more characters can touch; and 3) stray ink marks, paper defects, or dirt can be mistaken for a character. Of course, any combination of these can occur together. Examples of each are shown in Fig. 9. In virtually every instance, these problems cause a bad match.

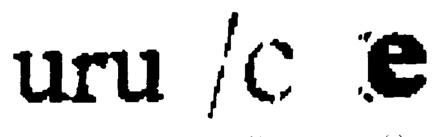


Fig. 9. Defects causing problems that are resolved after preliminary classification: (a) touching characters; (b) broken characters; and (c) dirt fragments.

We observed that it was never necessary to reach across a word-break to fix such a problem. The average length of a word is three characters, and only 7.4% of words on average have at least one character with a bad match. This suggested a strategy for correcting these defects in a unified way by a nearly-exhaustive examination of alternatives generated within the bounds of individual words. An outline of the heuristic is as follows:

```
for each bad word (7.4%) {
  repeat {
    try merging sets of adjacent characters (0.3%)
    for each bad character {
      try splitting into two (or more) (0.9%)
      if character is dirt, then delete it (0.6%)
    }
    if word is isolated dirt, then delete it (0.2%)
  }
  until word is deleted or not improved
}
```

Improvement in a word is defined as an increase in a composite word-match score, computed as the average of the character-match scores, weighted by area. About half of the bad words are improved by the heuristic. The percentages shown for each statement are for tries that succeed in improving the character or word, expressed as fractions of the total number, good *and* bad.

A character is considered to be *dirt* if it has a low match score (<0.30) and a small area (smaller than about two periods). A word is *isolated dirt* if it consists of one dirt character. These rules catch almost all cases, failing when the dirt is attached to a character (Fig. 10) or lies so close to the baseline that it resembles a period. Deleting dirt is delayed until everything else has been tried, since it cannot be undone.

In an attempt to recombine fragmented characters, an exhaustive list of strings of adjacent characters is generated. Each list is anchored at a particular bad character, and all strings are generated that reach no further than 0.5 em from the anchor, and are no more than 1.1 em wide.

When splitting characters, a modification of Kahan's method [8] is used to purpose a small number of candidate splitting points.

Although runtime of the heuristic is potentially expo-

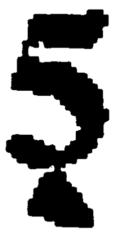


Fig. 10. Dirt fragment touching a character.

```
COLUMN 1
1.1  10p   108.* (R 76/a) A 62
1.2  10p        KORTCIINOI — TRINGOV
1.3  10p            Luzern (ol) 1982
1.4  10p   1. d4 ♘f6 2. c4 e6 3. ♘f3 c5 4. d5 ed5
1.5  10p   5. cd5 d6 6. ♘c3 g6 7. g3 ♗g7 8. ♗g2
1.6  10p   0—0 9. 0—0 ♘a6!? 10. h3 [10. e4 ♗g4=]
1.7  10p   ♘c7 [RR 10. . . ♖e8!? 11. ♗f4 ♘c7 12.
1.8  10p   a4 ♘e4 13. ♖c1 b5! 14. ♖e1 ♖b8 15. ♘d2
1.9  10p   g5! 16. ♘de4 gf4 17. ab5 f5 18. ♘d2 fg3
1.10 10p   19. fg3 ♕g5 20. ♘f1 ♘b5∓ Csom – Şubă,
1.11 10p   Băile Herculane 1982; 11. ♘d2!?] 11. e4
```

Fig. 11. Galley proof format, used for manual proofreading, for the game fragment shown in Fig. 1. Each line is labeled with the point size.

nential in the length of the word, the low frequency of bad words and the short average length of words holds it to about 1/7 of the cost of performing the preliminary classification, on average (with a large variance).

VI. Extracting Games and Moves

The result of the preceding stages of computation is a hierarchical analysis of the page into columns, lines, words, and characters, with each character owning a list of interpretations. A sample of a galley proof format of this data structure is shown in Fig. 11, showing only the top choice interpretation of each character. A more detailed format, including each character's bounding box and interpretation list, is shown in Fig. 12. The results on a sequence of pages (in detail) are concatenated and passed to the stages that apply chess knowledge.

