Recognition Technology Frontiers

Henry S. Baird

AT&T Bell Laboratories
600 Mountain Avenue, Room 2C-557
Murray Hill, New Jersey 07974-0636 USA

ABSTRACT

Rapid improvements in hardware and algorithms are reshaping the technological basis of postal address recognition. An increasingly important theme is automation of the engineering process itself, through trainable classifiers, realistic distortion models, and statistical contextual analysis. Relevant basic research at AT&T Bell Laboratories includes VLSI neural networks, algorithmic pattern recognition, computational linguistics, and artificial intelligence. Interdisciplinary application of these has stimulated improvements in handwritten ZIP code recognition, machine-print address recognition, and address block location.

1. Introduction

The technological basis of postal address recognition is rapidly evolving due to progress in several research disciplines, including VLSI neural networks, algorithmic pattern recognition, computational linguistics, and artificial intelligence. We believe that a unifying theme is emerging: an increasing emphasis on automation of the engineering process itself.

Many past exercises in machine vision system engineering can be characterized as follows: *build a machine to meet the given accuracy and speed goals, by any means at hand*. This strategy has encouraged *ad hoc* custom engineering practices, using methods overspecialized to the given problem, such as manually designed rule-driven systems. This often results in high initial costs and even higher incremental costs for improvements. For this reason, many of today's machine vision systems appear to be engineering dead ends. This is unfortunate, since demands for improvements in accuracy and versatility will persist as long as machine vision remains inferior to human vision — that is, for the foreseeable future.

By contrast, many new approaches to machine vision can be characterized as follows: *build a machine to meet the given accuracy and speed goals, by the most automated means possible*. This strategy encourages the use of algorithms that are trainable by example and supported by tools for automatically choosing engineering tradeoffs (among speed, accuracy, space, etc). Such a vision system should be incrementally improvable: it can benefit from experience as larger training sets are accumulated; and, as computing technology evolves, different speed/space/accuracy tradeoffs can be readily explored.

For this promise of sustained improvement to be realized, it seems to us crucial that the underlying recognition technologies rely as *little as possible* on *a priori* problem-specific assumptions. This may seem paradoxical, since most engineering experience up to this time suggests that it is only by exploiting problem-specific knowledge to the fullest that given

accuracy and speed goals can be met. Our reply to this objection is first to agree that problem-specific knowledge must be exploited, but then to stress that this should occur ideally through automatic inference from representative samples of correct system behavior. For this reason, we look to the most generally-applicable methods arising in a wide variety of disciplines.

The rest of this paper illustrates this strategy through detailed discussion of a number of basic recognition technologies that have been explored in recent years by AT&T Bell Laboratories. We also point out their actual or potential application to postal address recognition.

Each of the topics touched on here has a large and interesting literature. For reasons of space, we have omitted many references: however, the Bell Labs' papers that we do reference give careful surveys of prior art.

2. Neural Networks

Learning and generalization from training sets [16] are two attractive features of neural networks. However, the large size of highly accurate network classifiers imply computation and data-transfer rates too great for general-purpose serial computers. For this reason, in addition to basic studies of network architectures [JBG90] and back-propagation training algorithms [13], we have explored a variety of special purpose VLSI processors.

Le Cun et al [14],[15] have described a neural network for OCR that performs particularly well on noisy, handwritten characters. This is a five-layer feed-forward network specialized for image pattern recognition, featuring local convolutions and shared weights. Experiments have revealed that low-precision arithmetic (6 bits/weight) is sufficient for the 97% of the connections located in the first 4 layers: only for the 3000 connections in the last layer is greater precision required.

A network with 136,000 connections for recognition of handwritten digits has been implemented on a mixed analog/digital neural-network chip. It is implemented in a single poly, double metal 0.9 μm CMOS technology. The die measures 4.5×7 mm^2 and contains 180,000 transistors. The architecture allows at total of 4096 synapses, and up to 256 neurons, in various configurations. Each synapse performs a 6 bit (weight) times 3 bit (state) multiply-accumulate operation. Analog circuits are used internally for reduced power dissipation, high density, and higher speed, while all input and output circuits are digital for easier integration. Its parallel architecture allows up to 2000 multiplications and additions to be computed simultaneously. The chip is optimized for locally connected, weight-sharing networks and time-delay neural networks (TDNN), but is reconfigurable to a wide variety of topologies including fully connected and recurrent networks. The peak computation rate of the chip in a maximally-parallel configuration is 5 Gconnections/s, but when specialized to character recognition, its effective rate drops to 130 Mconnections/s. The chip, considered in isolation, is capable of processing 1000 character images per second with essentially the same error rate (5%) as a simulation of the network with 32-bit floating-point precision [9],[15].

