IJDAR (2008) 11:111–126
DOI 10.1007/s10032-008-0065-5

ORIGINAL PAPER

Recognition and quality assessment of data charts in mixed-mode documents

Sudhindra Shukla · Ashok Samal

Received: 19 April 2007 / Revised: 27 March 2008 / Accepted: 23 April 2008 / Published online: 2 October 2008

Abstract Data charts can be used to effectively compress large amounts of complex information and can convey information in an efficient and succinct manner. It is now easier to create data charts by using a variety of automated software systems. These data charts are routinely inserted in text documents and are widely disseminated over many different media. This study addresses the problem of finding goodness of data charts in mixed-mode documents. The quality of the graphics can be used to assist the document development process as well as to serve as an additional criterion for search engines like Google and Yahoo. The quality measures are motivated by principles of visual learning and are based on research in educational psychology and cognitive theories and use attributes of both the graphic and its textual context. We have implemented the approach and evaluated its effectiveness using a set of documents compiled from the Web. Results of a human study shows that the proposed quality measures have a high correlation with the quality ratings of the users for each of the five classes of data charts studied in this research.

1 Introduction

Pictures or graphics help to stimulate human interest, since they are often easier to understand and are more interesting to the reader than text. Graphics and images also provide a more compact medium to convey information, as the saying "A picture is worth thousand words." A special class of graphics called data charts (e.g., pie charts, bar charts, line charts, etc.) are widely used now in academic, business, and scientific settings to support description and analysis of quantitative data. They help to locate and compare groups of quantitative data more easily, and to make generalizations about the data quickly. The use of graphics makes more vivid impact than a set of numbers or their description alone. According to Mayer [21], visual learning techniques that use images combined with written words is a powerful tool for enhancing cognition skills. Petre also notes that good graphics link perceptual cues to important information [26].

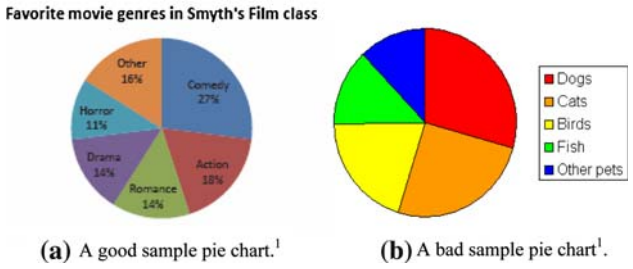
While data charts have been used over many centuries, their use has exploded with the introduction of computer-aided tools (e.g., Excel, SAS, SPSS, STATA, and Gnuplot) for creating them. This has many latent benefits and challenges. While it is now easier to develop data charts than ever before, it is now more difficult to maintain their quality. In the past, trained professionals who had a variety of skills in graphics design, art, statistics, and mathematics created data charts. This enforced a certain degree of quality control.

A good data chart conveys information in an efficient and, succinct manner, while a bad chart is confusing and places additional burden on the user for comprehension [35–37]. Figure 1a has a caption; its segments are labeled and the percentages are shown for each pie segment. Figure 1b, on the other hand, has no caption, and does not clearly show the percentage for each slice. Clearly, Fig. 1b does not convey the intended information as efficiently as Fig. 1a.

1.1 Motivation and applications

The goal of this research is to advance the state-of-the-art in visual learning by developing an automated system that will assess the goodness of data charts in mixed-mode documents. The goodness of a data chart is measured by its effectiveness in conveying the information to assist visual learning. This

S. Shukla · A. Samal (✉)
Department of Computer Science and Engineering,
University of Nebraska-Lincoln,
Lincoln, NE 68588-0115, USA
e-mail: samal@cse.unl.edu



(a) A good sample pie chart.[1] **(b)** A bad sample pie chart[1].

Fig. 1 Examples of good and bad pie charts: **a** A good sample pie chart; **b** A bad sample pie chart. Source: http://www.statcan.ca/english/edu/power/ch9/piecharts/pie.htm

can be used to help users assess the quality of the graphics they create and/or suggest deficiencies in data charts that already exist in mixed-mode documents.

This work can be used in many applications. It can be used in search engines to rank the retrieved documents that have graphics; a document with a higher quality of graphics is given a higher rank. A graphic evaluation tool can be used to develop better data charts in semiautomated authoring systems (e.g., Microsoft Office) that are used for generating reports and other types of mixed-mode documents. Statistical, mathematical, and drawing software that generate graphics can use the concept of graphic quality to create more effective data charts. While the techniques described in this study apply directly to data charts, the ideas can be extended to other types of graphics including pictograms, icons, animations, control charts, and flow charts.

1.2 Data charts

The main reasons for focusing on data charts are (1) data charts are widely used; (2) they are often generated by automated tools; (3) they are structured and have well-defined properties. After an extensive survey of data charts, we chose the five classes, which we found most common: line charts, column charts, bar charts, pie charts, and bubble charts. We estimated that these classes constitute over 75% of the charts used for displaying quantitative information today. Figure 2 shows an example of each class of charts. It should be noted that while charts can be an effective vehicle to communicate relational information, they are not as effective for simple information lookup [7].

1.3 Problem formulation

Our goal is to determine the quality of a graphic in a mixed-mode document. The problem can be formally stated as follows:

Given a mixed-mode document, $\mathcal{D} = [\mathcal{T}, \mathcal{G}]$, where $\mathcal{T}$ represents the textual component of the document $\mathcal{D}$ and $\mathcal{G}$ represents the graphics component and is represented as

$$\mathcal{G} = \{g_1, g_2 \ldots g_m\} \tag{1}$$

where g_i is a single isolated graphic in the document and m is the number of graphic components. Our goal here is to define a function QoG that is defined as

$$\text{QoG} : g_i \times \mathcal{T} \rightarrow [0, 1] \tag{2}$$

Thus, the QoG function maps each graphics component in the image (in the context of the associated text) to a goodness value. This function can be integrated to get a quality measure of the overall document, but is not addressed here. Since each graphic is evaluated independently, without loss of generality, we assume that each document contains only one graphic. In addition, we assume that document has no text wrapping, that its text lines are horizontal, and that the data charts have explicit or implicit rectangular frames (boundary lines).

1.4 Overview of the approach

Figure 3 describes the schematic of our approach. First, the document is split into (1) text and (2) graphic parts. The information extracted from each part is then combined to compute the quality metrics. The *textual processing* involves finding the chart references with their locations, text stemming, finding keywords in the adjacent paragraphs of the chart(s), and finding caption keywords [if caption(s) is/are outside the chart]. In *graphics-processing*, we derive the underlying components of the graphic component to both classify the graphic and derive its quality measure. The *text-matching* stage involves keyword-matching between the reference text keywords, label keywords, and caption keywords. Using this information, we compute a set of *quality measures* (QMs), which are integrated to yield an overall *quality of graphic* (QoG) function. This approach is applicable for any document in the digital form that is amenable to text-and graphics-processing. Any html page, pdf document, and a scanned (and OCR'ed) document can be processed using this approach.

1.5 Previous work

We briefly review the existing literature on graphics- and text-processing, two important intermediate stages in our QoG computation. Literature related to quality of graphics is discussed in Sect. 2, since it is quite vast and merits a more detailed analysis.

