Automated Table Processing: An (Opinionated) Survey[1]

Daniel Lopresti
`dpl@research.bell-labs.com`

Bell Laboratories
Lucent Technologies, Inc.
600 Mountain Avenue
Murray Hill, NJ 07974

George Nagy
`nagy@ecse.rpi.edu`

Department of Electrical, Computer,
and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Abstract

Tables are the only acceptable means of communicating certain types of structured data. A precise definition of "tabularity" remains elusive because some bureaucratic forms, multicolumn text layouts, and schematic drawings share many characteristics of tables. There are significant differences between typeset tables, electronic files designed for display of tables, and tables in symbolic form intended for information retrieval. Although most research to date has addressed the extraction of low-level geometric information from scanned raster images of paper tables, the recent trend toward the analysis of tables in electronic form may pave the way to a higher level of table understanding.

Recent research on table composition and table analysis has improved our understanding of the distinction between the logical and physical structures of tables, and has led to improved formalisms for modeling tables. The present study indicates that progress on half-a-dozen specific research issues would open the door to using existing paper and electronic tables for database update, tabular browsing, structured information retrieval through graphical and audio interfaces, multimedia table editing, and platform-independent display.

1 Introduction

1.1 Why tables?

Tables are the prevalent means of representing and communicating structured data. They may contain words, numbers, formulas, and even graphics. Developed originally in the days of printed or handwritten documents, they have been adapted to word-processors and page composition languages, and form the underlying paradigm for spreadsheets and relational database systems.

Some common examples of data usually presented in the form of tables are calendars, rail and flight schedules, financial reports, experimental results, and grade reports. Note that not all tables can be easily interpreted using only common sense: consider, for instance, the Periodic Table of the Elements (see Figure 1). Appendix A presents a number of other examples chosen primarily to illustrate difficult cases from the standpoint of automated table understanding. Some of these are quite challenging even from a human perspective.

The only other common representation for structured data is a list. If we consider ordered lists analogous to vectors, then we can think of tables as analogous to matrices. Unlike vectors and matrices, lists and tables may contain non-numeric data items. Graphs are required for relationships more complex than can be represented by tables and are used

[1]Presented at *Third International Workshop on Graphics Recognition*, Jaipur, India, September 1999.

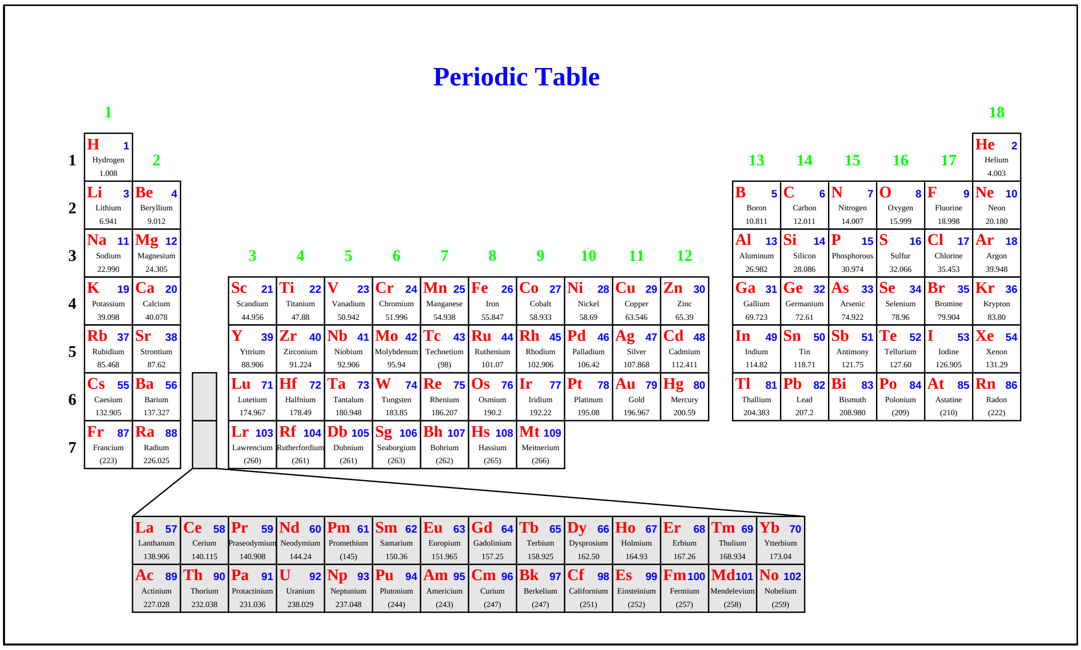


Figure 1: Periodic Table of the Elements (from http://www.trends.net/~mu/misc.html).

primarily for *inter*-document structure. Trees are often used to represent *intra*-document structure.

It is noteworthy that the need to analyze and reformat the 1890 U.S. Census forms launched the punched-card "tabulator" industry. Electronic computers were commissioned during WWII for computing ballistic tables. The major commercial applications envisioned for computers in the fifties centered on database manipulation, which remains the mainstay of business data processing.

1.2 What is a table?

Some multicolumn text configurations, like telephone directories, have tabular layouts (*e.g.*, the center region in Figure 6 in the appendix). Some engineering drawings also look like tables. Furthermore, there is no precise distinction between "table" and "form."

In common parlance we refer to two-dimensional assemblies of cells used to *present* information as *tables*. When two-dimensional assemblies of blank cells are used to *collect* information, we call them *forms*. Some forms are laid out in a regular grid like a table, others are not. Peterman *et al.* distinguish between tables and forms as follows: tables have a regular, repetitive structure along one axis, so that the data type is determined by either the horizontal or the vertical index. Forms represent only a one-to-one mapping between indices and data, in which there are no implications of regularity of data [41].

Forms processing is now a major industry. Large applications, such as medical claims processing, state income tax, insurance and retirement systems require conversion of several hundred thousand forms per day. In many such applications most forms are filled out by hand. The similarities between table and form processing are emphasized in [52] and [7]. Continuing efforts to pass processing costs down to the end users will cause many of these mass form-processing applications to be migrated to the Web. Electronic forms are based on HTML, JAVA, Active-X, or XML. JetForm, the market leader, has over five million users.

So far there is no comparable table-processing industry, but some service bureaus do offer conversion of printed tables to electronic form. Research on the conversion and interpretation of tabular material has been popular since the mid 1980's.

1.3 Metadata

Metadata is (formal) data that describes some collection of data, just as a metalanguage is a (formal) language that describes some language. Table processing is often mentioned in the context of metadata. It is, of course, only one example of metadata, and is only tenuously related to the use of the term in library science, Web searches, optical character recognition, document image analysis, or programming and scripting languages. Nevertheless, it is clear that unlike natural-language text conversion, where sequential character-by-character conversion of printed text into a simple symbolic form like ASCII may suffice, in the case of tables it is essential to extract the metadata that represents the relationships among the entities. In the sequel we shall endeavor to clarify the role of metadata in table processing.

1.4 Rationale for this study

It appears likely that the automated or semi-automated interconversion of tables from one medium to another (*e.g.*, from paper or electronic text to a spreadsheet, database, or query-answer system), or from one format to another in the same medium (*e.g.*, for different display sizes) will prove desirable in a variety of computing environments. In some applications it may be advantageous to query and reference tabular data without regard to the underlying medium or form.

The object of this study is to collect information about the composition, use, interpretation and understanding of tables that may prove useful in the development of tools for manipulating multimedia tables.

1.5 Guide to the remainder of this paper

There is no useful precedent for organizing a background study about tables such as this. Our organizing principle was simply to attempt to orthogonalize the various issues, so as to be able to make independent decisions regarding tool development.

We first consider the underlying media under the headings of paper, electronic, and symbolic form. Tables on paper must be optically scanned for any type of automated processing. Electronic tables such as those found in word-processing documents, e-mail, and the Web, already have the content of the leaf cells in symbolic form, so no OCR is necessary, but the structure is seldom available in a convenient form. Symbolic tables such as spreadsheets and databases reveal not only the content, but also the structure, in symbolic form.

Format considerations are of primary interest only in paper tables. We enumerate the graphic conventions, including layout and typography, that are used to designate the relationship between the various elements. Some of these conventions are mimicked in electronic tables: for instance, horizontal rulings may be represented by lines of underscores, asterisks, or hyphens (see Figure 6). Leaders of dots are common in both (Figure 4).

Next we consider the syntax and semantics of tables. We proceed from simple structures with no explicit headings to the most complex cases that include nesting. We discuss the various formal paradigms that have been developed for representing and using table syntax and explain the various viewpoints that have emerged on distinguishing form from meaning in tables.

In any actual project, the source of the tables to be processed is of major concern. We examine plausible sources of tables, including publications, business records, electronic mail, and the Web. It is important to keep in mind that most tables represent recycled data. They are prepared for publication or posting (Figure 4), for broadcasting to a select group (Figure 6), or for communication between individuals (Figure 3).

In this section we also discuss three operations on tables: table spotting, table-similarity detection, and the extraction of structure and content. The latter may be initiated with layout analysis, or by analysis of the content of leaf cells. The economics favor automated processing of batches of similar tables, but the interpretation of a mixed stream of tables of various formats is more challenging.

Among the interesting aspects of table processing is the human-computer interface to both the table extraction software and the downstream data utilization. The interface depends, of course, on the application. At the top level, we differentiate between query applications and database update applications. We consider graphical and audio access as a function of the modality (electronic or symbolic) of the table.

The penultimate section reveals half-a-dozen applications that would result from new developments in multimedia table processing.

In the last section, we summarize potential applications of table processing and appropriate research directions, and append a bibliography of the topics discussed. (At the time of this writing, another excellent bibliography, compiled by Hurst, could be found on-line at http://www.cogsci.ed.ac.uk/∼matth/research/tables/bibliography/biblio.html.) An appendix of illustrative examples completes the paper.