The text from the character recognizer is examined for chess semantics with the goal of decoding the game played along with players, event name and who won. The analysis is performed in sequential phases as described below. There is no feedback between phases.

Isolating the game. Each line is examined and classified as potentially being one of the three title lines of a game. The first line (index) is left and right justified with a large space in the middle. The left part has numbers while the right part contains parentheses. The second line (players) is centered with a hyphen or dash. The third line (event) is centered. Whenever two or three consecutive lines are classified into potential title lines, the previously collected lines are analyzed as a complete game and subsequent lines are collected for the next game.

Stripping Comments. Comments are delimited by [· · ·] brackets. For this analysis brackets are matched in pairs. When a pair of brackets are found the following possibilities are recognized.

[· · ·]	The [is accepted as real.
[· · · [	If the character match scores of both brackets are above a "good" threshold, the intervening text is examined by *reverse typesetting* (described below) for the best potential]. The [and the invented] are accepted as real. Otherwise, the better [is accepted and the other is rejected as dirt.
] · · ·]	By analogy with [· · · [.

The beginning of the game is treated as if it contains a perfect], while the end contains a perfect [. The above algorithm accepts, rejects, and invents brackets so that they are balanced. Accepted brackets and their enclosed text are discarded as commentary. This eliminates about 50% of the characters from further analysis. It is good at locating brackets that were typeset but are unrecognizable due to printing defects, but is not at all good at supplying brackets that were omitted by the printer. The text contains lots of clues for real missing brackets, such as repeated move numbers, change in type face, commentary, etc; none of this is used.

```
1.060i 1.553i 1.123i 1.653i 1    9 0.90
1.137i 1.617i 1.163i 1.647i 1  per 0.95
1.220i 1.540i 1.500i 1.677i 0   \0 1.545n
1.220i 1.550i 1.287i 1.643i 1    0 0.93
1.297i 1.600i 1.427i 1.620i 1  dsh 0.94
1.433i 1.550i 1.500i 1.643i 1    0 0.93
1.560i 1.540i 1.963i 1.677i 0   \0 1.636n
1.560i 1.540i 1.683i 1.667i 1  chN 0.90
1.703i 1.577i 1.770i 1.643i 4    a 0.90    n 0.85    u 0.84    s 0.83
1.780i 1.547i 1.843i 1.647i 1    6 0.91
1.870i 1.547i 1.887i 1.643i 2   ex 0.90    1 0.81
1.917i 1.547i 1.963i 1.647i 1   qm 0.92
2.033i 1.540i 2.203i 1.677i 0   \0 1.909n
2.033i 1.543i 2.087i 1.643i 4    1 0.91    I 0.83 l-sl 0.83    i 0.82
2.100i 1.547i 2.163i 1.643i 1    0 0.95
2.173i 1.617i 2.203i 1.643i 1  per 0.92
2.260i 1.540i 2.413i 1.677i 0   \0 1.545n
2.260i 1.543i 2.343i 1.647i 2    h 0.94    b 0.89
2.347i 1.547i 2.413i 1.647i 1    3 0.89
```

Fig. 12. Detailed layout description of moves 9 and 10 of the game shown in Figs. 1 and 11. The four coordinates on the left describe bounding rectangles. The next number is a code n: if $n = 0$, then this is a layout feature, such as words (labeled "\0") or text lines (not shown); if $n > 0$, then it is a character possessing n interpretations. Each interpretation consists of a letter-name and a match merit (e.g., "chN 0.90" is a chess knight with merit 0.9).

Sketching the move sequence. The text is examined for move numbers in sequence. The test is rigorous enough so that any numbers less than almost perfect are deferred until a later phase.

Interpreting the game result. The next after the last move number is examined by reverse typesetting for one of: 1-0 (white wins), 0-1 (black wins), or 1/2-1/2 (draw).

Completing the move sequence. Missing numbers in the move sequence are filled in by reverse typesetting. Possibly missing numbers between the last backbone number and the result are similarly filled in.