When integrated in a board-level implementation with other components (including a DSP32C chip, clock generator, state and weight memory, and VME bus interface), throughput greater than 100 characters/second has been demonstrated. The 32-bit floating point digital signal processor on the board runs at 40 MHz without wait states (100 ns per instruction) and is connected to a 256 kbyte static RAM.

The network has been tested on 2000 hand-written digits provided by the U.S. Postal Service, with an on-chip error rate of about 7%. Isolated character images are normalized for scale and stored in a 20×20 pixel grey-level image.

This digit recognizer has been used in experiments on a variety of postal address recognition problems. I will focus here on a typical one: reading handwritten ZIP codes. Given an image of a ZIP code, it must be segmented into isolated characters before they can be recognized. The ability of the neural net classifier to report a confidence measure as well as a top-choice is used to control a combinatorial search among segmentations. First, the image is ‘‘deslanted’’; then, though a heuristic analysis of local connectivity, a set of potential vertical ‘‘cuts’’ is identified. Each subset of four cuts implies a segmentation of the image into five digits. These subsets are explored in a branch-and-bound fashion, with classifier confidence scores determining the sequence, so that it is not always necessary to analyze all subsets exhaustively. The ZIP code recognizer was trained on 7000 5- and 9-digit ZIP code images, and tested on another 3000 (these were provided by contractors for the U.S. Postal service). The training procedure was unusual in several aspects: the network was initialized using a recognizer trained on 10000 isolated handwritten digits; then, as correct segmentations began to emerge from the training set, they were used to enrich the training set; in addition, malformed image fragments were trained on, as members of an ‘‘unrecognizable’’ class, to improve the statistical properties of confidence scores. The results were verified against a dictionary of legal ZIPcodes. The current accuracy of this method on whole handwritten ZIP codes is 60% correct and 39% rejects, with 0.7% error.

At an even higher level of control, it is necessary in some postal applications to locate an address block in a crowded image. A second VLSI neural-net processor has been built that is particularly well-adapted to such problems. It can be used efficiently to locate all occurrences of small local patterns in large binary images.

3. Algorithmic Pattern Recognition

In another research effort, a versatile machine-print page reader has been developed, which can be adapted to new applications (e.g. languages) with a minimum of manual effort. It has been applied to multiple-typeface English text and single-typeface Swedish, Tibetan, chess notation, and mathematical equations, as well as the recognition of machine-print postal address images. An overview of the reader’s architecture appeared as [5]; recent details of its algorithmic components are published elsewhere (references are cited below).

Automatic page readers must cope with many sources of variation, including symbol sets, typefaces, sizes of text, page layouts, linguistic contexts, imaging defects, and output encodings. Our engineering strategy has been to organize the system so that each of these can be handled separately and independently, and the results combined in arbitrary ways. To achieve this, new algorithms have often been required. For example, our geometric layout analysis subsystem is virtually language-independent thanks to novel methods for skew-correction and line-finding.

Our approach to symbol (character) recognition is a hybrid of structural shape analysis and Bayesian statistical classification, and is trainable by example, usually off-line. Shape features are constructed (not merely selected) during an automatic analysis of the training set; as a result, accurate classifiers may be rapidly built for diverse symbol sets. A pseudo-random image defect generator permits the automatic construction of classifiers given as few as one prototype image per symbol. Thus the manual effort to learn new sets of symbols and typefaces is reduced to a minimum. The classifiers exhibit strong generalization and compression across typefaces, text sizes, and commonly-occurring image defects.

The algorithms for layout analysis and symbol recognition are applicable to any

writing system in which the symbols are rigid and disconnected (spaced apart from one another), most of the time. Where characters do touch, segmentation is controlled by classifier confidence scores, so that language-specific rules are needed only for cases that are ambiguous by shape. Other linguistic context, such as provided by dictionaries, punctuation rules, and other lexical constraints, is exploited in a uniform data-directed manner.