2 Media

We consider tables that appear in three different media: paper, electronic, and symbolic. (We are still looking for better terminology.)

2.1 Paper tables

Paper tables are usually typeset (Figure 7), or typewritten (Figure 5), or computer-generated (Figure 2). In principle they can also be hand-printed or drafted (like telephone-company drawings [2, 5, 3, 4, 6, 8, 9], and the header-blocks of old engineering drawings), but we deem such hand-drawn tables as more akin to forms and exclude them from consideration here.

Paper tables are converted to digital form by optical scanning. Printed tables are typically scanned at sampling rates of 200 to 600 dpi, but for some applications facsimile scans (100×200 dpi) may be important. High-speed duplex scanners have a throughput of 100 pages per minute at 300 dpi and 24-bit color depth.

Copying and scanning may introduce noise and skew. Both of these are more effectively corrected on a gray-level representation of the page. Image-reparation software is available from many vendors, including Lead Technologies, Mitek, Visual Image, Cardiff, and Captiva. The majority of the published work on table processing deals with the extraction of structure from scanned paper tables [1, 5, 8, 17, 22, 25, 28, 35, 52, 55, 61].

2.2 Electronic tables

Tables in plain text format may appear in e-mail (Figure 6) or on Web pages prepared by amateurs. The structure of the table is represented only by ASCII symbols for space (blanks), tab characters, and carriage returns. Occasionally printable ASCII symbols are used to show horizontal and vertical rules.

Electronic tables tend to be smaller and simpler than paper tables (but for a counter-example, see Figure 8). The amount of detail that can be displayed on a typical monitor is less than one tenth of what can be seen on a typeset page.

Mark-up languages like SGML, HTML, and XML have special conventions for tables, but there is no assurance that table tags are not abused or misused. Page composition languages have elaborate facilities for formatting tables, like TROFF Tbl [36] and the LaTeX table and array environments [34]. Many other table composition systems are surveyed in [54].

MS-Word has a table formatting subsystem and provides interconversion between tables in plain-text, Word-table, Rich Text Format (RTF) and Excel spreadsheets. FrameMaker offers PDF (Portable Document Format) for posting tables on the Web in non-editable form, and XML for applications where the structure needs to be accessible. Adobe is proposing PGML (Precision Graphics Makeup Language) for combining the benefits of PDF and XML. VXML is a proposed general-purpose format for audio access to Web documents.

Tables may also be reproduced in any raster image format, such as TIF or GIF, or rendered directly in PostScript [43]. Although directly-generated tables in image format may look superficially like scanned paper tables, they are not affected by noise or skew.

Among references that address electronic tables are [12, 27, 41, 42].

2.3 Symbolic tables

Arrays of structured data are often manipulated in spreadsheets like Lotus or Excel. A spreadsheet is a two or three dimensional array of named cells. Operations on selected cells, rows, and columns can be specified by formulas. Note, however, that spreadsheets carry only minimal semantics of the data they contain.

The most complete representation of structured data is undoubtedly a DBMS. Database systems have evolved from the Codasyl standard to Codd's relational formulation to the currently favored object-oriented paradigms. The modification and retrieval of database entries is accomplished through a language like SQL (the Structured Query Language) which is essentially isomorphic to the mathematically trustworthy relational algebra.

Although database systems have facilities for representing a very general set of relations, the appropriate relations and operations can be programmed directly for a specific application. Table operations can be readily coded in a general purpose language like C or C++, or in matrix-oriented languages like APL (A Programming Language) or M (Matlab).

For humans, paper tables are the easiest to assimilate, and symbolic tables, the hardest. For computers, it is the opposite.

3 Format

A representative set of layout conventions for tables may be found in [10], but such conventions address table synthesis rather than analysis. Effective table layout involves many considerations: see [21, 54, 57, 58, 59]. Good table layout is transparent and does not draw attention to itself. Even though most of us have been dealing with tables since the first grade, the complexity and variety of table formats tends not to be appreciated without explicitly looking for it in a large heterogeneous collection. A definitive reference on visual information display in general is [51]. While the emphasis here is primarily on graphics, table design receives some consideration as well.

The most striking aspect of tables is the isothetic (horizontal and vertical) arrangement of cells. Most of the conventions are directed at distinguishing the row and column headings from the leaf cells, and separating the cells from one another. It is convenient to distinguish the notion of header cells from that of leaf cells.

Peterman *et al.* consider a table a collection of five types of cells: *data*, *vertical indices*, *horizontal indices*, *title*, and *footnotes*. They present a detailed analysis of "table topology," *i.e.*, the conventions governing the layout of cells, and of the placement of data within the cells. The contents of each cell are analyzed by string matching to discover cells with similar letter syntax. The resulting rules for determining the "reading order" of the table are embodied in a PERL script. They present experimental results on a heterogeneous corpus of 100 electronic tables that they suggest mimic the results of processing typeset paper tables with 99% accurate OCR. It is clear that even aside from possible OCR and image processing errors, manual editing would be required for most applications [41].

3.1 Demarcations

In printed tables horizontal and vertical boxing and rules are often used for separating entries. Boxing differs from rules in the appearance of explicit corners between horizontal and vertical lines. Professional layout practice dictates using boxing and rules (especially vertical rules) only to the extent desirable to avoid ambiguity: white spaces make for easier reading (see Figures 7 and 8). Image processing techniques for the extraction of line segments include the Hough Transform [52], thinning, vectorization [1] and projection profiles [28]. Turolla *et al.* succeed in detecting 95% of 11,513 lines in 114 tables.

In many tables, the cells can be isolated by considering the horizontal and vertical channels of the background (white spaces) [8]. Inter-row spaces may be quite narrow, but inter-column spaces are typically wider than interword spaces. Multiple, regularly spaced vertical channels are the leading clue to the presence of a table surrounded by text or graphics. However, short passages of poorly typeset text may also display vertical white runs that are easily mistaken for table separators. Depending on the content of the cells, vertical columns may be left-justified, right-justified, or centered. The leaf cells may (Figure 7) or may not (Figure 2) follow the justification used in the header cells.

When a table is fully boxed, the minimal cycles of the corresponding graph can be used to locate the entries [52]. Itonori combines information derived from rulings and from the position of the text blocks [28].

3.2 Headers

Vertical headers may be distinguished by their position, explicit ruling or boxing, and larger or bolder type. They may span several columns and in such a case are often centered above them (or rarely, as in Figure 5, below). Horizontal headers may span several rows, in which case they may be above the top row (Figure 7), aligned with the top row, or vertically centered among the rows.

The vertical headers are called the "boxhead," and the horizontal headers are called the "stub." The content cells form the body of the table, and any rectangular configuration of cells is a "block" [10]. Headers are called "labels" in [54], "spanning labels" in [27], "indices" in [41], "headings" in [8], and "captions" in [42].

3.3 Leaf cells

The simplest tables have no column or row headers and physically constitute an $m \times n$ array. More complex tables have one or more levels of column or row headers which may correspond to single or multiple columns or rows. The most complex tables are nested, *i.e.*, a cell may be replaced by an entire table. (Does Figure 4 consist of several tables or one nested table?)

Single columns may be split proceeding down, and single rows may be split proceeding to the right. The merging of columns and rows is rare, except in the case of nested tables. Entire tables are sometimes split to accommodate page size. Logically merging such tables may require semantic analysis.

The presence of data items that do not fit on a single line of a cell complicates the analysis of both paper and electronic tables (Figures 4 and 2). In such a case, it is necessary to distinguish line-wrapped data from multiple rows of cells. Useful clues include hyphenation, indentation, and morphological homogeneity. Often, not every cell within a row contains line-wrapped data. In that case over-segmentation by horizontal separation reveals empty cells that are an artifact of multi-line cells. Cell alignment among sparse columns is addressed in [22] (Figure 5 is an egregious case).

As mentioned, in scanned paper tables error-prone OCR must be used to recover the contents of every cell. The other difficulties that distinguish format extraction in scanned paper tables from that in electronic tables are the potential presence of global or local skew, and overlap between boxing or ruling and alphanumeric cell contents.

Box-driven reasoning is proposed in [23] to mitigate content-separator overlaps. Instead of seeking the intersection of horizontal and vertical lines, inner (white) and outer (black) bounding boxes constitute the lowest-level structure analyzed. The proposed underlying model is described only as follows: "A table-form document is a type of form composed of strings and cells made from vertical and horizontal lines." The system was tried only on 10 fairly complex forms, and only the timing results are given in detail.

Skew may be handled by rotating the table to the nominal orientation, but the necessary pixel mapping often distorts both graphics and characters. A better strategy is to determine the skew, and perform further processing parallel to the skewed axes, or to perform skew correction on the gray-scale scan. Techniques for segmenting boxes and ruling from alphanumeric data have been developed in both the engineering drawing analysis and the form processing communities. A fast and elegant method of finding nearly horizontal and vertical rules in large run-length encoded tables appears in [9].

3.4 Models

In general, format extraction is greatly simplified by the availability of a model for the tables being analyzed [18, 17, 19, 16, 55]. The model ideally includes a database of cell contents and cell relations that can be drawn upon to resolve ambiguities. Such a model may either be specified by an operator explicitly for every batch of similar tables, or it can be derived from consistency constraints with the database and from any operator-entered corrections. We favor the latter approach because it requires less skilled operators and potentially decreases the amount of operator interaction necessary as more and more similar tables are processed.

Building on extensive previous work, Rus and Subramanian offer an interactive method of building models consisting of modular interactive agents for information access and capture in distributed databases [46]. They give examples of structure detectors and segmentation modules for both paper and electronic tables. These modules subdivide documents according to prevalent white spaces and match table rows by syntactic string matching. In an interesting digression, they predict the probability of incidental white streams from word length statistics.