Preparsing the moves. The text between move numbers is examined for two syntactically-correct chess ply, one for White and one for Black. Candidate ply are assembled from the alternatives offered by the classifier. Correct forms of a ply include Piece-Letter-Digit (PLD), LLD, LD, PLLD, PDLD, or castle. A piece is one of ♖♘♗♕♔, a letter **abcdefgh**, a digit **12345678**, and castle either **O—O** or **O-O-O**. Commentary characters may occur after each ply, outside of brackets; these are recognized and discarded. If one or more correct candidates are found for *both* ply, then they are recorded and the text is never interpreted again. When there is more than one candidate, they are sorted decreasing on the product of the match scores (i.e., classifier's top choices first). These *preparsed* moves comprise about 98% of the total moves.

Reverse typesetting is the use of the bounding rectangle of characters to prune lists of matches. Both height and width of boxes are compared, but not height above baseline. It is never required when pruning lists from the classifier, but it is helpful when scanning for missed brackets or eliminating wild choices made by the move generator.

VII. Correcting Moves Using Semantics

Preparsed moves are interpreted in the current chess context and, if legal, the move is made and the match continues. If the preparsed move is illegal the match is terminated. If the move is not preparsed, then we have reached an *impasse*, and all legal next moves from the current board position are generated and pruned using reverse typesetting. The potentially exponential runtime of this exhaustive step is managed by taking the product of the scores of the matched moves and pruning below a threshold. All matches above threshold are made and matching continues in a depth-first, best-match-first manner. The first longest match is taken as the game score.

It is conceivable that some game interpretation could be legal and still differ from the clear intention of the editor—but such forced interpretations have not been observed. An example of the results of semantic analysis are shown in Fig. 13.

The text that represents the players and event are matched against (separate) dictionaries of names. A largest substring algorithm is used to find the closest name in the dictionary. The scores for the matches are recorded on an error file and examined by hand for additions to the dictionaries. The dictionaries are initially created from the index in the *Informant*. This is highly redundant between issues with only about 5% new entries per volume after a two volume startup.

VIII. Results

Among the first 142 games, two were rejected by the system for typographical errors. Three others could not be corrected by the semantic analysis. The rest, 98% of the typo-free (correctly-printed) games, were accepted by the semantic analysis, and have been proofread by hand to confirm the interpretation. No forced interpretations (substitution errors) were found. If game interpretations were selected from those that could be assembled from the top three classifier choices (with syntactic but no semantic analysis) this fraction falls to 76%. Interpretation by shape alone (using only the classifier's top choices) would have yielded 40% of the games completely correct, for an error rate a factor of 30 worse than that achieved by semantic analysis.

One of the games that could not be fully corrected had

```
/ person: TRINGOV (Tringov =0)
/ person: KORTCI1NOI (Kortchnoi =2)
white: Kortchnoi
black: Tringov
/ event: Luzern(o1)1982 (Luzern (ol) 1982 =0)
event: Luzern (ol) 1982
result: 1-0
d4     Nf6    c4     e6     Nf3    c5     d5     e:d5   c:d5   d6
Nc3    g6     g3     Bg7    Bg2    O-O    O-O    Na6    h3     Nc7
e4     Nd7    Bf4    Qe7    Re1    f6     Nh2    Rb8    Be3    b5
f4     b4     Na4    Nb5    Rc1    Re8    Nf3    Qf8    Bf2    Bb7
Bf1    Nc7    Nd4    Kh8    Nc6    Ra8    N:b4   f5     e5     d:e5
N:c5   N:c5   B:c5   Qf7    Bd6    N:d5   Bc4    Qf6    f:e5   Qg5
Qf3    N:b4   Q:b7   Nd3    B:d3   Q:g3+  Qg2    Q:d3   Rcd1   Qb5
b4     Rac8   Qd5    Qa6    Bc5    Qa4    e6     Qa3    Rd3    Qb2
e7     a6     Qe6
class/ 004 108.*(R76/a)A62
```

Fig. 13. Results of the syntactic and semantic analysis for the game opening shown in Figs. 1, 11, and 12. Note that move numbers and commentary have been stripped off and a player's name has been corrected. The opening classification "(R76/a)A62" has been recomputed.

garbled characters at the very end, where there was no later context from which to backtrack. Another had too many consecutive impasses, at the outset of a game, leading to an unmanageably large search space. The line of text involved is illustrated in Fig. 14. The top line is the bitmap (after cleaning up dirt), and the bottom line the top-choice interpretations by the classifier. Bad matches are marked with surrounding boxes.