Table-driven algorithms have been used wherever possible. As a result, the programs for recognition of symbols and inference of text size and baseline possess no language-dependent special cases. Linguistic context may be specified by arbitrary word-lists and general-purpose regular-expression patterns. Final output encoding (such as ASCII) is via user-specified tables.

The sequence of computation stages is as follows:

1. *geometric layout analysis*: connected components analysis, skew- and shear-correction, segmentation into text blocks, line-finding within blocks, character-finding within lines, and determination of reading order;
2. *symbol recognition*: classification of characters by shape, inference of text size and baseline (or top-line), segmentation of lines into words by spacing, and shape-directed resegmentation of characters within words to handle touching and broken characters;
3. *linguistic contextual analysis*: segmentation of lines into words by lexical means, exploitation of dictionaries and punctuation rules, and enforcement of inter-word consistency constraints;
4. *logical layout analysis*: partitioning blocks into sections, labeling sections by function, and reassembling sections (across blocks) into reading order; and
5. *output encoding*: mapping the internal representation of the analysis into character codes (*e.g.* ASCII, JIS, or Unicode), possibly annotated by formatting directives (*e.g.* troff or SGML).

The analysis of layout geometry follows a global-to-local strategy [3], that is, greedy and guided by global evidence. A non-backtracking sequence of model-refinement steps is executed in an order constrained by dependencies among model parameters and maximizing the statistical support available at each step for inference of the parameters. Experiments suggest that this is more robust than bottom-up merging methods and more efficient than backtracking top-down methods, while requiring relatively little *a priori* information. In particular, the method requires no prior knowledge of the symbol set, and only rough estimates of the range of text sizes.

First, black 8-connected components are extracted. Using the set of approximate locations of components, the dominant skew and shear angles of the page as a whole are measured. If these angles differ from strictly horizontal or vertical by more than 0.03 degrees, they are corrected. Our skew estimation algorithm [1] is one of the fastest and most accurate reported, and works without modification on a variety of layouts, including multiple blocks, sparse tables, and mixed sizes, line-spacings, and typefaces. Its accuracy is unaffected by touching characters, as long as they are in the minority. Importantly for postal address recognition, the method is also very sensitive, and works well even on sparsely-populated text images. In recent trials, the method skew-corrected 98% of 2000 difficult† postal address block images supplied by the U.S. Postal Service.

Next, the text block is segmented into lines of text. Again, a global-to-local strategy

† These images were rejected, for a variety of reasons, by current USPS address recognition units.

is effective, even on columns in which line spacing and text size vary over a wide range. This is achieved by analysis of the horizontal projection profile using digital signal processing methods: local derivatives and autocorrelation to estimate the dominant line-spacing; smoothing to sharpen the peaks associated with text lines; non-maximum suppression to locate the peaks; and finally local-minima finding to choose horizontal cuts. Trials on pages printed in over a dozen writing systems suggest that this method succeeds on the great majority of layouts, without language-specific rules. The method successfully isolated text lines in 98% of the difficult USPS address blocks.

We have organized the recognition subsystem so that classifiers can be built for any given collection of rigid symbols under any specified image defect distribution: this normally occurs off-line during highly-automated training; the resulting classifier tables can be archived and selected at runtime.

A quantitative model of imaging defects [4] permits us to build accurate classifiers with a minimum of manual effort. The model includes parameters for size, digitizing resolution, blur, binarization threshold, pixel sensitivity variations, jitter, skew, stretching, height above baseline, and kerning. The model has been calibrated on image populations occurring in printed books and typewritten business material, and is expressed as a distribution over the model parameter space. Associated with it is a pseudo-random defect generator that reads one or more sample images of a symbol and writes an arbitrarily large number of distorted versions chosen from the distribution. This allows us to guarantee that our training sets possess at least the minimum number of training samples, uniformly distributed among symbols/fonts/sizes, required by the trainable classifier technology. In this way, we have constructed classifiers exhibiting highly uniform accuracy across more than 100 typefaces [6].