4 Deep Structure

All researchers struggle with the distinction between physical and logical structure in tables. The problem is that, unlike the case in text, the 2-D layout reveals certain relationships among items that evidently belong to the logical structure. Different researchers draw the line at different points of table analysis or synthesis. Some of the dichotomies we have noted are shown in Table 1.

metadata	$\Longleftrightarrow$	data
content	$\Longleftrightarrow$	format
logical structure	$\Longleftrightarrow$	physical structure
denotational view	$\Longleftrightarrow$	functional view
abstract table	$\Longleftrightarrow$	concrete table
structure	$\Longleftrightarrow$	layout
model	$\Longleftrightarrow$	instance
semantics	$\Longleftrightarrow$	syntax
relational structure	$\Longleftrightarrow$	geometry, topology

Table 1: Logical/physical dichotomies in table structure.

We choose to approach table structure from the relational database perspective. We consider the deep structure of a table as a set of tuples of attributes. Each column header is the name of an attribute, and each row has a unique key. Table interpretation is the recovery of the deep structure from a surface (paper or electronic) representation. The relational view is also advocated in [27], where it is called a "denotational" view, while the reading order, or physical arrangement, is called a "functional" view.

Unfortunately, in physical tables the names of the attributes are often not given explicitly. For instance, a column of names and corresponding telephone number may not be headed by "NAME" and "TELNO." Furthermore, it is understood that if multiple workers may share a telephone number, then the name, rather than the telephone number is the key.

Multiple column headers, where the top header subsumes several headers at the next level, are common:

NAME		ADDRESS					TELNO		
First	Last	#	Street	City	State	Zip-code	Area-Code	#	Extension

Even when present, the attribute names in tables are often assigned far more casually than in a database system, and therefore less consistency can be expected. Explicit identification of keys is rare. Tables may be incomplete, poorly composed, and contain erroneous labels and entries. Errors may also be introduced, of course, in the analysis process.

While in a relational table the order of the tuples and attributes is immaterial, tables are often presented in a sorted order. Whether this order needs to be preserved cannot be determined from a purely syntactic analysis and depends on the application. For instance, the names and telephone number of some job candidates may convey only the access information, or the names may already be ranked according to test or interview results.

Once a table is in relational form, we know everything we can about it. We therefore consider the relational paradigm as a possible target representation for paper and electronic tables. In the words of Hurst and Douglas [27]:

> "Once the relational structure of the table is known it can be manipulated for many purposes. Smart editors can allow restructuring of tables on demand to reflect different functional views, while keeping track of the underlying semantics. Also, very different presentation formats could be generated. A presentation of tables for, *e.g.*, telephone interfaces to Web pages, ought to reflect the information access structure of a table [60] rather than its physical structure; we are working on automatically recasting tables as specifications in information-seeking dialogs."

Other formal paradigms for describing the structure of tables are the Table Syntax [17, 16, 31, 32], the Structure Description Tree [55], and the Cohesion Domain Template [27]. All three model only local horizontal and vertical adjacency relationships between cells and aim at finding an appropriate tiling of the table. The foundations for a more sophisticated scheme are laid in [26].

Haas models tables with OSM (Object-oriented Systems Modeling) [20], a formal method of analysis developed by Embley, Kurtz and Woodfield that features an elaborate graphical interface [13].

A useful target representation is the widely used spreadsheet numbering system [37, 38]. For example, the second leaf cell in the second column of a three-column table with a single top level heading (A1) and three column headings (A1A2A1, A1A2B1, A1A2C1) is called A1A3B1A2. The notation uses wild cards to reference entire rows or columns. Green generates such a spreadsheet-like description from paper tables [16]. His test set contained 60 300-dpi tables of three types.

Abu-Tarif vectorizes paper tables, then converts them first to X-Y trees [40], and the X-Y trees to actual Excel spreadsheets using Excel macros [1]. The spreadsheet or equivalent X-Y tree organization (of successive horizontal and vertical "cuts") does not contain the same level of semantic information as the relational paradigm, but such a surface representation may suffice for some applications. Evidently, automated conversion from DBMS to Spreadsheet requires fewer assumptions or specifications than the other way.

Known (model-based) domain dependency relationships between cells can be exploited for validating an interpretation. Some examples are given in [55].

The distinction between logical and physical structure is perhaps best formalized in [54]. Wang defines an abstract table as an abstract data type, and its layout structure as the presentation form of a table. The logical structure consists of entries and labels. Labels are hierarchically divided into categories and subcategories, and each entry is associated with one label from each of the categories. The organization of the labels is called a frame, and the number of categories in the frame is the dimension of the abstract table. The size of the table is the total number of entries. A concrete table is generated by applying a layout specification (topological and style rules) to an abstract table, where "topology" is the arrangement of tabular items in two dimensions, and "style" governs the final appearance of the tabular components. Based on these constructs, Wang implemented X-Table, a practical table composition in a Unix X-Windows environment.

Wang points out that the basic difference between relational tables and her abstract tables is the logical dimension. A database table is two-dimensional with attributes in one dimension and tuples in the other. To represent an abstract table with attributes in a relational database, one must determine which category corresponds to attribute names, which category corresponds to the primary keys, and which category corresponds to the non-primary keys.

The object oriented dot notation *label1.label2.label3.entry* is used by both Wang and Hodge to represent a path in the tree between headers and leaf cells. It is also the basis of the Dewey Decimal System used in library catalogs.

None of the existing table interpretation systems bridge the gap completely between layout and logical structure.

5 Table Data Collection

In spite of the prevalence of tables, it is not easy to obtain a corpus of table data that would allow one to make reasonable predictions about the performance of a system in any particular application. Because of the variability of table formats and structures, processing only a few hundred or thousand tables is bound to give a very optimistic view.

Wang collected 886 tables from five sources and showed that X-Table could represent the logical structure (except for footnotes) of 97% and the layout of 94% of these tables [54]. Considering the size, complexity and versatility of X-Table, it seems unlikely that any automated table processing system can achieve such results. However, this collection of tables may serve as a good benchmark for non-model-driven systems.

A much more homogeneous collection of similar size (851 tables) is part of the Federal Register database distributed on CD-ROM by NIST [14]. The CD-ROM includes over 4,000 page images scanned bi-level at 400 dpi, and also contains the corresponding ground-truth extracted by combining the original typesetting data with OCR results. We are not aware of any attempt to apply table-processing techniques to this data.

Paper tables are readily found in technical journals, in data books, and even in popular magazines. Within a single journal or data book, there is some consistency of format and structure. Compendia, such as a Physics or Chemistry handbook, which are edited by dozens of specialized editors, exhibit a much larger variety of tables.

In the ASCII domain, Pyreddy and Croft report on a table extraction and retrieval experiment involving $6,509$ tables from a corpus consisting of six years of text from the Wall Street Journal [42]. This data, professionally written and from a single source, is likely to be unrealistically uniform, however. In DIA and OCR, researchers at DFKI and ISRI among others found it surprisingly difficult to collect a representative collection of business letters. Since the occurrence of tables in e-mail seems infrequent, we may anticipate much greater difficulty in collecting a broad enough range of examples.

Organizations post on the Web quantities of tabular data in various formats, and some offer subscriptions that provide time-critical data, like low-cost flight opportunities, by e-mail. Again, assembling a statistically useful collection of tables from such sources is not a trivial matter.

5.1 Table spotting

Table spotting really consists of two related tasks: detecting the presence of a table, and delineating the table. (Do the first three text lines belong to the table in Figure 6?) Both are complicated by the lack of general agreement of just what constitutes a table, and how much of the "secondary" information (*e.g.*, title, caption, footnotes) belongs to the table.

In some examples, it may be equally appropriate to classify a region as table or text, as demonstrated below:

table	text
graphics	table

Tables often constitute one of the designated categories, along with text, line-drawings, half-tones, and references, in digital image analysis of technical material [39]. Large, complex tables are easy to separate from the other classes, although some illustrations can be equally accurately designated table or line drawing [22]. A method of measuring the success rate of OCR systems to identify tables and thus avoid de-columnizing them is described in [29].

The detection of tables in electronic text can be accomplished with a high degree of accuracy with relatively simple techniques. The most important clue is the presence of correlated sequences of spaces on consecutive lines [24]. A variety of heuristics found to work well on a corpus of articles from the Wall Street Journal are described in [42]. The results reported for this last study, measured on 100 documents containing a total of 50 tables, show a miss rate of 1.8% and a false-hit rate of 5.4% when measured on a per-line basis.

A graph-grammar rewrite-rule based approach to spotting electronic tables in PostScript or Interpress is described in [43]. This approach seems a bit top-heavy and may be more appropriate for general page layout analysis.

5.2 Table-similarity detection

The clustering of tables to be processed into similarity groups would allow decreasing the amount of human interaction necessary for complete and accurate data extraction. While mixed forms are routinely processed in the form processing industry, this is an essentially model-driven approach where all form types are known ahead of time (except for a "reject" category).

The grouping of tables according to similarity in format, structure and content is potentially an interesting research project with little previous work reported in the literature (except for [55]).

5.3 Table content and structure extraction

Research conducted to date differs as much by the choice of tables to investigate as by the amount and form of the extracted information.

One of the first published papers [35] concentrates on locating the ruling-line structure to extract the corner coordinates of all the cells in the table. In addition, the coordinates of the bounding boxes of the text within each cell are found. No attempt is made towards further semantic interpretation.

These notions based on a table grid and simple and compound cells are taken up again in 1997 with little change except for an emphasis on horizontal and vertical profile analysis [61]. Although the proposed methods are apparently incorporated in a commercial OCR product and have been widely tested, no experimental results are presented. Here, too, there is no attempt at interpretation.