For 99.4% of the ply, at least one candidate suggested by the classifier was syntactically correct. Of these, the top choice was semantically legal 98.7% of the time. About 0.5% of the ply were associated with impasses, and the final correct rate among ply after semantic analysis was 99.99%. Since ply are made of about three characters on average, the effective per-character success rate is probably better than 99.995%. We have no direct measurement of the uncorrected character recognition rate, and it is difficult to infer it in an unbiased way since, among other problems, about half of the characters are ignored as commentary.

We have continued reading—without manually proofreading every game—and after four volumes (945 pages, 2850 games, 2 176 865 characters) performance is holding steady, with legal interpretations found for over 97% of the games free from typographical errors. Since about a million of these characters occur in commentary, the semantic analysis was applied to about a million characters for an effective rate of successful interpretation of 99.995%.

The system was written entirely in the C programming language and ran under various research editions of the UNIX® operating system. Runtime for the image analysis phases averaged 10 CPU minutes per page on a DEC VAX® 11/8550, of which 87% was consumed in exclusive-or operations during template matching. Connected components analysis and layout analysis required about 10 CPU seconds apiece. Resegmentation of dirty, etc. shapes required an average of one CPU minute, with a moderately large variance due to variations in image quality from page to page. The syntactic and semantic analysis was performed on a Sequent; CPU time averaged 2

®UNIX is a registered trademark of AT&T.
®VAX is a registered trademark of Digital Equipment Corporation.

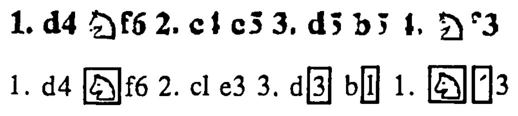


Fig. 14. A line of text containing enough impasses to induce failure of the semantic analysis. The top line is the original bitmap, after dirt fragments have been deleted, sometimes erroneously (as in "f"). The bottom line shows the classifier's top choices, with bad matches enclosed in boxes.

seconds per game, but of course with a large variance since several games required up to one CPU minute and a few (not averaged in) never terminated normally.

IX. Discussion

The *Chess Informant* is an unusual object of machine vision in that its content has effectively- and efficiently-computable semantics. The remarkably high competency achieved by exploiting this fact should encourage further attacks on images with similarly conventional, unambiguous consistency constraints.

In other respects, however, the *Informant* is representative of a broad class of printed documents. Nonrectilinear layout, tight column- and line-spacing, and broken, touching, or dirty character-images all occur in other documents. For this reason, we believe that several of the methods first applied here will be useful in achieving high performance in the general case.

The layout analysis approach is unusual in following a strictly top-down policy. Relying on a minimum of *a priori* assumptions, it is characterized by a sequence of refinement steps ordered so that the broadest statistical support may be applied to the inference of layout parameters. So, for example, text line orientation is estimated from evidence distributed over the whole page, word-break spacing over each column, and baseline is determined by consensus within entire text lines. The success of this strategy, without recourse to backtracking, in the face of manifold layout distortions is largely due to the accuracy of the skew-finding method.

We attempted to base as much of the error management as possible on a single robust measure of merit. The classifier's template match score, occasionally modified (by, e.g., height above baseline) or combined (to score words and ply), was used to control virtually all later analysis.

The attempt to apply semantic interpretation to long sequences of text ran a substantial risk of intolerable runtimes due to combinatorial explosion. Without a basis of consistently good results from the early states of layout analysis and character-shape recognition, the effort would have collapsed.

A striking feature of the engineering design is a progression from fast heuristics in the early stages to slower and more exhaustive algorithms at the end. For example, it is not until text has been segmented into words that a nearly-exhaustive search is unleashed to fix broken, touching, and dirty characters. Full backtracking is reserved for the last stage of semantic processing after several kinds of global and local context has been applied to prune interpretations to a manageable number.

Acknowledgment

An earlier version of the paper appeared as [3]. The character-splitting heuristic was a variant of one developed by S. Kahan. The UNIBUS interface for the Ricoh scanner was designed by J. Condon and built by W. Moy. Its UNIX driver was written by L. Cherry.