Graphics-processing. We use many low and intermediate level processing steps to extract the structure of the graphic. A full review of all the operations is beyond the scope of this paper. Interested reader may review some standard references in computer vision [13,29,32,38]. We focus on the more significant problem of line structure segmentation

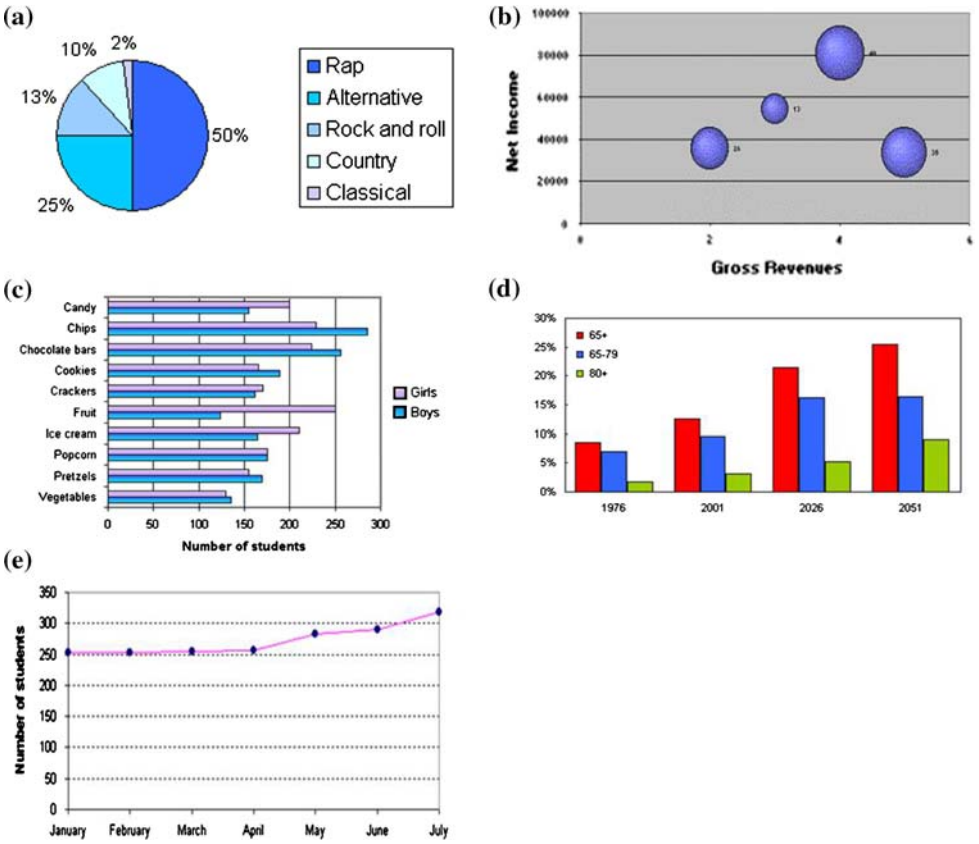


Fig. 2 Examples of different types of data charts: **a** A sample pie chart (source: http://www.statcan.ca/english/edu/power/ch9/piecharts/pie.htm); **b** A sample bubble chart (source: http://www.bris.ac.uk/is/selfhelp/documentation/exl97-r1/exl97-r1.htm); **c** A sample bar chart (source: http://www.statcan.ca/english/kits/issues/charts/chart2.htm); **d** A sample column chart (source: http://www.statcan.ca/english/kits/issues/charts/chart2.htm); **e** A sample line chart (source: http://www.statcan.ca/english/edu/power/ch9/linegraph/line.htm)

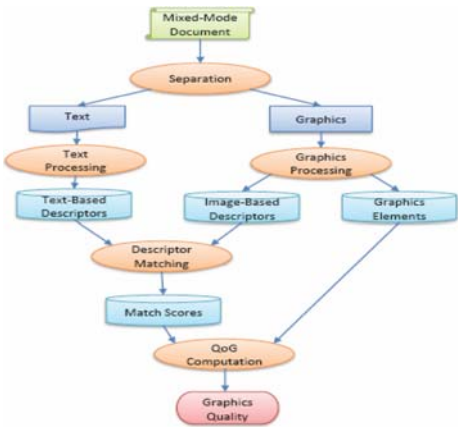


Fig. 3 Schematic of our approach for QoG computation

here, since it is more important in extraction of structure in data charts. A common approach to extract line structure is through vectorization, i.e., converting the image to raw fragment vectors. These smaller vectors are then grouped into longer lines and polylines. Three approaches in vectorization are common: thinning [23,31,34], border following [20,30], and direction distance propagation [20,30]. The problem with the first two methods in our application is that thinning can cause shape distortion at line junctions and figure intersections, while border following methods will mix line vectors and text vectors at figure intersections. Directional distance propagation is proposed to overcome these drawbacks, but this method requires many phases of postprocessing [14]. Myers et al. [22] proposed a verification-based data extraction approach in which linear features are extracted by (1) identifying linear-segment pixels, (2) linking detected line pixels to form a list of linear segments, and (3) applying semantic criteria to verify linear features. We follow this approach to detect and verify the features in data charts. All these methods are based on the same idea: extract line-feature and recover missing part with sufficient line information. Zhou and Tan [41] have proposed a Hough-based approach to recognize bar charts. They also proposed a HMM-based approach to recognize scientific charts [42]. Use of constraint grammars for syntactic and semantic analysis of diagrams have also been proposed in literature [6,10–12].

Text-Processing: There are many steps in text-processing and the review of all of them is beyond the scope of this study. Here, we focus on text-searching algorithms, since they are the most important step in our work. These algorithms try to find a place where one or several strings (also called patterns)

are found within a larger string or text. Over the years many algorithms for text-matching have been proposed including the *brute force* algorithm [3], and *Knuth–Morris–Pratt* algorithm [16]. *Boyer–Moore* algorithm [4] examines the target string that is being searched for, but not the string that is being searched. Empirical results show that the variations of Boyer and Moore's algorithm designed by Crochemore and Rytter are the most efficient in practice [8].

2 Visual learning with data charts

Visual learning techniques are graphical ways of working with ideas and presenting information. The effectiveness of information transmission from the graphic to the human is crucial in deciding the quality of the graphic. The quality of data charts is motivated by cognitive theories (Sect. 2.1), concepts of structural elements (Sect. 2.2), and the principles of graphics integrity. They are briefly described in this section. While the section is focused on data charts, most of the discussion is also applicable for graphics in general.

2.1 Cognitive theories

Human understanding of pictures is a vast research area without a universally accepted theory. The process is divided into multiple levels and encompasses actions from the first glance at a picture to detailed ideas about it. Arnheim [1] contends that picture perception is the viewer's response to the basic forms present in the picture and the Gestalt laws of organization are the primary conveyors of meaning. Cognitivism, one of most dominant paradigms still researched today, defines cognition with symbolic representations [33]. In the case of data charts, they correspond to keywords and structural elements. Mayer [21] carried out a comprehensive study to determine the characteristics of effective multimedia presentations. The verbal form (words) typically denotes speech and printed text. The pictorial form (pictures) can be static graphics (illustrations and photos) or dynamic graphics (animation and video). The focus of the study was on coordinating words and pictures to maximize learning. After rigorous testing and experimental results, Mayer [21] proposed a set of principles for an effective multimedia presentation. We summarize below the principles that apply directly to data charts:

- *Spatial contiguity*: Learning is more effective when a chart is presented near its associate verbal form and vice versa.
- *Coherence*: Learning is more effective when extraneous material (e.g., unnecessary labels and unrelated figure elements) is excluded in the presentation.
- *Redundancy*: It is best to present information in multiple formats to learners.
- *Individual differences*: The design of the document has a bigger impact on low-knowledge and high spatial learners than on high-knowledge and low-spatial learners.

Other significant research results are reported in [17,28]. Lewandowsky and Spense review several empirical studies that examined the suitability of graphs and the role of psychophysics in their perception [17]. Bertin proposed taxonomy of graphics elements and their perceptual properties and a grammar do describe them [2].

2.2 Structural elements of data charts

A chart can be regarded as an alternative representation of data already stored in a text format. Thus, it is important to reinforce the connection between text and pictures for enhanced cognitive processing. Data charts help in this regard, as they reveal patterns, relationships, and interdependencies.

The close interaction between the picture elements and their descriptions derived by human learners has been established by many researchers. Strothotte and Strothotte [33] state that, learning from pictures is achieved by first forming a mental model of the picture and then carrying out a mental dialogue with it. Furthermore, more engaged a user is in the process of interpretation, the deeper is the understanding. Steps involved in the cognitive processing of a data chart include the following (1) naming and categorizing the picture (2) identification of basic graphic elements (e.g., circles, rectangles, or lines), (3) naming the basic graphic elements, (4) recognizing and naming the structured elements, and (5) deriving the semantic information. We have used criteria similar to this approach to break down a data chart into structural components and then recognize it by examining their organization.

2.3 Graphical integrity of data charts

Graphical integrity refers to effective transmission of information via graphics. Pioneering work in this area was carried out by Tufte, whose focus was on printed text and corresponding figures (graphics) in data analysis [35–37]; it is especially important in the context of learning, since false or inaccurate learning of the concept may be difficult to reverse. Tufte [35–37] listed many criteria of excellence in the graphical display of quantitative information. The relevant criteria for data charts are listed below:

- The graphic should induce the reader to focus on the central concept and not on the methodology, or graphic design or tools used to derive them.
- The graphic should encourage the comparison of data, serve a clear purpose, and be coherent. It should also be closely integrated with its description.

