

レイアウト構造知識を用いた棒グラフの認識

横 倉 直 子[†] 渡 邊 豊 英[†]

棒グラフはビジネス・グラフの一種であり、棒グラフの認識は図面認識の1つと見なされる。一般に、図面認識では認識対象のモデルを構成することが困難で、今日まで試みられた機械部品図面、論理回路図面などではボトム・アップ主体の認識法が用いられてきた。しかし、本稿では、棒グラフにもそれを構成するグラフ要素間に何らかの構成規則があるとして、レイアウト構造を設定し、そのレイアウト構造に関する知識を用いた構造認識の手法を提案する。このレイアウト構造知識をネットワークで表し、これをレイアウト・ネットワークと呼ぶ。レイアウト・ネットワークでは、棒グラフの構造を統一的に表現し、グラフ要素間の位置関係を解釈することによって様々な表現形態の棒グラフを認識可能とする。本稿では、このような認識法に従って試作システムを実装し、その実験結果を示して本アプローチの有効性を明らかにする。

Recognition of Business Bar-graphs Using Layout Structure Knowledge

NAOKO YOKOKURA[†] and TOYOHIDE WATANABE[†]

Bar-graphs are a kind of business graphs, which are useful to present the meaningful contents visually, and the recognition approach is generally categorized into graphics recognition. Also, many approaches in the graphics recognition focused on the bottom-up processing paradigm because it is very hard to construct the recognition model, even if the targets to be recognized were any objects. In this paper, we propose an experimental recognition approach, based on the top-down processing paradigm in the interpretation step of composite elements though the extraction step of the composite elements from input images is organized in the traditional bottom-up processing. Our idea is derived from the assumption that bar-graphs have their own uniformly integrated layout structure by themselves. This layout structure is represented in a network form, in which the nodes indicate individual graph elements and the edges point out logical relationships among graph elements, concerning the neighboring locations. This paper discusses such a layout network and recognition method based on this layout network, and shows that our approach is very successful through some experiments.

1. はじめに

紙面に記述された、または描写された情報の電子化は、文書理解、図面認識などの分野で今日まで種々の文書・図面に対して試みられた^{1),2)}。従来の画像処理に依存したアプローチから知識情報処理に活路を求めたアプローチは大いに注目を集め³⁾、複雑な認識対象、高度な認識能力、優れた認識結果が今日の研究・開発の目標となっている。特に、文書理解では文書モデルに基づいたモデル駆動型処理が、データ駆動型処理に比べて注目され、その報告も多い⁴⁾。モデル駆動型処理はデータ駆動型処理に比べて、認識対象に対する論理的な処理能力を明確化し、モデルの解釈手続きにも

依存するが、適用性、柔軟性に富んでいる⁵⁾。一方、図面認識ではモデル駆動型処理はほとんど報告されていない。認識すべきオブジェクトを論理特性と物理特性から処理可能な形式で明示的に定義できないことが、モデルの構成を困難にしている。

項目間の量的比較、項目値の時間的な変化などを視覚的に描写するビジネス・グラフは機械部品設計図面などに比べて、その描写にはある程度規格化があるが、様々な表現が可能となっている。ビジネス・グラフを認識対象とした研究に、Futrelleらの研究⁶⁾やLeeらの研究⁷⁾がある。彼らの研究では、グラフ要素の認識、表現内容の解釈を目的とし、データ駆動型アプローチに基づいてグラフ要素を抽出し、手続き的な認識・解釈法の下に認識システムを構成している。個々のグラフ要素の抽出・識別、およびそれらの解釈はグラフ構造に依存して実現され、認識対象であるグラフの形式・

[†]名古屋大学大学院工学研究科情報工学専攻
Department of Information Engineering, Graduate
School of Engineering, Nagoya University