References

[1] H. S. Baird, "The skew angle of printed documents," in *Proc. SPSE 40th Conf. Symp. Hybrid Imaging Systems*, Rochester, NY, May 1987, pp. 21-24.

[2] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," in *1959 Proc. Eastern Joint Computer Conf.*, 1959.

[3] H. S. Baird and K. Thompson, "Reading chess," in *Proc. IEEE Comput. Soc. Workshop Computer Vision*, Miami Beach, FL, Nov. 30-Dec. 2, 1987.

[4] J. H. Condon and K. Thompson, "Belle," in *Chess Skill in Man and Machine*, P. Frey, Ed. New York: Springer-Verlag, 1982.

[5] *Chess Informant*, vols. 29, 30, 33. Beograd, Yugoslavia: Šahovski Informator, 1980-1982.

[6] *The Chicago Manual of Style*, 13th ed. Chicago, IL: University of Chicago Press, 1982, p. 254.

[7] J. J. Hull, and S. N. Srihari, "Experiments in text recognition with binary *n*-gram and Viterbi analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 520-530, Sept. 1982.

[8] S. Kahan, T. Pavlidis, and H. S. Baird, "On the recognition of printed characters of any font and size," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 274-288, Mar. 1987.

[9] H. O. Kida, O. Iwaki, and K. Kawada, "Document recognition system for office automation," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, Oct. 1986, pp. 446-448.

[10] E. Meynieux, S. Seisen, and K. Tombre, "Bilevel information coding in office paper documents," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, Oct. 1986, pp. 442-445.

[11] W. S. Rosenbaum and J. J. Hilliard, "Multifont OCR postprocessing system," *IBM J. Res. Develop.*, vol. 19, pp. 398-421, July 1975.

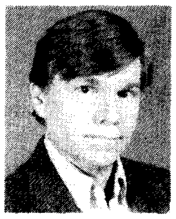
[12] H. F. Schantz, *The History of OCR*, Recognition Technologies Users Assoc., 1982.

[13] S. N. Srihari, J. J. Hull, and R. Choudhari, "Integrating diverse knowledge sources in text recognition," *ACM Trans. Office Inform. Syst.*, vol. 1, pp. 68-87, Jan. 1983.

[14] R. Shinghal, and G. T. Toussaint, "Experiments in text recognition with the modified Viterbi algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 184-193, Apr. 1979.

[15] S. N. Srihari and G. W. Zack, "Document image analysis," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, France, Oct. 1986, pp. 434-436.

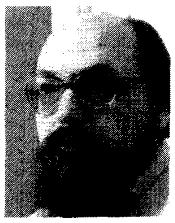
[16] J. R. Ullman, "A binary *n*-gram technique for automatic correction of substitution, deletion, insertion, and reversal errors in words," *Comput. J.*, vol. 20, pp. 141-147, 1977.



Henry S. Baird (M'80) received the Ph.D. degree from Princeton University, Princeton, NJ.

He is a Member of the Technical Staff at the Computing Science Research Center, AT&T Bell Laboratories, Murray Hill, NJ. His research focuses on algorithms for machine vision, with emphasis on the interpretation of images of printed documents. In 1976 he gave the first complete description of the sweep-line algorithm, a fundamental technique in computational geometry and an essential engineering component in the analysis of VLSI mask artwork.

Dr. Baird is an Associate Editor of IEEE T-PAMI, and a member of the Editorial Board of *Pattern Recognition* journal. He will serve as Program Chair of the 1990 IAPR Workshop on Syntactic and Structural Pattern Recognition. His Ph.D. thesis on algorithms for image matching won a 1984 ACM Distinguished Dissertation Award and was published by the MIT Press. He is a member of the Association for Computing Machinery and is active in the IAPR.



Ken Thompson was born in New Orleans, LA, in 1943. He received the B.S. and M.S. degrees in electrical engineering from the University of California, Berkeley.

In 1966 he joined the Computing Science Research Center of Bell Laboratories where he has worked until the present. He was involved in Bell Laboratories' participation in the Multics project. In 1975, he returned to Berkeley to teach Computer Science for a year. He is one of the principal designers of the UNIX time-sharing system. He is also one of the principal designers of the former World Computer Chess Champion, "Belle."

Mr. Thompson is a member of the National Academy of Science and the National Academy of Engineering.