- The graphic should show data at different levels of granularity or resolution.
- The graphic should show the data and present many pieces of information clearly.
- The graphic should avoid distortion of data and overload of information.

3 Quality measures for graphics

In this section, we define a set of measures to evaluate the quality of data charts. We first summarize the framework used to design the quality measures (QMs). The measures are motivated by research in educational psychology and cognition (Sect. 2) and use structural components of the data charts as the basic units of analysis.

3.1 Framework for quality measures

In Sect. 2, we described important principles from psychology that are critical in the cognition of data charts. Here, we describe the principles, based on which we define the quality of data charts. Before we list them, we briefly define some central terms used in our research and referred to in the rest of this article:

1. *Caption*: A title, short explanation, or description accompanying an illustration or a graphic.
2. *Labels*: Text that is used to identify something within a graphic.
3. *Reference text*: Text in paragraphs adjacent to the graphic (i.e., above and/or below the graphic).
4. *Caption keywords*: A significant or descriptive word present in the caption.
5. *Label keywords*: A significant or descriptive word present in the labels.
6. *Reference text keywords*: A significant or descriptive word present in the reference text.
7. *Structural elements*: These are elements of a graphic that are the building blocks of a graphic. Examples include x-axis, y-axis, circles, horizontal, and vertical bars.
8. *Necessary elements*: These are the structural elements that are central to the definition of a graphic type and must be present in all instances of the graphic type. For example, horizontal bars must be present in bar chart and segments must be present in a pie chart.
9. *Optional elements*: These are structural elements that are used to provide additional information for a graphic, which is not needed in all instances of the graphic type. For example, a legend box may or may not be present in a data chart.

Many observations and principles in the literature (Sect. 2) are qualitative or imprecise and hence cannot be directly quantified. Examples include "encourage the comparison of data" or "serve the purpose and are coherent." We have designed quantitative measures to approximate the qualitative principles proposed in the literature. Identification of structural elements allows for comparison between them and improves coherency. For example, identifying the bars in a bar chart, segments in a pie chart, and circles in a bubble chart fosters a comparative analysis between them. Using these principles, we list the key principles to compute the quality of a graphic:

1. *Ability to divide the graphic into known structural components*: This refers to the ease of identifying the basic elements in a graphic without difficulty (Sects. 2.1, 2.3). A graphic in which it is difficult to extract the structural elements is harder to understand and hence is of lower quality and vice versa. The assumption here is that logical decomposition process employed by humans can be the basis for automated methods.
2. *Graphics modality in terms of structural components and labels*: This is the ability to name (or label) the basic graphic elements that have been identified. This helps users to interact textually with the graphic and refer to the names of basic graphic elements (Sects. 2.1, 2.3), which improve its understanding.
3. *Distance between the graphic and its references*: This is the space in the document, which separates the graphic and its references, especially the first. Spatial contiguity principle (Sect. 2.2) suggests that this should be as small as possible.
4. *Text modality in the whole document*: This is the multiplicity of the relevant texts in the document with respect to the graphic. The three textual parts of a document, i.e., labels in the graphic, figure caption (in or outside the graphic), and reference text (outside the graphic), yield three sets of keywords. In accordance with Mayer's redundancy principle and individual differences principle (Sect. 2.2), greater redundancy between them would foster easier understanding.
5. *Contrast of the graphic*: This refers to the clarity of the graphic. A graphic with good contrast provides a better medium for learning than one that has poor contrast (Sect. 2.3).

3.2 Structural elements of data charts

In this section, we describe the graphic (structural) elements present in data charts. The goal is to systematically analyze their structure. This analysis is analogous to the implicit recognition and classification process employed by humans described in Sect. 2.1. The recognition is achieved by

Table 1 Necessary and optional elements of data charts

Data chart	Necessary elements	Optional elements
Pie chart	Circle (complete), concurrent line segments	Legend box, figure labels, figure caption, hashed segments, segment position markers, ticks on perimeter of circle
Bubble chart	X-axis, Y-axis, Origin, two or more circles	Legend box, X- and Y-axis labels, figure labels, figure caption, hashing (inside circles), axis position markers
Bar chart	X-axis, Y-axis, origin, horizontal bars	Legend box, X- and Y-axis labels, figure labels, figure caption, hashing (inside bars), axis position markers
Column chart	X-axis, Y-axis, origin, vertical bars	Legend box, X- and Y-axis labels, figure labels, figure caption, hashing (inside bars), axis position markers
Line chart	X-axis, Y-axis, origin, skewed lines	Legend box, X- and Y-axis labels, figure labels, figure caption, axis position markers (blobs)

recognizing the structural elements in the data charts, classifying them, and finally forming a mental model of them. The subsequent cognition process then relies on the mental model to integrate and assimilate facts about the data charts.

We divide the elements of a data chart into two classes: *necessary* elements and *optional* elements (Sect. 3.1). The necessary elements correspond to the core structural elements used by humans to understand and classify a data chart. The optional elements may or may not further reinforce this process depending on the complexity of the context. The necessary and optional elements for the five classes of data charts used in this research (Sect. 1.1) are summarized in Table 1.

3.3 Quality measures

In this section, we define the quality measures that can be used to evaluate the quality of a data chart. It is important to define the measures such that their values are on a uniform scale and are normalized. This facilitates the comparison between different data charts. In the following, all the QMs are defined so that their values are normalized (between 0 and 1).

- Spatial location quality: This is a measure of the distance between the first figure reference and the location of data chart, measured in number of characters. Given a graphic G in a document D, Spatial location quality (SLQ) is defined as

$$\mathrm{SLQ}(G, D) = 1 - \frac{|l_\mathrm{r}(D) - l_\mathrm{f}(D)|}{\mathrm{size}(D)} \tag{3}$$

  where $l_\mathrm{r}(D)$ is the location of the first figure reference, $l_\mathrm{f}(D)$ is the location of the figure, and size(D) is the size of the document D. The locations and the size are defined in number of characters. This measure is based on the spatial contiguity principle (Sect. 2.2). To make the analysis independent of the font shape and size and formatting of the documents (e.g., scanned document vs. a web page), we use the number of characters. It is equal to 1 when the figure reference is next to the figure, i.e., when $l_\mathrm{r}(D) = l_\mathrm{f}(D)$. The lowest value for this measure is 0 and occurs when the figure reference is separated by a whole document page from the figure. In this case, $l_\mathrm{f}(D) - l_\mathrm{r}(D)$ is equal to size(D).
- Label completeness: Ideally, all structural elements of a data chart should be labeled. We compute label completeness (LC) as the fraction of the elements that are actually labeled in the graphic. This is motivated by Strothotte's guidelines and Tufte's principles of graphical integrity (Sect. 2). Given a graphic G in a document D, it is defined as

$$\mathrm{LC}(G, D) = \frac{n_\mathrm{l}(G, D)}{n_\mathrm{SE}(G, D)} \tag{4}$$

  where $n_\mathrm{l}(G, D)$ is the number of label keywords and $n_\mathrm{SE}(G, D)$ is the number of separate necessary structural elements for graphic G in document D. Since, it is not always feasible to exactly recognize all the necessary structural elements, this is difficult to get the exact value every time. It is an approximation, which we have verified by extensive testing. The value is normalized between 0 and 1.
- *Graphic contrast*: This is motivated by Tufte's principles of graphical integrity (Sect. 2). Graphic contrast (GC) is defined as the block distance between the histogram-equalized grayscale image and its original grayscale

image. Let the histogram-equalized image be G_{heq}. Let H_G be the histogram for the graphic G and $H_{G_{\text{heq}}}$ be the histogram for the histogram-equalized graphic, G_{heq}. Assuming the number of bins in the histogram to be n, we compute GC as follows:

$$\text{GC}(G, D) = \sum_{i=0}^{n-1} H_G[i] - H_{G_{\text{heq}}}[i] \tag{5}$$

The value is normalized between 0 and 1.The ideal case is when the corresponding grayscale image has an equalized histogram, i.e., GC is 1. The worst case is when the image is completely dark, i.e., GC is 0.

- *Modality measures:* We define three different types of modality measures to determine the consistency between the caption keywords, labels, and the reference text keywords. As described in Sect. 2, these are motivated by Mayer's individual differences principle, Strothotte's guidelines, and Tufte's principles of graphical integrity.