Some recent work addresses electronic rather than paper tables. A small experiment with 29 training tables and 4 test tables marked up in SGML is described in [27]. In a slightly larger-scale study involving 50 ASCII tables taken from the Wall Street Journal, 9.4% of caption lines were mis-tagged as table lines, while 7.4% of table lines were mis-tagged as caption lines [42]. No attempt was made to distinguish structure any finer than this.

T-Recs (Table REcognition System), an elaborate program for the structural analysis of ASCII tables based on bottom-up clustering of words, is described in [30]. A demo was available on the Web at the time this report was written (http://www.dfki.uni-kl.de/da/kieni/t_recs/).

Table interpretation systems that we have already discussed are [22, 23, 55]. Work at RPI on extracting useful information from scanned paper tables is reported in [1, 16, 25, 33].

6 User Interfaces and Interaction

The reading and understanding of tables has been subjected to a surprising amount of study. Wang summarizes the three cognitive processes that are considered important in interacting with a table [54]:

1. A comprehension process, needed for understanding the principle on which the table is organized to grasp the underlying logical structure of the table.
2. A search process, needed for locating the relevant information within the table.
3. An interpretive and comparative process, needed to answer specific questions after the relevant information has been obtained.

Any automated table interpretation system will make some errors. In most applications, these errors need to be corrected by the user or operator. The errors are necessarily unpredictable, therefore a flexible user interface is required to correct them. In fact, the user interface must be powerful enough to enter any table from scratch, because the automated system may fail at any point. Consequently, it makes sense to begin the development of an automated table interpretation system by implementing a completely manual table composition system, and gradually automating the easiest and most repetitive functions.

The user interface may address the logical structure or the physical structure. According to Wang, three types of operators are necessary to manipulate the logical structure of an abstract table:

1. change logical dimension
2. reorganize the label structure of categories
3. update labels and entry values

Wang defines 18 operations for these purposes and claims that this set is both complete and non-redundant. A much larger and more complex set of rules is required for topological and style specification of the concrete table. While Wang's X-Table is a table composition system rather than a table interpretation system, the complexity and variability of composition is bound to be reflected in the interpretation.

6.1 Graphical query interface

We conjecture that a well-designed interface may be used for addressing queries to a set of (hidden) tables even without elaborate table understanding routines. The key here is exploiting the user's understanding of the expected form and content of the tables. The amount of table processing necessary is an open question, but is sure to be less of a hurdle for electronic tables than for paper tables.

Rus and Subramanian describe a sophisticated, agent-based system that extracts tabular information (*e.g.*, stock prices) from Usenet newsgroup postings and presents the results to the user as, say, a time-series chart [46]. Such an approach has undeniable appeal, although it is not clear how difficult it would be for an average user to construct queries in this environment, nor is it obvious what kind of impact recognition errors, which are unavoidable, would have.

Rao and Card's Table Lens is a collection of visualization techniques for rendering certain kinds of tabular information graphically, thereby making it possible to fit more on a given display [44]. The FOCUS interface developed by Spenke *et al.* facilitates browsing large object-attribute tables (tables with hundreds of rows and/or columns) through a fisheye method and also supports dynamic queries [48]. Neither of these systems addresses badly formed input.

The interface for interacting with the results of table processing will undoubtedly need to be extremely flexible and forgiving of failures. Some combination of a rich (but simple) query language along with powerful support for navigation via browsing stands the best chance of success in our opinion.

Moreover, the interface must be designed to convey feedback back to the system so that errors in recognition may be identified and possibly recovered from. Almost none of the work we have seen addresses the question of correcting residual errors from table processing. Green measures the number of mouse clicks required to correct errors [16]. Kornfeld and Wattecamps describe using a programming-language-like debugger that allows an operator to locate and correct errors [31].

6.2 Audio interface

Walker *et al.* describe a spoken language interface (SLI) for general e-mail navigation [53]. The emphasis is on (man-machine) dialog issues – no mention is made of using document structure to facilitate information access (other than the standard notion of filing related messages in the same folder). DeHaemer *et al.* examine automated speech recognition (ASR) for spreadsheet tasks [11]. They found that, at the time, keyboard input was more efficient than ASR, but that users were still interested in the idea of a voice interface for such activities.

For an audio interface to tables in e-mail, there are two extreme scenarios with everything in between. The two boundary points are:

1. A minimal table format recognition program that just recognizes most of the leaf cells and grids the table, and a sophisticated interactive navigation and query interface.

2. A full-fledged autonomous table interpretation system that lets the user access only the data and structure already extracted from the table.

We believe that the first alternative may be superior with regard to the amount of functionality achievable with any reasonable expenditure of resources. The development of a minimal table spotting and gridding system would require only a few weeks of work, whereas none of the automated systems that have been attempted (some of which are based on many person-years of work) appears to achieve usable performance. Further effort would be far better spent in improving the interactive navigation/query interface.

The major rationale offered for this claim is that the recipient of the table is likely to have a very good model of the structure and semantics of any table he or she may want to deal with. Such a model immensely facilitates extracting information. A general table processing system (as opposed to a batch system targeted to specific tables) would lack any such model.

We believe that for most tables likely to be communicated by electronic mail, the user would be able to ascertain its structure with very few probes. A probe might be "read me the top row" or "read me the leftmost column," or "give me the contents of the $(2, 2)$ cell." Once the user understands the structure of the table, more complex operations can be readily specified at the physical level.

7 Potential Applications

7.1 Large-volume, homogeneous table conversion

An example of an application in this area is the work done at AT&T/Lucent on the conversion of telephone billing statements to a usable form [47]. Although the tables may vary in format and content, all contain similar types of data that is compatible with an existing database. The database itself can be used to facilitate and validate data extraction from the tables [15]. This application is very similar to forms processing and could probably make use of advanced existing commercial software developed for this purpose.

The authors of the above paper emphasize the importance of a well-designed Graphical User Interface (GUI) to allow customization of the table-processing tools for specific formats. The use of table templates eliminates the need for elaborate structure hypotheses, and the success of the approach depends mainly on thorough preprocessing and accurate OCR.

7.2 Large-volume, mixed table conversion

This is a preliminary step for data mining from sources that are available only as paper or electronic tables. This application may require table spotting and table-similarity detection in addition to content and structure extraction.

Note that a successful approach to table understanding could be used to facilitate what is regarded as traditional information retrieval. The answers to certain kinds of queries seem most naturally expressed in tabular form. Consider, for example, the following ad hoc topic (#219) from the TREC 4 evaluation [50]:

> "How has the volume of U.S. imports of Japanese autos compared with exports of U.S. autos to Canada and Mexico?"

A document relevant to such a query will likely contain a table comparing auto imports/exports over time or by country.

7.3 Individual database creation

This is a filing application for data that arrives in e-mail, by post, or is discovered on the Web [56]. The individual sets up some goal-oriented digital filing system and populates it with items that arrive at unpredictable times. The tables are processed either as they arrive, or batched for more convenient interactive processing. An important consideration here is minimization of the original set-up time and level of skill required.

7.4 Tabular browsing

Interactively extracting specific information from a large table is somewhat similar to addressing queries to a database with a language like SQL. Wang gives examples where the results of a query consist of highlighting specific cells in a table. She also mentions the possibility of creating subtables in response to a query, which is similar to view generation in a database [54].

7.5 Audio access to tables

In the EMU project [49], it may be desirable to detect and access newly received tables in e-mail by telephone. Access may take the form of an abbreviated reading or summarization of the table, a query-answer interface directly to the table, or conversion of the table to a database and access through an existing audio-database interface (if one were to exist). A protocol for direct access to tables was devised for "talking books" for the blind [45]. It requires repeating the appropriate table heading before each content cell is voiced, which can be a slow and painful process.

7.6 Table manipulation

Existing tables often need to be reformatted, combined, or modified for specific target audiences. Such manipulation may take place at the level of format, using a word processor, page-composition language, or spreadsheet, or at the deeper level of the underlying database. The latter can use independently-developed facilities for view generation and database output formatting. This application is mentioned in [27, 54].

7.7 Table modification for display

A relatively superficial but perhaps important type of modification is that required for displaying an existing table at a different resolution than originally intended. In addition to accommodating small-format displays such as a personal digital assistant (PDA), one may wish to modify a page-width table to single-column width. Additional headers must be inserted to divide long tables to fit pages. A research issue here that may draw on database concepts is the division of tables into a set of equivalent tables (*cf.* "Large Tables" in [54]).

In this section we have attempted to break down applications into as many discrete categories as possible. It may or may not be advantageous to develop a table-processing framework that can handle several of these applications in a unified way.

8 Conclusions

We have identified several classes of potential applications for table processing and some research problems on which little work has been reported so far. We have also formed opinions of the relative difficulties of the tasks involved. The applications are:

- Large-volume, homogeneous table conversion
- Large-volume, mixed table conversion
- Individual database creation
- Tabular browsing

- Audio access to tables
- Table manipulation
- Table modification for display

The next step would be to analyze these applications to determine their commonalities and differences.

The new research problems appear to us to be:

- Query mechanisms for freeform electronic tables
- Audio navigation and access to a gridded table
- Subdividing a table into a set of equivalent tables
- Spotting tables in electronic mail
- Clustering tables into similarity groups
- Converting a paper or electronic table into an abstract representation
- Effects of "noise" in tables and correction of errors introduced in processing
- Performance evaluation of both table conversion and table query

The ways in which the applications and problems interrelate are depicted in Figure 2. Unless we make headway on performance evaluation, including acquisition of statistically adequate test material, it will be difficult to evaluate progress on any of the other tasks.

Application	*Research Problem*							
	Query mechanisms	Audio navigation	Table subdivision	Table spotting	Table clustering	Conversion to abstract form	Overcoming recognition errors	Performance evaluation
Large-volume, homogeneous conversion						•	•	•
Large-volume, mixed conversion				•	•	•	•	•
Individual database creation	•		•	•	•	•	•	•
Tabular browsing	•	•	•	•	•	•	•	•
Audio access to tables	•	•	•	•	•	•	•	•
Table manipulation			•		•	•	•	•
Table modification for display			•		•	•	•	•

Table 2: Applications and research problems in table processing and their interrelations.