Many published decision-theoretic recognition methods require, as a manual first step, the exhaustive specification of a set of features (often real components of a fixed-length vector); later, during the training phase, the feature set may be pruned or transformed automatically for improved results. An interesting aspect of our method [2] is that the feature set need not be specified in advance: instead, only a handful of primitive shape types — edges, holes, convex and concave boundary arcs, *etc* — are provided, in the form of algorithms to extract them from boundary lists and moments of area. Any given set of extracted primitive shapes is converted into a feature vector, suitable for a statistical procedure, by means of a mapping which is itself constructed during automatic analysis of the distribution of primitives in the training set.

The first use of classifier confidence scores is the inference of text size and baseline within lines of text — the *local geometric context* of symbols required in many languages (*e.g.* to distinguish upper from lower case in the Latin alphabet). Each alternative symbol interpretation implies a text size (estimated from per-class statistics collected during training): the median of these sizes, weighted by confidence, is selected as the line's dominant text size. This size is then used to prune the interpretations. It is as though a distinct classifier, sensitive to both shape and size, had been used, but the incremental computational cost is negligible. In an analogous way, baseline is selected; this works equally well with no change for writing systems with a top-line convention, such as Tibetan. Thus, by exploiting confidence scores provided by the symbol recognition subsystem, we have avoided building a separate system of symbol-set-dependent rules (*e.g.* ''typographical rules'') merely to infer local geometric context. The baseline inference method succeeded on 95% of the difficult USPS images.

Linguistic context, such as provided by dictionaries, punctuation rules, and other lexical

constraints, is often effective in resolving residual ambiguities of shape caused by badly-designed symbol sets, overlapping typeface variations, and distortions due to imaging defects. We exploit these in a data-directed manner by filtering the lattice of word interpretations.

We have experimented principally with veto filters, which merely accept or reject a word interpretation. These include all-alphabetic or all-numeric rules (quite effective on Latin languages), punctuation prefix/suffix patterns, dictionary and word-list lookup, and regular expression patterns. Some common filters are built in, but we also permit the user to provide arbitrary programs, executed as UNIX processes and attached to the OCR program by pipes.

The contextual-analysis control algorithm works as follows: alternative word interpretations are generated (from the lattice that is output by symbol recognition), in descending order of word confidence scores derived from the symbols' scores. The list is run through each filter in turn: if a filter accepts no interpretation, the list is not modified (in case the filter is not relevant); but if any interpretation is accepted, then all rejected interpretations are pruned. The user may control at run-time how far down in word-confidence order the filters look, to trade off runtime for accuracy. When applied to address block recognition, these filters include ZIP-code recognizers as well as specialized lexicons for state, city, and street names; consistency checking among words must be performed by procedures specific to the postal application.

These contextual-analysis methods are all *data-directed*: they merely select among alternatives generated by shape recognition. In some applications, *model-directed* analysis may be required: these are able to supply missing alternatives by appeal to statistical, linguistic, or semantic models. In an experiment of this kind, on several volumes of a chess encyclopedia [7], we obtained dramatically improved results. The initial error rate of 0.5%, achieved by symbol recognition, was reduced to 0.005% by contextual analysis. This included syntactic analysis tailored to the books' chess notation, and semantic analysis based on the rules of chess. Compared to natural languages, short runs of chess moves imply relatively small sets of alternative interpretations, making possible an efficient model-driven analysis. It is interesting to note that this record-breaking accuracy was achieved without backtracking to the symbol recognition stage. It is conceivable that adaptations of such a method could be helpful in checking consistency among the parts of address blocks.

4. Computational Linguistics

Two decades of basic research in computational linguistics has resulted in a number of recognition technologies useful also in image recognition. We will focus on one of these: hidden markov models (HMM).

The recognition of extremely poorly printed text is often difficult due to the high frequency of connected characters (requiring automatic segmentation), and the degraded and distorted shape of the characters themselves. In Section 1, we have described a recognition-driven segmentation heuristic that works well on moderately degraded text, but that will often fail in these extreme cases. I say ‘‘extreme’’ only because many present recognition algorithms fail, not because such images are rare (they are in fact common, in multiple-generation copied images, or FAX images), or because they seriously impair human reading performance (in fact people can often read them with high accuracy, with a little extra attention).

Recent work at Bell Labs [8] has applied hidden markov modelling methods to this problem. The states of the HMM correspond to subdivisions of word-images (called ‘‘segments’’), that may be (and, by design, often

are) fragments of characters. This permits recognition of words in cases where the individual characters, even if they could be isolated, could not reliably be recognized.