Reference text and caption consistency: Reference text and caption consistency (RCC) measures the degree of match between the reference text keywords and caption text keywords. It is motivated by Mayer's individual differences principle, Strothotte's guidelines, and Tufte's principles for graphical integrity (Sect. 2). It can be computed as the percentage match between the number of reference text keywords and caption keywords. Ideally, it should be 1. The value is normalized between 0 and 1. Given a graphic G in a document D, it is defined as

$$\text{RCC}(G, D) = \frac{n_{\text{c}}(G, D)}{n_{\text{r}}(G, D)} \tag{6}$$

where $n_{\text{c}}(G, D)$ is the number of caption keywords and $n_{\text{r}}(G, D)$ is the number of reference text keywords for graphic G in document D.

Reference text and label consistency: Reference text and label consistency (RLC) is the degree of match between the reference text keywords and caption text keywords and is defined as follows:

$$\text{RLC}(G, D) = \frac{n_{\text{l}}(G, D)}{n_{\text{r}}(G, D)} \tag{7}$$

where $n_{\text{l}}(G, D)$ is the number of label keywords and $n_{\text{r}}(G, D)$ is the number of reference text keywords for graphic G in document D. The value is normalized between 0 and 1.

Caption and label consistency: Caption and label consistency (CLC) is the degree of match between the caption keywords and label keywords. Given a graphic G in a document D, it is defined as

$$\text{CLC}(G, D) = \frac{n_{\text{c}}(G, D)}{n_{\text{l}}(G, D)} \tag{8}$$

where $n_{\text{c}}(G, D)$ is the number of caption keywords and $n_{\text{l}}(G, D)$ is the number of label keywords for graphic G in document D. The value is normalized between 0 and 1.

4 Text-processing

In this section, we describe the processing of the main-text body of the document. The main-text body consists of the document without the data chart(s), i.e., a complete document page with blank space inserted at the location of the data chart. If the figure caption is outside the data chart (e.g., below the figure), it is considered to be a part of the main-text body. The main purpose of the text-processing is to locate the text that is related to the graphic and extract the important information from it. The end result is a list of reference text keywords. We assume that each graphic has a reference. However, the absence of any reference to the graphic is easily detected in our approach. In such a scenario, we assign a predefined score to the document and short circuit the quality computation.

4.1 Text-parsing

The objective of this step is to understand the structure of the main-text body. We segment the text into words, text lines, and structural blocks (paragraphs, which consist of a group of text lines) using a structural layout analysis. This is carried out in a top–down fashion, where the page is split into blocks of text (paragraphs) and each paragraph is split into lines. The steps involved in text-parsing are as follows:

1. *Compute the total number of lines in a page*: The number of lines in a mixed-mode document is not well defined, since it contains both text and graphics. However, the size of the graphic can be approximated by using the relative size of the graphic to the size of the document.
2. *Form a character matrix*: All characters are assumed to be of the same size and blank spaces between words are considered as invisible characters. At the end of this step, we have a character matrix called the textmap of size $m \times n$, where m is the number of lines in the document and n is the number of characters in a line.
3. *Compute the distance between words*: A word is a contiguous group of charters that is terminated by a null (blank) character. The distance between two words w_i and w_j is defined as the number of characters separating the two words.
4. *Detect and mark figure references*: The locations in the main-text body where the figure is formally referred are identified and stored as figure references. This can be done easily by searching for keywords like "figure," "diagram," "chart," and labels or numbers. The respective

positions are marked in the *textmap* array (described in Step 2).

4.2 Reference text-stemming

Text stemming is the process of reducing a word to its stem or root form, so that variations of particular words such as past tense, plural, and singular usage are recognized with the same *stem* form. The objective of this step is to determine the stemmed forms of the words in the reference text. The stemmed words are used for keyword-matching later on. This is carried out with the caption keywords. We used the Porter stemming algorithm for our research. This is a popular algorithm that is widely used and has been adapted over many years [27]. It is defined as "a process for removing the commoner morphological and inflexional endings from words in English." We used an ANSI C version of the porter stemming algorithm (http://www.tartarus.org/~martin/PorterStemmer).

4.3 Reference text keywords compilation

The objective in this step is to accumulate the keywords in the reference text. This can be done by parsing the entire stemmed paragraph (Sect. 4.2) and recursively removing high frequency words like "and," "the," and "is." The remaining words are then marked as keywords. This is done in accordance with Zipf's law [39]. The words exceeding a predefined frequency are considered to be common and are removed from consideration. This process is also applied to the figure caption. As with stemming, any standard software can be used to find stop words.

5 Graphics-processing

In this section, we describe the process of extracting high level information from the graphics image (data chart). First, we perform several image-processing operations to prepare the image for extraction of structural elements. Since lines form an important class of structuring or substructuring elements, we describe how they can be extracted efficiently. After identifying relevant structural elements, we verify their presence, and use them for classification.

5.1 Initial processing

Initial processing is commonly performed in image analysis tasks to simplify the complexity and effectiveness of downstream analysis. In the case of data charts, the goal of initial processing steps is to go from pixels to linear structures. The initial processing steps includes binarization, noise reduction, thinning, and edge detection.

Image binarization: Data charts typically have significant amount of white space (background). Using a threshold operator, we can derive a quick and simple segmentation of the image into background (white space) and foreground (text and linear structures). An adaptive thresholding scheme proved effective for this task [25].

Noise reduction: Image noise occurs mainly from image transmission across different media such as the conversion from physical samples to scanned images. Salt and pepper noise is the most common form of noise, which we encountered in data chart images. Standard filtering operations were effective in removing this type of noise.

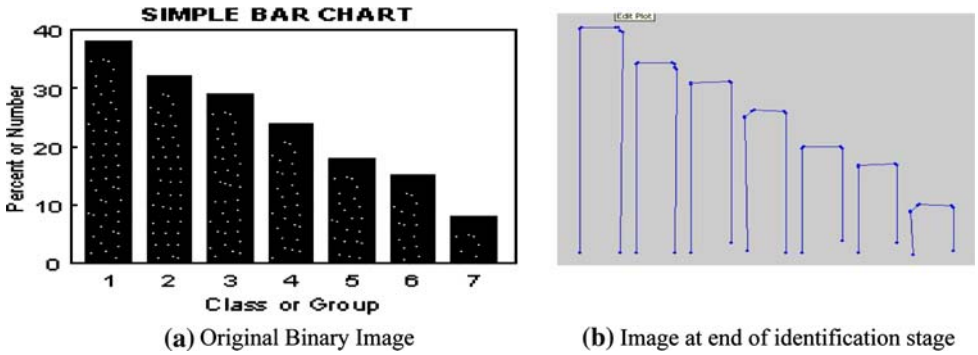
Thinning: It is an image-processing operation in which the regions of a binary image are reduced to lines that approximately fit center lines or skeletons. This is especially effective in images where main structures are elongated. In the case of data charts where many central elements are linear, this is an important step in information compaction. Thinning significantly improved the subsequent recognition process.

Edge detection: Edges characterize boundaries that are areas with strong intensity contrasts—a jump in intensity from one pixel to the next. Many different edge detection methods are reported in literature. In our experimentation with different operators using our images, the Canny edge detector [5] was found to be the most effective and hence is used in our work.

5.2 Line-processing

Since lines form an important class of structures in our images, we extract the linear structures. Our line detection algorithm starts with edge images and is a generalization of vectorization [9,15], which yields straight lines and arcs. It broadly consists of two phases: (1) primitive line detection is carried out to obtain line fragments from the edge image pixels and (2) a postprocessing step in which the line fragments are joined, manipulated, and fitted to refine the output. Overall, we formulated the line detection algorithm to yield fine lines that are as accurate as possible. The different operational steps in our line detection process are based on [18,19] and are described next:

1. *Edge-linking*: In this first step, we form lists of connected edge points from a binary edge image. The edge points are linked together into chains.
2. *Fitting line segments*: We use an array to store the edge points from a binary edge image. The edge points are linked together into chains based on the size and position of the maximum deviation from the line that joins the endpoint. If the maximum deviation exceeds the allowable tolerance, the edge is shortened to the point of maximum deviation and the test is repeated. Details of this algorithm are given in [18,19].



(a) Original Binary Image **(b)** Image at end of identification stage

Fig. 4 Original binary image and image obtained after identification: **a** Original binary image; **b** Image at end of identification stage

3. *Merging colinear segments*: In this step, we merge co-linear line segments that may have been separated in the edge-linking process. The list of line segments are scanned to merge segments together. Segments are merged if the orientation difference is less than a tolerance angle and if the ends of the segments are within a tolerance distance of each other.