Although the logical interpretation of paper and electronic tables is similar, the overhead of image processing and OCR makes the former a much more difficult task. Most work to date is based on table geometry, *i.e.*, processing the graphic elements of the table. Very little has been reported on combining such image processing with the results of character recognition of the cell contents. Current OCR systems often de-columnize tables because superficially they look like multicolumn text. No test on a large, heterogeneous corpus, has been reported, and few researchers consider the need to provide a mechanism for the correction of residual errors from automated processing.

More recently, the trend has shifted to the easier problem of electronic table conversion. Several commercial organizations advertise their capability of converting electronic tables to various forms, including spreadsheets. Some advertise conversion of tables in raster image form. We do not know what kind of a benchmark would allow testing of their claims.

Simple electronic tables, whether ASCII, PDF, RTF, SGML, HTML, XML, LATEX, Tbl, or other, can probably be converted with moderate effort to an abstract form with over 90% accuracy. Spotting large tables in electronic documents is relatively easy, but delineating them precisely is more difficult. A limit on achievable accuracy is imposed by the ambiguity inherent in these tasks.

The derivation of information from a table could be accomplished by converting the table to a relational database or equivalent and formulating queries in SQL. Alternatively, queries can be answered by direct interactive access to a preprocessed table. Such preprocessing need not be much more elaborate than division into rows and columns.

However, tables do not generally contain sufficient information for conversion into a database, although they can be converted into an abstract table or spreadsheet. To add the necessary semantics, a model of the table is required. The model can be derived from an existing database corresponding to similar tables, or it can be provided by the user/operator. The user can either provide the model explicitly, or implicitly by correcting errors. Except for large volumes of similar tables, it appears sensible to take advantage of the user's understanding of the context of the table: endowing a table-understanding system with such context is difficult.

The economics of table processing is another important point that has often been ignored. Clearly, an investment in table processing must bring with it benefits that exceed the expenses involved. If it is always easier to recover the desired information through some other means (by browsing, say, or via a simple keyword query), then table processing serves no purpose. The formulation of such a model would be invaluable, and may very well provide insight into where we should apply our efforts to obtain the greatest possible return.

We conclude by noting that the vast majority of papers published to date have concentrated either on the problems associated with low-level analysis of printed tables, or on guidelines for table presentation, with comparatively little work on the topic of making tabular information *useful* (other than for highly specialized applications). What has changed to make this an interesting question to consider? The unprecedented explosion in the amount of information people are confronted with each day. Whereas large-scale databases were once the province of a select few, nowadays anyone with Internet access and an e-mail account is inundated with vast quantities of unstructured (or at best loosely structured) data. Automated table processing presents one promising way of recovering useful, familiar structure making it possible to realize more of the benefits of universal data access.

9 Acknowledgements

The authors would like to thank Jianying Hu, Gordon Wilfong, and Thomas Kieninger for their helpful comments on an earlier draft of this paper.

References

[1] A. Abu-Tarif. Table processing and table understanding. Master's thesis, Rensselaer Polytechnic Institute, May 1998.

[2] J. F. Arias, S. Balasubramanian, A. Prasad, R. Kasturi, and A. Chhabra. Information extraction from telephone company drawings. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 729–732, Seattle, Washington, June 1994.

[3] J. F. Arias, A. Chhabra, and V. Misra. Efficient interpretation of tabular documents. In *Proceedings of the International Conference on Pattern Recognition (ICPR'96)*, volume III, pages 681–685, Vienna, Austria, August 1996.

[4] J. F. Arias, A. Chhabra, and V. Misra. Interpreting and representing tabular documents. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 600–605, San Francisco, CA, June 1996.

[5] J. F. Arias and R. Kasturi. Efficient techniques for line drawing interpretation and their application to telephone company drawings. Technical Report CSE TR CSE-95-020, Penn State University, August 1995.

[6] S. Balasubramanian, S. Chandran, J. F. Arias, R. Kasturi, and A. Chhabra. Information extraction from tabular drawings. In *Proceedings of Document Recognition I (IS&T/SPIE Electronic Imaging'94)*, volume 2181, pages 152–163, San Jose, CA, June 1994.

[7] L. Bing, J. Zao, and X. Hong. New method for logical structure extraction of form document image. In *Proceedings of Document Recognition and Retrieval VI (IS&T/SPIE Electronic Imaging'99)*, volume 3651, pages 183–193, San Jose, CA, January 1999.

[8] S. Chandran and R. Kasturi. Structural recognition of tabulated data. In *Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 516–519, Tsukuba Science City, Japan, October 1993.

[9] A. K. Chhabra, V. Misra, and J. Arias. Detection of horizontal lines in noisy run length encoded images: The FAST method. In R. Kasturi and K. Tombre, editors, *Graphics Recognition – Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*, pages 35–48. Springer-Verlag, Berlin, Germany, 1996.

[10] *The Chicago Manual of Style.* The University of Chicago Press, 1982.

[11] M. J. DeHaemer, G. Wright, and T. W. Dillon. Automated speech recognition for spreadsheet tasks: Performance effects for experts and novices. *International Journal of Human-Computer Interaction*, 6(3):299–318, 1994.

[12] S. Douglas, M. Hurst, and D. Quinn. Using natural language processing for identifying and interpreting tables in plain text. In *Proceedings of the Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, pages 535–545, Las Vegas, NV, April 1995.

[13] D. Embley, B. Kurtz, and S. Woodfield. *Object-oriented Systems Analysis: A Model Driven Apprach.* Yourdon Press, 1992.

[14] M. Garris, S. Janet, and W. Klein. Federal Register document image database. In *Proceedings of Document Recognition and Retrieval VI (IS&T/SPIE Electronic Imaging'99)*, volume 3651, pages 97–108, San Jose, CA, January 1999.

[15] P. Gray, S. Embury, W. Gray, and K. Hui. An agent-based system for handling distributed design constraints. In *Proceedings of Agents'98*, 1998.

[16] E. A. Green. *Model-based analysis of printed tables.* PhD thesis, Rensselaer Polytechnic Institute, May 1996.

[17] E. A. Green and M. Krishnamoorthy. Model-based analysis of printed tables. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 214–217, Montréal, Canada, August 1995.

[18] E. A. Green and M. Krishnamoorthy. Model-based analysis of printed tables. In *Proceedings of the First International Workshop on Graphics Recognition (GREC'95)*, pages 234–242, PA, 1995.

[19] E. A. Green and M. Krishnamoorthy. Recognition of tables using table grammars. In *Proceedings of the Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, pages 261–277, Las Vegas, NV, April 1995.

[20] T. B. Haas. The development of a prototype knowledge-based table-processing system. Master's thesis, Brigham Young University, December 1997.

[21] R. Hall. *Handbook of Tabular Presentation.* The Ronald Press Company, New York, NY, 1943.

[22] Y. Hirayama. A method for table structure analysis using DP matching. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 583–586, Montréal, Canada, August 1995.

[23] O. Hori and D. S. Doermann. Robust table-form structure analysis based on box-driven reasoning. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 218–221, Montréal, Canada, August 1995.

[24] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Table detection across multiple media. In *Proceedings of the Workshop on Document Layout Interpretation and its Applications (DLIA'99)*, Bangalore, India, September 1999. To appear.

[25] T. Hu. Recognizing table entries in a scanned document. Master's thesis, Rensselaer Polytechnic Institute, October 1993.

[26] M. Hurst. Layout and language: Beyond simple text for information interaction – modelling the table. In *Proceedings of the Second International Conference on Multimodal Interfaces*, Hong Kong, January 1999.

[27] M. Hurst and S. Douglas. Layout and language: Preliminary investigations in recognizing the structure of tables. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 1043–1047, August 1997.

[28] K. Itonori. A table structure recongnition based on textblock arrangement and ruled line position. In *Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 765–768, Tsukuba Science City, Japan, October 1993.

[29] J. Kanai, G. Nagy, T. Nartker, and S. Rice. Automated evaluation of OCR zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):86–90, 1995.

[30] T. G. Kieninger. Table structure recognition based on robust block segmentation. In *Proceedings of Document Recognition V (IS&T/SPIE Electronic Imaging'98)*, volume 3305, pages 22–32, San Jose, CA, January 1998.

[31] W. Kornfeld and J. Wattecamps. Automatically locating, extracting and analyzing tabular data. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–348, Melbourne, Australia, August 1998.

[32] M. Krishnamoorthy. TBL, an easy to use table description language. Internal document, Rensselaer Polytechnic Institute, 1992.

[33] G. Kyriazis. Analysis of digitized tables. Senior project report, Rensselaer Polytechnic Institute, 1990.

[34] L. Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, MA, 1985.

[35] A. Laurentini and P. Viada. Identifying and understanding tabular material in compound documents. In *Proceedings of the Eleventh International Conference on Pattern Recognition (ICPR'92)*, pages 405–409, The Hague, 1992.

[36] M. Lesk. Tbl – a program to format tables. In *UNIX Programmer's Manual*, volume 2A. Bell Telephone Laboratories, Murray Hill, NJ, 1979.

[37] *Lotus 1-2-3 User's Handbook.* Ballantine Books, New York, NY, 1984.

[38] *Microsoft Excel User's Guide.* Microsoft Corporation, Redmond, WA, 1990.

[39] G. Nagy, M. Krishnamoorthy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7):737–747, 1993.

[40] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proceedings the International Conference on Pattern Recognition (ICPR)*, pages 347–349, 1984.

[41] C. Peterman, C. H. Chang, and H. Alam. A system for table understanding. In *Proceedings of the Symposium on Document Image Understanding Technology (SDIUT'97)*, pages 55–62, Annapolis, MD, April/May 1997.