The method is trainable, at several stages: (a) segments are identified by an automatic clustering method; (b) transition probabilities between segments are inferred (separately for each character) automatically from a labeled training set; and (c) transition probabilities between characters may be estimated from samples of input text (in easily-processed ASCII). Parts of the algorithm that are not trainable by example include the division of input word-images into segments (this is a heuristic relying on prior estimates of text size and baseline location).

During recognition, a level-building dynamic programming algorithm combines segmentation and recognition of characters in a unified computation, choosing the most probable segmentation of the word into characters. A modified Viterbi scoring algorithm is used to match the unknown connected segments against the single-character HMMs.

This experimental work has so far been applied to small-alphabet, single-font images of grossly distorted words. These trials suggest that is far superior to the recognition-driven segmentation algorithms described in the previous sections. The low digitizing resolution often used in postal address imaging equipment, combined with the highly variable quality of printing faced by these systems, suggest that methods of type may be required for significant accuracy improvements.

5. Artificial Intelligence

Basic studies in artificial intelligence have led to methods for combining multiple knowledge sources in complex decision systems. One interesting and successful application of these is the automatic correction of the output of OCR systems [11], where the OCR system is considered as a ‘‘black box.’’ Synthetic (*a priori*) and statistical (empirical) models of the recognition device and of the characters and words of the text corpus to be read are combined in a unified hierarchical Bayesian computation.

The synthetic model captures *a priori* assumptions about the problem domain such as: (a) most characters are correctly identified; (b) spaces are more reliably identified than printing characters; and (c) the presence of ascenders and descenders are likely to be preserved.

The empirical model captures string rewriting rules and character and word *n*-gram occurrence statistics; these are inferred automatically from samples. We allow *m*-to*n* string rewrites, where $0 \leq m \leq 3$ and $0 \leq n \leq 2$. Our model grants equal importance to non-alphabet characters such as digits and punctuation. Space is used to delimit words, but is also allowed to participate in corrections. Problems of left and right punctuation are taken seriously as well as adjacent punctuation symbols. We have restricted our attention to unigram and digram frequencies, since exploiting higher-order *n*-gram statistics is impeded by undertraining due to sparse data.

Training is carried out off-line using lists of input-output pairs of text strings provided by an automatic alignment algorithm that compares ground truth text with the OCR output. The model of recognition errors is constructed by a branch-and-bound search of alternative hypotheses, where each hypothesis is a set of rewrite rules, and each node of the search graph is a partial hypothesis. Character and word *n*-gram statistics are inferred straightforwardly from the ground truth data.

The on-line correction algorithm runs in three phases. In the first phase, for each word (space-delimited string), one or more candidate output strings, labeled with probabilities, are proposed, using the rewriting rules. These are verified by reference to the ‘‘dictionary’’

(words with non-zero unigram probability); words not in the dictionary may still be accepted based on character trigram occurrences. In the second phase, adjacent strings which may have resulted from word-splitting errors are proposed. In the third phase, candidates for each string are reordered based on word digram probabilities, and word digram conditional probabilities.

The system has been tested on several different OCR systems running on specialized corpora (including software testing documents), and on large general corpora (including a 37.5 million words off the 1990 AP wire). Experimental trials show improvements of approximately 70-80% under a wide range of conditions on documents with an initial word error rate of 7-16%. It seems likely that these methods will perform similarly on OCR problems arising in postal address recognition.

6. Summary

We have discussed a number of general recognition technologies, chosen from among those explored in recent years by researchers at AT&T Bell Laboratories, together with a description of their actual or potential application to postal address recognition. Essential to the success of each of these is that they are automatically trainable to a significant degree. We expect to see further interdisciplinary application of these methods to postal and telecommunications problems in the next few years.