5.3 Text extraction

To extract the text labels in the data charts, we used optical character recognition (OCR). In addition to the labels, sometimes the figure caption is also embedded in the data chart and has to be extracted as well. There is a large body of literature on OCR [24,40] and a number commercial OCR software exists. For our purpose, we used standard OCR software to derive a text output (labels and/or captions) of the scanned data chart to a file along with the respective locations of the text. Then label-stemming is carried in the same way as the reference text-stemming described in Sect. 4.2. The label keywords are then compiled in the same manner as reference text keywords (Sect. 4.3).

5.4 Extraction of structural elements

Using the lower level features obtained so far, we derive the structural elements of the data charts. The set includes circular segments, horizontal and vertical bars, axes, legends. These correspond to the structural elements identified in Sect. 3.2. The process is complicated by the fact that these features touch and overlap and by inherent inaccuracy of the lower level operations. We use a two-step approach for this purpose: (1) we group linear structures to form structural elements and (2) use the semantic properties of the structural elements to verify their presence.

5.4.1 Formation of structural elements

The operational steps involved in forming an initial list of structural elements are as follows:

1. Remove outer bounding boxes in the figure: Sometimes figures have a rectangular box to demarcate it from other figures and from text. This is done by removing the longest straight lines from the left and right sides and the top and bottom of the picture.
2. Extract the x-axis and y-axis if present: The x-axis is the longest horizontal line present at the bottom of the figure and the y-axis is the longest vertical line in the figure. They intersect at the origin. This is a tedious step in which all the lines in the figure are analyzed for these properties recursively to eventually derive the axes.
3. Remove elements below the x-axis and to the left of the y-axis: This may involve labels and/or other elements such as axis markers, etc. The graphical structural elements are all located in the first quadrant with respect to the axes. This step is applicable for the data charts that have axes.
4. Remove elements outside the rectangular region formed by the x-axis and y-axis together. Again, this step is applicable for the data charts that have axes.

These results of using the four steps are shown in Fig. 4. Figure 4a shows the original binary image and Fig. 4b is the image obtained after the four operations.

5.4.2 Verification of structural elements

We use semantic rules to verify the presence of shapes formed by the structural elements found in data charts. The semantic rules used in our work are described below.

Circle: A number of methods have been proposed in the literature to compute the roundness of a shape. A simple roundness metric is given by

$$\text{Roundness} = \frac{4\pi A}{P^2} \tag{9}$$

where P and A denote the perimeter and area of the shape, respectively. This metric is equal to one only for a circle and it is less than one for any other shape. Given a shape S, we verify it to be a circle if its roundness is greater than 0.90.

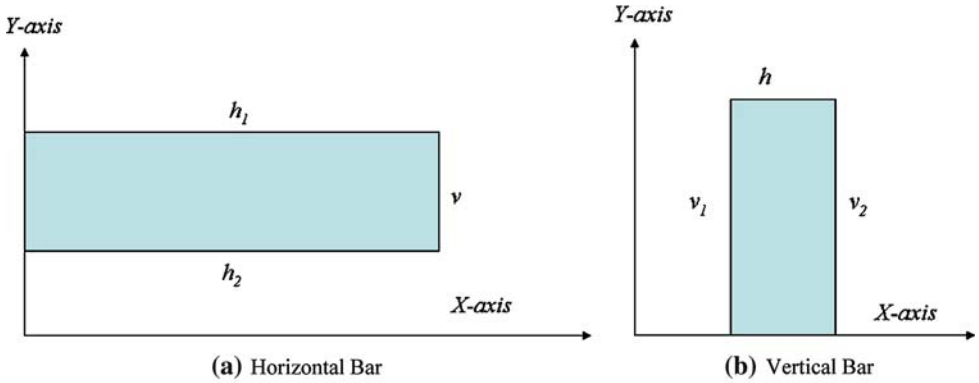


Fig. 5 Structure of horizontal and vertical bars: **a** Horizontal bar; **b** Vertical bar

Thus,

$$\text{Roundness}(S) \geq 0.9 \Rightarrow \text{Circle}(S)$$

Horizontal bar: Horizontal bars are characterized by pairs of horizontal lines that are of equal length, equidistant from each other, and emanating from the y-axis. They are terminated at both ends by a vertical line that is parallel to the y-axis (see Fig. 5a). A set of three line segments h_1, h_2, and v form a horizontal bar if and only if

$$h_1 || h_2 \wedge \text{length}(h_1) = \text{length}(h_2) \wedge h_1.\text{left} = y\text{-axis} \wedge h_2.\text{left} = y\text{-axis} \wedge h_1.\text{right} = v.\text{top} \wedge h_2.\text{right} = v.\text{bottom} \wedge v || y\text{-axis}$$

Vertical bar: Vertical bars are characterized by pairs of vertical lines that are of equal length, equidistant from each other, and emanating from the x-axis. They are terminated at both ends by a horizontal line h that is parallel to the x-axis (see Fig. 5b). A set of three line segments v_1, v_2, and h form a vertical bar if and only if

$$v_1 || v_2 \wedge \text{length}(v_1) = \text{length}(v_2) \wedge v_1.\text{bottom} = x\text{-axis} \wedge v_2.\text{bottom} = x\text{-axis} \wedge v_1.\text{top} = h.\text{left} \wedge v_2.\text{top} = h.\text{right} \wedge h || x\text{-axis}$$

Skewed line: Lines in the grid area that are not parts of horizontal bars, vertical bars, circles or any closed polygon are considered to be skewed lines. Skewed lines are characterized by the absence of symmetry in the relevant grid area formed by x- and y-axis, i.e., they are not equidistant from each other; they do enclose any known shape and may or may not touch the axes.

5.5 Classification of data charts

Classification of the graphic is an important step in the understanding of the data chart. Different classes of the chart will induce different mental models and analyses (Sect. 2.2). Our classification scheme is motivated by the literature in cognitive theories (Sect. 2) that strongly suggest that structuring elements play a crucial role in the understanding of data charts. To enact this classification scheme, we use a decision tree based on these structural elements to classify the graphic being examined into one of the five classes of charts described in Sect. 1.2. An *unknown* category is included to allow for data charts that cannot be classified. Structural elements that have been identified in Sect. 3.2, and extracted using techniques described in Sect. 5.4, play a central role in the classification process.

A set of simple rules to identify each class of data chart based on its necessary features is given below. The rules are graphically shown in the form of a decision tree in Fig. 6.

- X-axis $\wedge$ Y-axis $\wedge$ Origin $\wedge$ Horizontal bars $\rightarrow$ Bar chart
- X-axis $\wedge$ Y-axis $\wedge$ Origin $\wedge$ Vertical bars $\rightarrow$ Column chart
- X-axis $\wedge$ Y-axis $\wedge$ Origin $\wedge$ Skewed Lines $\rightarrow$ Line chart
- X-axis $\wedge$ Y-axis $\wedge$ Origin $\wedge$ Circles $\rightarrow$ Bubble chart
- Circle + Line Segments (inside circle that intersect at center) $\rightarrow$ Pie chart

Figure 6 shows that the high-level features are used for classification, and it is thus critical that they be identified accurately. The process of extracting these features (e.g., axes, bars, columns, etc.) is complex, because we need not only to extract them from complex images, but also to verify them using contextual information.

6 Quality of graphic computation

In this section, we discuss how text and graphics-base descriptors for the graphic are integrated to evaluate its quality. First, the text-matching of the keywords (Sect. 3.1) is discussed. We then describe how the different quality measures are computed. Finally, we present how these measures are combined to compute an overall quality of the graphic.