[42] P. Pyreddy and W. B. Croft. TINTIN: A system for retrieval in text tables. Technical Report UM-CS-1997-002, University of Massachusetts, Amherst, January 1997.

[43] M. A. Rahgozar and R. Cooperman. A graph-based table recognition system. In *Proceedings of Document Recognition III (IS&T/SPIE Electronic Imaging'96)*, volume 2660, pages 192–203, San Jose, CA, January 1996.

[44] R. Rao and S. K. Card. The Table Lens: Merging gaphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'94)*, pages 318–322, Boston, MA, April 1994.

[45] *The 1.7 Tag Set Usage Guide.* Recording for the Blind and Dyslexic, Princeton, NJ, 1994.

[46] D. Rus and D. Subramanian. Customizing information capture and access. *ACM Transactions on Information Systems*, 15(1):67–101, 1997.

[47] J. H. Shamalian, H. S. Baird, and T. L. Wood. A retargetable table reader. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 158–163, August 1997.

[48] M. Spenke, C. Beilken, and T. Berlage. FOCUS: The interactive table for product comparison and selection. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'96)*, pages 41–50, Seattle, WA, November 1996.

[49] R. Sproat, J. Hu, and H. Chen. EMU: an e-mail preprocessor for text-to-speech. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pages 239–244, Los Angeles, CA, December 1998.

[50] TREC Data – English Test Questions (Topics). http://trec.nist.gov/data/testq_eng.html.

[51] E. R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, 1983.

[52] E. Turolla, Y. Belaid, and A. Belaid. Form item extraction based on line searching. In R. Kasturi and K. Tombre, editors, *Graphics Recognition – Methods and Applications*, volume 1072 of *Lecture Notes in Computer Science*, pages 69–79. Springer-Verlag, Berlin, Germany, 1996.

[53] M. A. Walker, J. Fromer, G. D. Fabbrizio, C. Mestel, and D. Hindle. What can I say?: Evaluating a spoken language interface to email. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 582–589, Los Angeles, CA, April 1998.

[54] X. Wang. *Tabular abstraction, editing, and formatting.* PhD thesis, University of Waterloo, 1996.

[55] T. Watanabe, Q. L. Quo, and N. Sugie. Layout recognition of multi-kinds of table-form documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):432–445, 1995.

[56] S. Whittaker and C. Sidner. Email overload: exploring personal information management of email. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 276–283, Vancouver, British Columbia, Canada, April 1996.

[57] P. Wright. Using tabulated information. *Ergonomics*, 11(4):331–343, 1968.

[58] P. Wright. Understanding tabular displays. *Visible Language*, 7:351–359, 1973.

[59] P. Wright. The comprehension of tabulated information: some similarities between prose and reading tables. *NSPI Journal*, XIX(8):25–29, October 1980.

[60] P. Wright. A user-oriented approach to the design of tables and flowcharts. In D. H. Jonassen, editor, *The Technology of Text.* Educational Technology Publications, 1982.

[61] K. Zuyev. Table image segmentation. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 705–708, August 1997.

A Table Examples

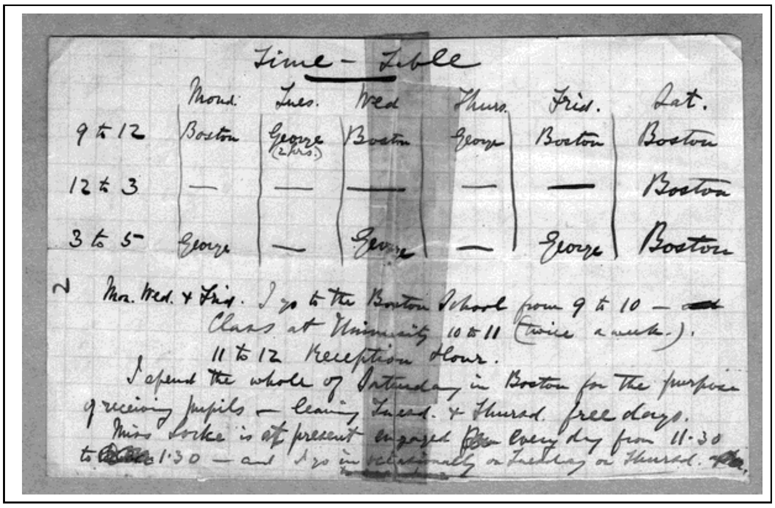
In this appendix, we present a number of examples of paper and electronic tables.

TABLE 1
A BRIEF SURVEY OF GEOMETRIC PAGE-LAYOUT ANALYSIS METHODS

No.	Author	Year	Approach	Features
1	Wahl et al. [11]	1982	Run length smoothing	Time consuming and skew sensitive
2	Nagy et al. [12]	1984	X-Y tree cut	Skew sensitive; Assumes rectangular blocks
3	Wang et al. [13]	1989	Run length smoothing and recursive X-Y cut	Newspaper analysis; Sensitive to skew
4	Fujisawa et al. [14]	1990	Top-down	Japanese patent documents
5	Fisher et al. [15]	1990	Run length smoothing and connected component extraction	Identifies text and nontext zones; Skew sensitive
6	Pavlidis et al. [16]	1991	Column oriented projection	Identifies text and nontext regions; Accommodates moderate skew
7	Baird [17]	1992	Global-to-local strategy	Accommodates different languages; Skew correction;
8	Jain et al. [18]	1992	Gabor filtering	Multichannel texture features from gray-scale images; Time consuming
9	Lebourgeois et al. [19]	1992	8× 3 window filtering	Unconstrained documents; Skew not considered
10	Pavlidis et al. [20]	1992	Horizontal smearing and bottom-up	Accommodates small skew; Fixed parameters
11	Akindele et al. [21]	1993	White space tracing	Polygonal blocks; Only text zones considered
12	Amamoto et al. [22]	1993	Morphological operation on white space	Identifies horizontal and vertical writing; Skew not considered
13	Ittner et al. [23]	1993	White space and minimum spanning tree	Language and orientation free; Large computation
14	O'Gorman [24]	1993	*k*-nearest neighbor clustering	Can handle arbitrary orientation with high accuracy; Large computation
15	Antonacopoulos et al. [25], [26]	1994	Contours from white tiles	Finds nonrectangular and skewed regions; Error in classifying large fonts
16	Zlatopolsky [27]	1994	Connected component extraction	Multiple skewed document; Sensitive parameters
17	Doermann [28]	1995	Wavelet multiscale analysis	Segments nonblock-nested pages; Gray-scale image processing; High computational complexity
18	Drivas et al. [29]	1995	Connected component grouping	Skew correction with a time consuming algorithm
19	Ha et al. [30]	1995	Connected component-based projection profile	Faster than pixel-based projection profile; Skew sensitive
20	Sylwester et al. [31]	1995	trainable X-Y cut	Relatively robust; Skew and noise free
21	Tang et al. [32]	1995	Modified fractal signature	Handles documents with high geometrical complexity; Gray-scale image processing; Time consuming
22	Jain et al. [33], [34]	1996	Masks and neural network	Handles documents with multiple languages; Gray-scale image processing; Time consuming
23	Kise et al. [35]	1996	Background thinning	Skewed nonrectangular layout; Bounding box is not very tight
24	Liu et al. [36]	1996	Adaptive top-down and bottom-up	Nonrectangular regions; Skew free
25	Yamashita et al. [37]	1996	Run length smearing and adaptive thresholding	Less sensitive to font size and spacing; Skew free

Figure 2: A table with considerable text comparing document layout analysis methods.[2]

[2]From "Document Representation and its Application to Page Decomposition" by A. K. Jain and B. Yu, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 1998, pg. 297.

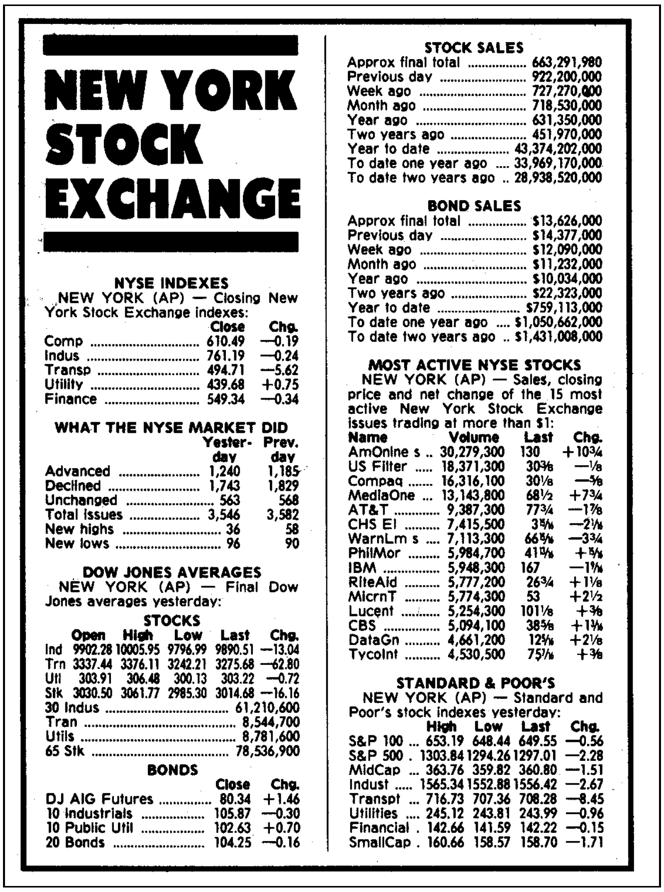


Time - Table

	Mond.	Tues.	Wed	Thurs.	Frid.	Sat.
9 to 12	Boston	George (2 hrs.)	Boston	George	Boston	Boston
12 to 3	—	—	—	—	—	Boston
3 to 5	George	—	George	—	George	Boston

Mon. Wed. & Frid. I go to the Boston School from 9 to 10 —
Class at University 10 to 11 (twice a week).
11 to 12 Reception Hour.
I spend the whole of Saturday in Boston for the purpose of receiving pupils — leaving Tuesd. & Thursd. free days.
Miss Locke is at present engaged every day from 11.30 to 1.30 — and I go in occasionally on Tuesday or Thursd.