7. Acknowledgements

Larry Jackel has led the exploration of neural nets and special hardware for character recognition, described in Section 2: his technical staff has included Bernhard Boser, Jane Bromley, John Denker, Hans-Peter Graf, Isabelle Guyon, Donnie Henderson, Wayne Hubbard, Yann LeCun, Ofer Matan, Eduard Säckinger, and Sara Solla. The page reader described in Section 3 is a lineal descendant of one [12] built by Theo Pavlidis, Simon Kahan, and the present author; later, Susan Jones, Steve Fortune, and David Ittner contributed algorithms for geometric layout analysis. The hidden Markov modeling strateg of Section 4 was explored by Chinmoy Bose and Shyh-Shiaw Kuo. The Bayesian post-processor of Section 5 was developed by Mark Jones, Guy Story, and Bruce Ballard. The applications teams that applied these ideas to postal address recognition problems were led by Charlie Stenard, Ivan Strom, and Tom Shoemaker.

8. References

[1] Baird, H. S., ‘‘The Skew Angle of Printed Documents,’’ *Proceedings, 1987 Conference of the Society of Photographic Scientists and Engineers*, Rochester, New York, May 20-21, 1987.

[2] Baird, H. S., ‘‘Feature Identification for Hybrid Structural/Statistical Pattern Classification,’’ *Computer Vision, Graphics, & Image Processing* **42**, 1988, pp. 318-333.

[3] Baird, H. S., ‘‘Global-to-Local Layout Analysis,’’ *Proceedings, IAPR Workshop on Syntactic and Structural Pattern Recognition,* Pont-à-Mousson, France, 12-14 September, 1988.

[4] Baird, H. S., ‘‘Document Image Defect Models,’’ *Proceedings, IAPR 1990 Workshop on SSPR,* Murray Hill, NJ, June 13-15, 1990.

[5] Baird, H. S., ‘‘Anatomy of a Page Reader,’’ *Proceedings, IAPR Workshop on Machine Vision Applications,* November 28-30, 1990, Tokyo, Japan.

[6] Baird, H. S., and R. Fossey, ‘‘A 100-Font Classifier,’’ *Proceedings, 1st Int’l Conf. on Document Analysis and Recognition*, St-Malo, France, 30 September - 2 October, 1991.

[7] Baird, H. S., and K. Thompson, ‘‘Reading Chess,’’ *IEEE Trans. PAMI,* Vol.

PAMI-12, No. 6, June 1990, pp. 552-559.

[8] Bose, C., and S.-S. Kuo, ‘‘Connected and Degraded Text Recognition Using a Hidden Markov Model,’’ [submitted to] *IEEE Trans. on PAMI*.

[9] Boser, B., and E. Säckinger, ‘‘An Analog Neural Network Processor with Programmable Network Topology,’’ *ISSCC Dig. Tech. Papers*, pp. 184-185, IEEE Int’l Solid-State Circuits Conf., 1991.

[10] Jackel, L. D., B. Boser, H. P. Graf, J. S. Denker, Y. Le Cun, D. Henderson, O. Matan, and R. E. Howard, ‘‘VLSI Implementations of Electronic Neural Networks: An Example in Character Recognition,’’ *Proc., IEEE Int’l Conf. on Systems, Man and Cybernetics*, 1990.

[11] Jones, M., G. Story, and B. Ballard, ‘‘Integrating Multiple Knowledge Sources in a Bayesian OCR Post-Processor,’’ *Proceedings, 1st Int’l Conf. on Document Analysis and Recognition*, St-Malo, France, 30 September - 2 October, 1991.

[12] Kahan, S., T. Pavlidis, and H. S. Baird, ‘‘On the Recognition of Printed Characters of any Font or Size,’’ *IEEE Trans. PAMI*, Vol. **PAMI-9**, No. 2, March, 1987.

[13] Le Cun, Y., et. al., ‘‘Backpropagation applied to Handwritten Zip Code Recognition,’’ *Neural Computation I*, pp. 541-551, 1989.

[14] Le Cun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, and H. S. Baird, ‘‘Constrained Neural Network for Unconstrained Handwritten Digit Recognition,’’ *Proceedings, Int’l Workshop on Frontiers in Handwriting Recognition*, Montreal, 2-3 April, 1990.

[15] Le Cun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, ‘‘Handwritten Digit Recognition with a Back-propagation Network,’’ in David S. Touretzky, editor, *Neural Information Processing Systems*, pp. 396-404, Morgan Kaufmann Publishers, San Mateo, CA 1990.

[16] Solla, S., ‘‘Supervised Learning and Generalization,’’ *Neural Networks: Biological Computers or Electronic Brains*, pp. 21-28, Springer-Verlag (Paris, 1990).