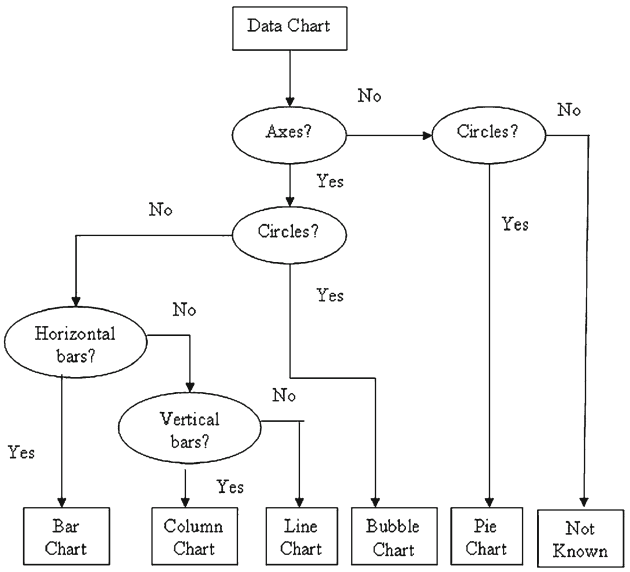


Fig. 6 Decision tree for data chart classification

6.1 Text-matching

We described how the keywords are extracted from the text in Sect. 4.2. With the keywords established, we need to detect and check for matches within the three text modes described in Sect. 3.1: (1) label keywords and caption keywords, (2) label keywords and reference text keywords, and (3) label caption keywords and reference text keywords. We use a version of the *Knuth–Morris–Pratt* algorithm [16], which searches for occurrences of a pattern (keyword) by employing the simple observation that when a mismatch occurs, we have enough knowledge simply by possessing the pattern to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

6.2 Quality measures computation

From the text-processing (Sect. 4), graphics-processing (Sect. 5), and text-matching stages, we obtain all the descriptors needed for computation of the quality measures defined in Sect. 3.3. The text-matching stage checks the modality of keywords obtained from the text- and graphic-processing stages, respectively. The text-based descriptors include the following:

- Location of data chart on document page
- Location of figure references on document page
- Character count of document page
- Number of characters in a line, i.e., line character count
- Reference text keywords
- Caption keywords

The image-based descriptors include the following:

- Location and coordinates of lines
- Location and coordinates of structural elements
- Location of labels
- Label keywords
- Contrast block distance of equalized histogram

6.3 Quality of graphic computation

The quality of a graphic (QoG) is an integrated quantitative measure of the overall quality of a data chart in a mixed-mode document. The formulation here is designed for data charts, but it can be suitably extended to other types of graphics as well. In general, QoG is a function of all the quality measures (QMs) defined in Sect. 3.3, i.e.,

$$\mathrm{QoG} = f(\mathrm{SLQ},\ \mathrm{LC},\ \mathrm{GC},\ \mathrm{RLC},\ \mathrm{CLC},\ \mathrm{RCC}) \tag{10}$$

In general, the function f can take any form. Arithmetic mean and weighted mean are two common functions that can be used for this. If normalized quality measures are used (i.e., the range of values for QMs is between 0 and 1), a QoG value of 1 would indicate a perfect data chart and vice versa.

If a weighted mean approach is used, the weights can be derived according to the graphic type being evaluated or the features being checked for. For example, if the emphasis is on checking the labels in a graphic, the weight for LC can be set to be relatively high. If the emphasis is on checking the modality, we can increase the weight for the text modality measures. This flexibility is an important feature, which allows for applying the QoG measurement to a variety of graphic types and emphases on quality aspects.

In our research, we have considered all the quality measures to be of equal importance and have thus used a simple mean to compute the QoG. While this formulation is by no means binding, we use this formulation as a base model. This can be expressed as

$$\mathrm{QoG} = \left(\frac{\mathrm{SLQ} + \mathrm{LC} + \mathrm{GC} + \mathrm{RLC} + \mathrm{CLC} + \mathrm{RCC}}{6}\right) \tag{11}$$

7 Implementation and results

In this section, we discuss the implementation details and the results of using the approach on a test set. The system was implemented on MATLAB (http://www.mathworks.com). In the training stage, we tuned our algorithms on a set of data charts. The system was then tested with an independent set of data charts that formed our test set.

7.1 Implementation details

The MATLAB package was chosen as the base platform. It integrates mathematical computing, visualization, and is a powerful technical language. Customized algorithms were written in "C" and were integrated with the MATLAB engine. The MATLAB image-processing tool box provides support for color space conversion, image analysis, image files I/O, image registration, image transforms, filter design, and image enhancement. The API support provided in the MATLAB package enables user to link the application with other platforms also. All the text-processing algorithms were implemented in the C++ programming language with the assumption that the alphabet is the set of ASCII codes or any subset of it.

7.2 Training phase

Our training data set consisted of 30 mixed-mode documents with six documents representing each of the five classes of data charts. In this step, the goal was to debug the system and tune the parameters involved. Two issues were considered in choosing the documents. We chose documents that had English text in only horizontal and vertical orientations. This was done primarily to accommodate the requirements of the OCR systems that recognize only letters of the English alphabet and whose orientations are either vertical or horizontal. This merely simplifies the text search and matching processes and does not affect the generality of our approach. It also simplifies the process of specifying the location of figure labels within a figure. Second, since our emphasis is on the evaluation of the quality of a data chart in a mixed-mode document, we chose documents that did not contain special icons or special decorative fonts embedded within a body of text. This again does not affect the generality of our approach, but simplifies the text-processing stages, which would otherwise be very computationally intensive.

A number of parameters had to be tuned during the training stage. The document character count, line character count, distance measure between words, and text-matching had to be adjusted to account for blank spaces and different font types. In all cases, we manually checked the results to ascertain the accuracy of the results obtained.

In training, we found several problems and we briefly list them below:

- *OCR errors*: Some characters are wrongly recognized by the OCR system. This affects the label keywords extraction process and consequently the overall keyword-matching process. Leaving the errors uncorrected will only minimally affect the overall quality of graphics, since many factors are used to compute the overall quality.
- *Vectorization limitations*: In some data charts, no good line vectors can be derived during vectorization. Sometimes, fine lines get merged together in the process and the subsequent thinning loses the originally intended appearance and the resulting line vectors do not provide accurate vector information. As a consequence, the structural elements are not extracted accurately. An example is shown in Fig. 7a.

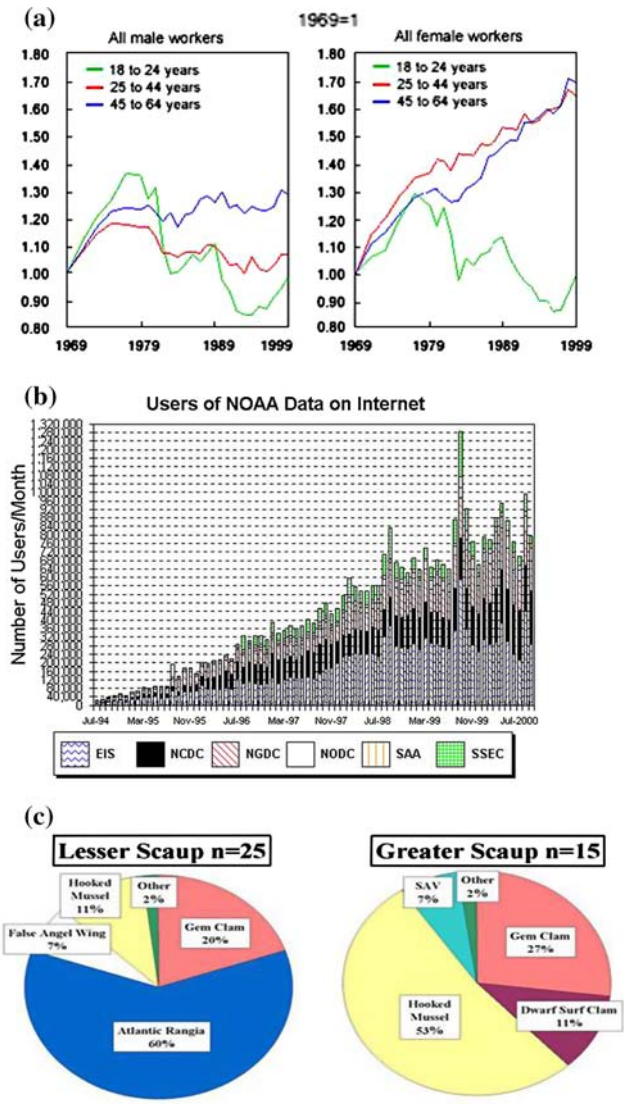


Fig. 7 Sample problematic data charts: **a** Line Chart with vectorization problems (source: Statistics Canada Internet site, Extracted October 06, 2005, from http://www.statcan.ca/english/kits/issues/charts/chart8.htm); **b** Uneven grid lines in a sample column chart (National Oceanic & Atmospheric Administration (NOAA), U.S. Dept of Commerce Extracted October 06, 2005, from http://www.esdim.noaa.gov/charts/noaadatause1.html); **c** Pie charts with interfering labels (source: Diving Duck Distribution, Abundance, and Food Habits in Chesapeake Bay, Perry M., Osenton P.C., Lohnes E.J.H., USGS Patuxent Wildlife Research Center, Extracted October 06, 2005, from http://www.pwrc.usgs.gov/resshow/perry/foodhabits.htm)

- *Complex charts*: Sometimes, the structure of the charts is too complex and does not lend itself to the description used in this research. In Fig. 7b, the vertical bars all merge into each other and this makes it difficult to extract the bars effectively. This problem, which is uncommon, can be addressed using a more sophisticated grid line removal approach.
- *Interfering labels*: The verification and classification process is also affected in some pie charts where label boxes affect the line extraction and shape recognition processes (see Fig. 7c). Here, the label boxes interfere with the circle detection process because of their location on the boundary of the circle. We avoided this problem by removing such charts from the dataset, since it would require major changes to our circle-finding algorithm. Morphological operations can be used to fill up gaps in contours.