Figure 3: A handwritten table showing a personal schedule.[3]



NEW YORK STOCK EXCHANGE

NYSE INDEXES

NEW YORK (AP) — Closing New York Stock Exchange indexes:

	Close	Chg.
Comp	610.49	—0.19
Indus	761.19	—0.24
Transp	494.71	—5.62
Utility	439.68	+0.75
Finance	549.34	—0.34

WHAT THE NYSE MARKET DID

	Yester-day	Prev. day
Advanced	1,240	1,185
Declined	1,743	1,829
Unchanged	563	568
Total issues	3,546	3,582
New highs	36	58
New lows	96	90

DOW JONES AVERAGES

NEW YORK (AP) — Final Dow Jones averages yesterday:

STOCKS

	Open	High	Low	Last	Chg.
Ind	9902.28	10005.95	9796.99	9890.51	—13.04
Trn	3337.44	3376.11	3242.21	3275.68	—62.80
Uti	303.91	306.48	300.13	303.22	—0.72
Stk	3030.50	3061.77	2985.30	3014.68	—16.16

30 Indus	61,210,600
Tran	8,544,700
Utils	8,781,600
65 Stk	78,536,900

BONDS

	Close	Chg.
DJ AIG Futures	80.34	+1.46
10 Industrials	105.87	—0.30
10 Public Util	102.63	+0.70
20 Bonds	104.25	—0.16

STOCK SALES

Approx final total	663,291,980
Previous day	922,200,000
Week ago	727,270,000
Month ago	718,530,000
Year ago	631,350,000
Two years ago	451,970,000
Year to date	43,374,202,000
To date one year ago	33,969,170,000
To date two years ago	28,938,520,000

BOND SALES

Approx final total	$13,626,000
Previous day	$14,377,000
Week ago	$12,090,000
Month ago	$11,232,000
Year ago	$10,034,000
Two years ago	$22,323,000
Year to date	$759,113,000
To date one year ago	$1,050,662,000
To date two years ago	$1,431,008,000

MOST ACTIVE NYSE STOCKS

NEW YORK (AP) — Sales, closing price and net change of the 15 most active New York Stock Exchange issues trading at more than $1:

Name	Volume	Last	Chg.
AmOnlne s	30,279,300	130	+10 3/4
US Filter	18,371,300	30 3/8	—1/8
Compaq	16,316,100	30 1/8	—5/8
MediaOne	13,143,800	68 1/2	+7 3/4
AT&T	9,387,300	77 3/4	—1 7/8
CHS El	7,415,500	3 15/16	—2 1/16
WarnLm s	7,113,300	66 15/16	—3 3/4
PhilMor	5,984,700	41 13/16	+5/16
IBM	5,948,300	167	—1 7/16
RiteAid	5,777,200	26 3/4	+1 1/8
MicrnT	5,774,300	53	+2 1/2
Lucent	5,254,300	101 1/8	+3/8
CBS	5,094,100	38 5/8	+1 3/16
DataGn	4,661,200	12 5/16	+2 1/8
TycoInt	4,530,500	75 7/16	+3/8

STANDARD & POOR'S

NEW YORK (AP) — Standard and Poor's stock indexes yesterday:

	High	Low	Last	Chg.
S&P 100	653.19	648.44	649.55	—0.56
S&P 500	1303.84	1294.26	1297.01	—2.28
MidCap	363.76	359.82	360.80	—1.51
Indust	1565.34	1552.88	1556.42	—2.67
Transpt	716.73	707.36	708.28	—8.45
Utilities	245.12	243.81	243.99	—0.96
Financial	142.66	141.59	142.22	—0.15
SmallCap	160.66	158.57	158.70	—1.71

Figure 4: Tables of daily financial results.[4]

[3]From the Library of Congress archive of the Alexander Graham Bell family papers, http://memory.loc.gov/ammem/bellhtml/bellhome.html.

[4]From *The Trenton Times*, March 23, 1999, pg. D2.

																8
																38
																31
															34	34
															86	86
															84	84
															87	87
														40	40	40
														39	39	39
														88	88	88
														81	81	81
														7	7	7
														85	85	85
													36	36	36	36
													91	91	91	91
													79	79	79	79
													76	76	76	76
												29	29	29	29	29
												37	37	37	37	37
												90	90	90	90	90
												92	92	92	92	92
												89	89	89	89	89
												83	83	83	83	83
												78	78	78	78	78
											80	80	80	80	80	80
											30	30	30	30	30	30
											33	33	33	33	33	33
											6	6	6	6	6	6
											27	27	27	27	27	27
											4	4	4	4	4	4
											28	28	28	28	28	28
											35	35	35	35	35	35
											82	82	82	82	82	82
										77	77	77	77	77	77	77
									3	3	3	3	3	3	3	3
									5	5	5	5	5	5	5	5
								32	32	32	32	32	32	32	32	32
						41	41	41	41	41	41	41	41	41	41	41
					42	42	42	42	42	42	42	42	42	42	42	42
			26	26	26	26	26	26	26	26	26	26	26	26	26	26
		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
			1917									1918				

Figure 5: A table showing the stationing of U.S. Army Divisions in France during WWI.[6]

```
********************************************************
               LUCENT TECHNOLOGIES TODAY
         For the People of Lucent Technologies
               Friday, February 12, 1999
********************************************************

                 ***  STOCK WATCH  ***

               TODAY'S   YESTERDAY'S   YESTERDAY'S
                OPEN       CLOSE         CHANGE

Lucent         100 13/16  101 1/16     + 3 13/16
Ascend         73 5/8     74 7/8       + 2 3/8
AT&T           87 1/2     88 3/16      + 2 3/8
Alcatel        21 7/8     22           + 13/16
Ericsson       26 1/4     26 3/8       + 1 5/16
Motorola       67 1/2     67 1/4       + 1 13/16
DJIA           9367.32    9363.46      + 186.15
NASDAQ         2375.99    2405.55      + 96.05

********************************************************

*** NEWS IN A NUTSHELL ***      *** LUCENT HERITAGE ***

* New software tool             On Feb. 17, 1998, Lucent
* America's most admired        announced that it would
* Switch lands in winter games  acquire Hewlett-Packard's
* Students visit Bell Labs      local multipoint distribution
* World of Science Seminars     service wireless business
* Client feedback survey        and launch a new Wireless
                                Broadband Networks Division.

****************  LUCENT IN THE NEWS  *****************

STUDENTS VISIT BELL LABS -- Hosted by Lucent Korea,
elementary school students from Korea visited Bell Labs
in New Jersey to explore its advanced science and
technology.  Lucent Korea provided the six-day tour for
the students to encourage their education in science.
[Naeway Economic Daily (Korea), 2/12]
```

Figure 6: One (or perhaps two?) tables embedded in ASCII text.[7]

[6] From *The Visual Display of Quantitative Information* by Edward R. Tufte, Graphics Press: Cheshire, CT, 1983, pg. 141. Note the subtle partition between the regions governed by the first-level headers (footers), "1917" and "1918."

[7] From *Lucent Technologies Today*, February 12, 1999.

How Different Groups Voted for President

Based on 12,782 interviews with voters at their polling places. Shown is how each group divided its vote for President and, in parentheses, the percentage of the electorate belonging to each group.