7.3 Performance evaluation

7.3.1 Test set

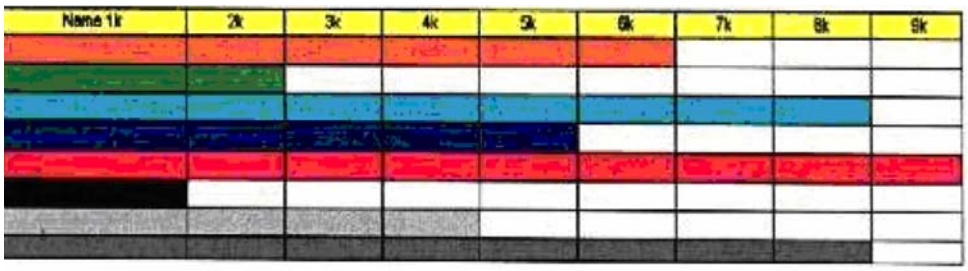
Our test set consisted of 75 mixed-mode documents with 15 documents for each one of the five classes of data charts. These documents were compiled from a variety of sources all available on the World Wide Web (WWW).

7.3.2 Classification

Table 2 summarizes the results of our classification process in a confusion matrix format. The results show that all charts were classified as their true class type or as unknown. All 15 bubble charts were classified correctly. Overall, the system

Table 2 Confusion matrix of classification of data charts

	Pie	Line	Bubble	Bar	Column	Unknown
Pie	13	–	–	–	–	2
Line	–	13	–	–	–	2
Bubble	–	–	15	–	–	0
Bar	–	–	–	12	–	3
Column	–	–	–	–	14	1



(a) A sample bar chart classified as unknown

(b) A sample line chart classified as unknown

Fig. 8 Sample misclassified data charts: **a** A sample bar chart classified as unknown; **b** A sample line chart classified as unknown

classified the charts with 93.3% accuracy. It did not misclassify any chart into another class but could not classify about 7% of the charts. Figure 8 shows some sample charts that were misclassified.

7.3.3 QoG computation

We obtained QoG measures for all the figures we tested. The values of the QoG measures intuitively matched the quality of the graphic being examined. In general, data charts with poor labels got lower QoG values compared to data charts with good labels, with all other aspects being approximately the same. The QoG measures varied from 0.25 to 0.82. The measures range from 0.44 to 0.68, 0.52 to 0.64, 0.37 to 0.70, 0.49 to 0.82, and 0.25 to 0.62, for pie charts, bubble charts, column charts, line charts, and bar charts, respectively. The figures below show some sample data charts and their QoG values. In Fig. 9, the QoG value is relatively high (0.68) for the column chart, since both the axes are labeled. In addition, all the structural elements are also labeled. The score is not very high, because the keyword-matching is not complete ("childbirth" was not mentioned in the description of the graphic).

The sample line chart shown in Fig. 10 has the structural elements (lines) labeled, but the axes are not labeled. Figure 10 was also mentioned relatively far away from the figure.

The bar chart shown in Fig. 11 has a low QoG value (0.35), since it does not have a caption, the axes are not labeled, it has a large character distance greater than 850 (i.e., it is located far away from where the figure is first mentioned), and has low keyword-matching.

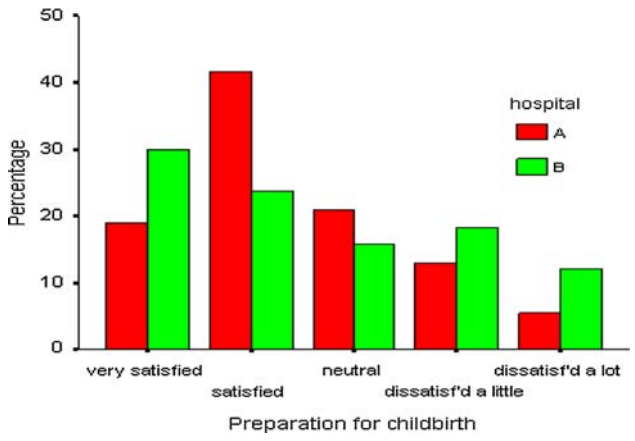


Fig. 9 A sample column chart with QoG = 0.68

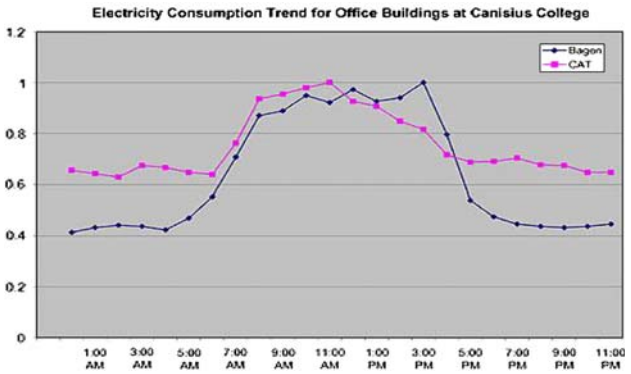


Fig. 10 A sample line chart with QoG = 0.56

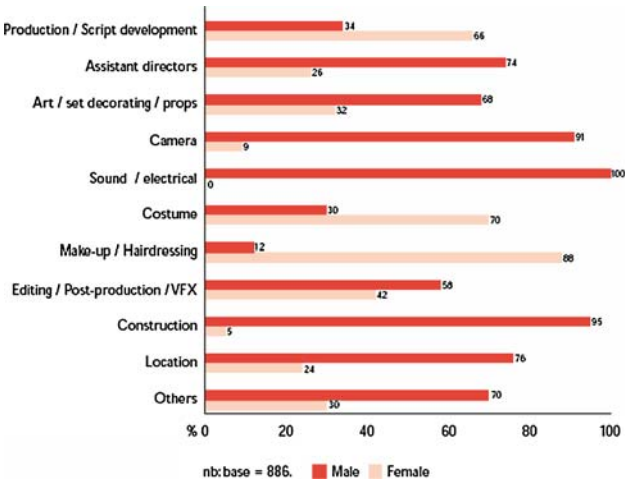


Fig. 11 A sample bar chart with QoG = 0.35

7.3.4 Evaluation using a human study

To further evaluate the approach, we undertook a human study. We used the same set of documents (15 documents from each of the five classes) that is used for automated analysis. We used a partially balanced incomplete block design for our study. A set of 25 adults were randomly chosen for the study. Each subject was given a set of three documents from each class to evaluate. The subjects were asked to evaluate the goodness of the graphic in the document using a seven-point Likert scale (7 = high quality and 1 = low quality). Thus, each document was examined by five subjects. We computed the Pearson correlation between the computed QoG scores using our approach and the average ratings for each document from the human study. The correlation values are summarized in Table 3.

The table shows that the computed QoG scores are highly correlated to scores given by human subjects. The bar charts and column charts were the most highly correlated and the pie and line charts were the least. The results thus validate the QoG computation proposed in this research.

Table 3 Correlation between QoG scores with ratings from human study

	Pie charts	Line charts	Bubble charts	Bar charts	Column charts
Correlation (p value)	0.62465 ($p = 0.0128$)	0.64568 ($p = 0.0093$)	0.73967 ($p = 0.0016$)	0.87675 ($p < 0.0001$)	0.87062 ($p < 0.0001$)

8 Summary and future work

This paper is one of the first attempts to determine the quality of graphics in a mixed-mode document. We have integrated research from several fields including image analysis, educational psychology, and human cognition. In this study, we formulated the problem of recognition and quality assessment of data charts in documents. We described five major classes of data charts that are widely used today. We also developed automated approaches to classify data charts into these five classes of charts based on their structural elements and also to compute their quality. The approach is based on cognition related theories, guidelines, and principles.

The steps in our automated processing include text-processing, graphics-processing, and text-matching. The algorithms for line extraction, methods for text-processing, and methods for graphics-processing are described in this article. We extract text-based and image-based descriptors for the embedded graphic, which are used to compute the overall quality of graphic (QoG). Experimental results show that our approach is efficient for recognition of data charts and effective for assessing their quality. A formal quantitative validation of the QoG measures by humans is beyond the scope of this work. This may be based on surveys of population samples, which can also be used to establish a scale for QoG measures. The quality measures and consequently the QoG measures can be tuned according to the data obtained. Experimental results show that a large number of documents do not have very high QoG implying that they are not designed for graphics quality.

Future work: The recognition and quality assessment of graphics is a complex problem and this is an early work in this direction. The scope of evaluating the quality of graphics is immense and much more work can be done in this area. The problem of graphics goodness or quality is going to get worse. Automated software to generate graphics is now ubiquitous and is used by novices and experts alike to create graphics and to embed them in multimedia documents. However, the users do not know what is important in creating an effective graphic. Our work can be extended in many different directions, some of which are listed below.

Our focus is on single page documents and the approach can be extended to multipage documents (scanned documents, pdf documents, etc.), since the distances are based on number of characters. Our work can be extended to other types of data charts, graphics, and multimedia. This includes developing graphic-specific quality measures. Including support for 3-D figures such as bar charts made with cones or pyramids would also be useful. Mixed data charts, pie charts within bubble charts, and pie charts within line charts are getting popular today and our work can be extended to include them as well. Color is an integral part of data charts today and formulation of additional color-related perceptual measures, e.g., tone, color density, would also be useful. Color perception and retention are vast independent areas of research by themselves and can be used to develop more color-related quality measures. The text recognition from graphics can be broadened to include a variety of font types and document layouts. Developing a *Graphics Quality Advisor* along the same lines as the spell checker tool available in MS Word can be used to advise people on the quality of graphics that they create.

References

1. Arnheim, R.: Entropy and Art, Disorder and Order. University Of California Press, Berkeley (1971)
2. Bertin, J.: Semiology of Graphics. University of Wisconsin Press, Wisconsin (1983)
3. Black, P.E.: Dictionary of Algorithms and Data Structures. NIST (2005)
4. Boyer, R.S., Moore, J.S.: A fast string searching algorithm. Commun. ACM **20**, 762–772 (1977)
5. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. **8**(6), 679–698 (1986)
6. Carriero, C., Futrelle, R., Nikolakis, N., Tselman, M.: Informational diagrams in scientific documents, AAAI Symposium: Reasoning with Diagrammatic Representations, Stanford University, pp. 185–188 (1992)
7. Coll, R., Coll, J., Thakur, G.: Graphs and tables: A four-factor experiment. Commun. ACM **37**(4), 77–86 (1994)
8. Crochemore, M., Rytter, W.: Text Algorithms. Oxford University Press, New York (1994)
9. Doermann, D.: An introduction to vectorization and segmentation. In: Graphics Recognition: Algorithms and Systems, Lecture Notes in Computer Science, vol. 1389, pp. 1–8. Springer, Berlin (1998)
10. Futrelle, R.: The conversion of diagrams to knowledge bases. In: IEEE Workshop on Visual Languages, pp. 240–242 (1992)
11. Futrelle, R., Nikolakis, N.: Efficient analysis of complex diagrams using constraint-based parsing. In: Proceedings of Third International Conference on Document Analysis and Recognition, Montreal, Canada, pp. 782–790 (1995)
12. Futrelle, R.: Ambiguity in visual language theory and its role in diagram parsing. In: Proceedings of 1999 IEEE Symposium on Visual Languages, Tokyo, Japan, pp. 172–175 (1999)
13. Jain, R., Kasturi, R., Schunk, B.: Machine Vision. McGraw-Hill, New York (1995)

14. Kaneko, T.: Line structure extraction form line-drawing images. Pattern Recognit. **25**(9), 963–971 (1992)
15. Kasturi, R., Tombre, K. (eds): Graphics recognition: methods and applications, Lecture Notes in Computer Science, vol. 1072, pp. 190–203. Springer, Berlin (1996)
16. Knuth, D.E., Morris, J.H., Pratt, V.R.: Fast pattern matching in strings. SIAM J. Comput. **6**(2), 323–350 (1977)
17. Lewandowsky, S., Spense, I.: The perception of statistical graphs. Sociol. Methods Res. **18**(2, 3), 200–242 (1989/1990)
18. Li, L.: Adaptive text/line separation in document images based on vectorization and OCR, MS Thesis, University of Nebraska Lincoln (1998)
19. Li, L., Nagy, G., Samal, A., Seth, S., Xu, Y.: Integrated text and line-art extraction from a topographic map. Int. J. Document Anal. Recognit. **2**(4), 177–185 (2000)
20. Martinez-Perez, M.P., Jimenez, J., Navalon, J.L.: Thinning algorithm based on contours. Comput. Vis. Image Process. **39**, 186–201 (1987)
21. Mayer, R.E.: Multimedia Learning. Cambridge University Press, New York (2001)
22. Myers, G.K., Mulgaonkar, P., Chen, C., DeCurtins, J., Chen, E.: Verification based approach for automated text and feature extraction from raster-scanned maps. In: Kasturi, R., Tombre, K. (eds) Graphics Recognition: Methods and Applications. Lecture Notes in Computer Science, vol. 1072, pp. 190–203. Springer, Berlin (1996)
23. Nagasamy, V., Langrana, N.A.: Engineering drawing, processing and vectorization system. Comput. Vis. Graph. Image Process. **49**(3), 379–397 (1990)
24. Nagy, G., Xu, Y.: Automatic prototype extraction for OCR. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Ulm (1997)
25. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
26. Petre, M.: Why looking isn't always seeing: Readership skills and graphical programming. Commun. ACM **38**(6), 33–44 (1995)
27. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3), 130–137 (1980)
28. Roth, S., Mattis, J.: Data characterization for intelligent graphics presentation. In: Proceedings of ACM SIGCHI 1990 Conference on Human Factors in Computing Systems, Seattle, Washington, pp. 193–200 (1990)
29. Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall, Upper Saddle River (2001)
30. Shimotsuji, S.: A Robust Drawing recognition system based on contour shape analysis. In: 10th International conference on Pattern Recognition, pp. 717–719 (1990)
31. Smith, R.W.: Computer processing of line images: A survey. Pattern Recognit. **20**, 7–15 (1987)
32. Sonka, M., Hlavac, V., Boyle, R.: Image Procession, Analysis and Machine Vision. Brooks/Cole, Pacific Grove (1999)
33. Strothotte, C., Strothotte, T.: Seeing between the Pixels, Pictures in Interactive Systems. Springer, Berlin (1997)
34. Suzuki, S.: Graph based vectorization method for line patterns. In: IEEE Computer Vision and Pattern Recognition, pp. 616–621 (1998)
35. Tufte, E.R.: The Visual Display of Quantitative Information. Graphics Press, Chesire (1983)
36. Tufte, E.R.: Envisioning Information. Graphics Press, Chesire (1990)
37. Tufte, E.R.: Visual Explanations: Images and Quantities, Evidence and Narrative. Graphics Press, Cheshire (1997)
38. Umbaugh, S.: Computer Imaging: Digital Image Analysis and Processing. Taylor & Francis, New York (2005)
39. Weisstein, E.W.: Zipf's Law, MathWorld (1999)
40. Xu, Y.: Prototype Extraction and OCR. Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy (1998)
41. Zhou, Y., Tan, C.: Bar charts recognition using Hough based syntactic segmentation. In: Proceedings of Diagrams 2000, LNAI, vol. 1889, pp. 494–497. Springer, Berlin (2000)
42. Zhou, Y., Tan, C.: Chart analysis and recognition in document images. In: Procedings of Sixth International Conference on Document Analysis and Recognition (ICDAR 2001), Seattle, Washington, pp. 1055–1058. IEEE Computer Society Press, New York (2001)