	CARTER	REAGAN	ANDERSON	CARTER-FORD in 1976
Democrats (43%)	66	26	6	77 - 22
Independents (23%)	30	54	12	43 - 54
Republicans (28%)	11	84	4	9 - 90
Liberals (17%)	57	27	11	70 - 26
Moderates (46%)	42	48	8	51 - 48
Conservatives (28%)	23	71	4	29 - 70
Liberal Democrats (9%)	70	14	13	86 - 12
Moderate Democrats (22%)	66	28	6	77 - 22
Conservative Democrats (8%)	53	41	4	64 - 35
Politically active Democrats (3%)	72	19	8	—
Democrats favoring Kennedy in primaries (13%)	66	24	8	—
Liberal Independents (4%)	50	29	15	64 - 29
Moderate Independents (12%)	31	53	13	45 - 53
Conservative Independents (7%)	22	69	6	26 - 72
Liberal Republicans (2%)	25	66	9	17 - 82
Moderate Republicans (11%)	13	81	5	11 - 88
Conservative Republicans (12%)	6	91	2	6 - 93
Politically active Republicans (2%)	5	89	6	—
East (32%)	43	47	8	51 - 47
South (27%)	44	51	3	54 - 45
Midwest (20%)	41	51	6	48 - 50
West (11%)	35	52	10	46 - 51
Blacks (10%)	82	14	3	82 - 16
Hispanics (2%)	54	36	7	75 - 24
Whites(88%)	36	55	8	47 - 52
Female (49%)	45	46	7	50 - 48
Male (51%)	37	54	7	50 - 48
Female, favors equal rights amendment (22%)	54	32	11	—
Female, opposes equal rights amendment (15%)	29	66	4	—
Catholic (25%)	40	51	7	54 - 44
Jewish (5%)	45	39	14	64 - 34
Protestant (46%)	37	56	6	44 - 55
Born-again white Protestant (17%)	34	61	4	—
18 - 21 years old (6%)	44	43	11	48 - 50
22 - 29 years old (17%)	43	43	11	51 - 46
30 - 44 years old (31%)	37	54	7	49 - 49
45 - 59 years old (23%)	39	55	6	47 - 52
60 years or older (18%)	40	54	4	47 - 52
Family income				
Less than $10,000 (13%)	50	41	6	58 - 40
$10,000 - $14,999 (14%)	47	42	8	55 - 43
$15,000 - $24,999 (30%)	38	53	7	48 - 50
$25,000 - $50,000 (24%)	32	58	8	36 - 62
Over $50,000 (5%)	25	65	8	—
Professional or manager (40%)	33	56	9	41 - 57
Clerical, sales or other white-collar (11%)	42	48	8	46 - 53
Blue-collar worker (17%)	46	47	5	57 - 41
Agriculture (3%)	29	66	3	—
Looking for work (3%)	55	35	7	65 - 34
Education				
High school or less (39%)	46	48	4	57 - 43
Some college (28%)	35	55	8	51 - 49
College graduate (27%)	35	51	11	45 - 55
Labor union household (26%)	47	44	7	59 - 39
No member of household in union (62%)	35	55	8	43 - 55
Family finances				
Better off than a year ago (16%)	53	37	8	30 - 70
Same (40%)	46	46	7	51 - 49
Worse off than a year ago (34%)	25	64	8	77 - 23
Family finances and political party				
Democrats, better off than a year ago (7%)	77	16	6	69 - 31
Democrats, worse off than a year ago (13%)	47	39	10	94 - 6
Independents, better off (3%)	45	36	12	—
Independents, worse off (9%)	21	65	11	—
Republicans, better off (4%)	18	77	5	3 - 97
Republicans, worse off (11%)	6	89	4	24 - 76
More important problem				
Unemployment (39%)	51	40	7	75 - 25
Inflation (44%)	30	60	9	35 - 65
Feel that U.S. should be more forceful in dealing with Soviet Union even if it would increase the risk of war (54%)	28	64	6	—
Disagree (31%)	56	32	10	—
Favor equal rights amendment (46%)	49	38	11	
Oppose equal rights amendment (35%)	26	68	4	—
When decided about choice				
Knew all along (41%)	47	50	2	44 - 55
During the primaries (13%)	30	60	8	57 - 42
During conventions (8%)	36	55	7	51 - 48
Since Labor Day (8%)	30	54	13	49 - 49
In week before election (23%)	38	46	13	49 - 47

Source: 1976 and 1980 election day surveys by The New York Times/CBS News Poll and 1976 election day survey by NBC News.

Figure 7: A table analyzing voter preferences in the 1980 U.S. Presidential Election.[8]

[8]From *The Visual Display of Quantitative Information* by Edward R. Tufte, Graphics Press: Cheshire, CT, 1983, pg. 179. Tufte notes: "This type of elaborate table, a *supertable*, is likely to attract and intrigue readers through its organized, sequential detail and reference-like quality. One supertable is far better than a hundred little bar charts."

1996 DDVPC 2400bps Candidate Coder Intellibility Performance Calibrations

	Wgt	Wgt	Wgt	A	A	A	A	B	B	B	B	C	C	C	C	D	D	D	D
DRT	0.35			A(M)	A(F)	A(C)	A(SE)	B(M)	B(F)	B(C)	B(SE)	C(M)	C(F)	C(C)	C(SE)	D(M)	D(F)	D(C)	D(SE)
Quiet		0.20	0.100	90.8	88.9	89.9	0.55	90.1	90.8	90.5	0.48	93.8	90.8	92.3	0.70	92.7	90.1	91.4	0.71
Vinson Quiet			0.100	91.5	91.4	91.4	0.42	91.4	88.4	89.9	0.83	91.3	91.1	91.2	0.78	92.1	89.4	90.8	0.57
Office		0.60	0.067	91.8	91.2	91.5	0.91	88.6	88.2	88.4	0.55	91.5	91.0	91.2	0.75	92.4	89.8	91.1	0.59
Auto			0.067	87.9	77.6	82.7	0.93	86.3	79.2	82.7	0.91	86.8	79.9	83.4	0.81	87.5	81.3	84.4	0.88
Humvee			0.067	61.8	66.9	64.4	1.06	64.8	64.9	64.8	1.30	61.8	64.4	63.1	0.72	61.2	69.6	65.4	0.62
M2 Bradley			0.067	67.4	67.1	67.3	1.36	66.3	68.8	67.6	0.88	61.4	67.2	64.3	1.17	66.0	67.1	66.6	0.71
Helicopter			0.067	61.5	68.5	65.0	1.15	68.6	68.9	68.7	1.07	63.5	69.9	66.7	0.61	66.2	71.2	68.7	1.04
F-15			0.067	74.9	78.1	76.5	0.85	71.1	71.1	71.1	0.52	77.5	76.7	76.7	0.91	73.4	75.2	74.3	0.57
E3A			0.067	87.6	83.0	85.3	0.75	86.9	84.8	85.9	0.81	89.5	85.5	87.5	0.85	89.1	83.8	86.5	0.66
P3C			0.067	89.2	82.2	85.7	0.71	87.4	81.5	84.4	0.75	84.5	85.4	85.4	0.59	87.3	81.9	84.6	0.88
MCE			0.067	88.5	88.4	88.5	0.72	86.7	87.5	87.1	0.77	89.0	90.6	89.8	1.08	86.6	88.9	87.8	0.84
BER		0.10	0.050	86.2	82.6	84.4	0.89	88.5	85.7	87.1	0.52	88.6	89.2	88.9	0.63	87.2	86.1	86.7	0.71
BLER			0.050	92.3	89.7	91.0	0.86	92.4	89.7	91.1	0.63	93.4	90.0	91.7	0.64	90.2	88.2	89.2	0.45
S_Tandem		0.10	0.050	85.0	85.3	85.2	0.73	87.4	84.3	85.9	0.60	87.5	84.5	86.0	0.72	87.6	84.7	86.2	0.71
D_Tandem			0.050	84.1	81.9	83.0	0.86	83.0	78.0	80.5	0.62	85.2	80.6	82.9	0.58	84.3	81.2	82.8	0.75
Intell. Perf				82.984	81.872	82.437	0.261	82.829	81.132	81.984	0.293	83.279	82.779	83.032	0.241	83.259	82.214	82.759	0.217
		1.00	1.000	A(M)	A(F)	A(C)	A(SE)	B(M)	B(F)	B(C)	B(SE)	C(M)	C(F)	C(C)	C(SE)	D(M)	D(F)	D(C)	D(SE)
Rank				4	4	4		5	6	5		2	2	2		3	3	3	

1996 DDVPC 2400bps Reference Coder Intellibility Performance Calibrations

	Wgt	Wgt	Wgt	CELP	CELP	CELP	CELP	CVSD	CVSD	CVSD	CVSD	LPC	LPC	LPC	LPC
DRT	0.35			Celp(M)	Celp(F)	Celp(C)	Celp(SE	KY(M)	KY(F)	KY(C)	KY(SE)	LPC(M)	LPC(F)	LPC(C)	LPC(SE)
Quiet		0.20	0.100	90.9	90.5	90.7	0.34	88.2	88.8	88.5	0.85	87.3	85.1	86.2	0.60
Vinson Quiet			0.100	91.9	90.3	91.1	0.44	91.4	90.2	90.8	0.66	81.9	82.7	82.3	0.65
Office		0.60	0.067	89.8	88.3	89.0	0.88	89.6	88.1	88.8	0.50	84.8	85.5	85.2	0.81
Auto			0.067	88.9	83.3	86.1	0.85	89.0	84.8	86.9	1.02	73.1	63.7	68.4	0.73
Humvee			0.067	60.6	65.4	63.0	0.95	65.2	73.3	69.3	1.33	21.7	41.7	31.7	2.26
M2 Bradley			0.067	60.7	66.9	63.8	1.14	74.3	78.4	76.4	0.94	34.2	42.5	38.4	1.27
Helicopter			0.067	61.0	66.6	63.8	0.94	75.6	78.9	77.2	0.78	39.4	55.8	47.6	1.24
F-15			0.067	73.0	75.5	74.3	0.79	74.7	78.6	76.6	1.11	70.5	69.4	69.9	0.88
E3A			0.067	84.6	85.6	85.0	0.62	88.2	89.9	89.0	0.94	66.7	65.3	66.0	1.09
P3C			0.067	85.7	82.7	84.2	1.19	89.5	86.0	87.7	0.72	80.9	78.5	79.7	1.00
MCE			0.067	90.5	87.8	89.1	0.96	90.8	90.0	90.4	0.75	77.5	78.7	78.1	1.10
BER		0.10	0.050	90.3	86.0	88.2	0.73	86.1	87.8	86.9	0.67	80.0	82.7	81.4	0.90
BLER			0.050	87.1	89.3	88.2	0.96	86.8	85.6	86.2	0.78	85.2	82.9	84.0	0.72
S_Tandem		0.10	0.050	84.8	83.7	84.3	0.61	89.0	87.3	88.2	1.03	75.8	75.5	75.6	1.10
D_Tandem			0.050	83.0	80.6	81.8	0.96	84.4	85.9	85.2	0.70	72.0	73.5	72.7	0.64
Intell. Perf				81.860	81.867	81.859	0.260	84.402	85.097	84.742	0.297	69.157	71.250	70.202	0.332
		1.00	1.000	Celp(M)	Celp(F)	Celp(C)	Celp(SE	KY(M)	KY(F)	KY(C)	KY(SE)	LPC(M)	LPC(F)	LPC(C)	LPC(SE)
Rank				6	5	6		1	1	1		7	7	7	

Figure 8: A wide, wrapped table giving the performance of various voice coding schemes.[10]

[10]From "A New Federal Standard Algorithm for 2400bps Coded Voice." Note the extra, inexplicable (in this context) box surrounding the performance and rank figures for the entry in the middle of the first part of the table. http://www.plh.af.mil/ddvpc/24results.htm.