表記, およびグラフ要素の特性・位置などを限定している。

本稿では, ビジネス・グラフ, 特に棒グラフを認識対象としてモデル駆動的に処理するための枠組みを提案し, その試作システム, および実験結果について報告する^{8)~11)}。我々の基本的なアイデアは棒グラフにレイアウト構造が付帯しているとし, そのレイアウト構造に関する知識を用いてモデル駆動的に処理するアプローチを採用することである。このレイアウト構造知識をネットワークで表現し, レイアウト・ネットワークと呼ぶ。レイアウト・ネットワークではグラフ要素を節で, グラフ要素間の空間的位置関係を枝で表している。

2. 棒グラフ認識の枠組み

棒グラフはビジネス・グラフの一種であり, 縦軸, 横軸の下に棒として表示された項目値を明示している。もちろん, 棒グラフといっても様々な表現形態がある。それは表示記号を規格化してそれぞれの構成要素とその関係を明示化した機械部品設計図面, 建築図面とは異なっている。描写者の意図, 目的, 工夫, 感性などによって同一表示内容でも種々様々な表現, 構成, 形式のグラフが生成される。

本研究では, 学会誌, 論文誌, 国際会議録に用いられている棒グラフを直接の認識対象とする。

2.1 棒グラフの特徴

本研究で認識対象とする棒グラフは, たとえば図1に示すようなものである。それぞれ異なった表現形態が観察され, またそこで使用されているグラフの構成要素(以降, グラフ要素と呼ぶ)にも相違がある。たとえば, 図1(a), (b), (d)は縦軸に沿って, 図1(c)は横軸に沿って棒が描かれている。また, 図1(a)は1つの項目の値に対して, 図1(b)は複数の項目に対して, 図1(c), (d)は複数の項目の属性に対して描かれている。それ以外にも, 様々な相違を観察できる。

棒グラフを認識する場合, これらの相違を明示的に判断できるための情報や, 処理手続きが必要となる。図1のグラフに対して, そのグラフ要素をとらえるとき, グラフ要素はグラフの特性に応じて様々な表現形態で用いられているが, 基本的にその意味内容, 表示関係が大きく異なることはない。たとえば, 図2に示した棒グラフは数種のグラフ要素から構成されている。

グラフ要素を便宜的に図形要素と文字列要素に分け, それぞれ図形表現と認識されるグラフ要素, 文字列表現と認識されるグラフ要素と定義する。この区分は認識システムを構成するときにグラフ要素の認識手順に

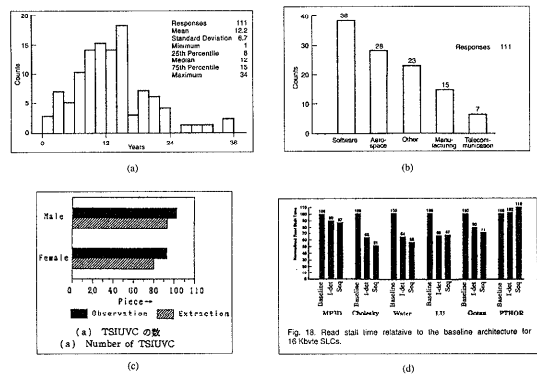


図1 様々な棒グラフの例
Fig. 1 Examples of bar-graphs.

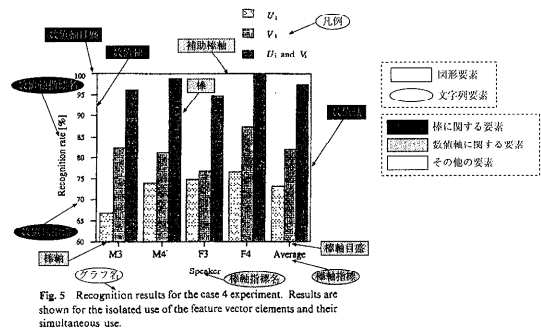


図2 棒グラフ要素の例
Fig. 2 Examples of graph-elements.

関係する。また, 図2に示すように, グラフ要素相互の役割から, 棒に関する要素, 数値軸に関連する要素, その他の要素に区分することもできる。これはグラフを解釈する場合に有用で, これらに基づいてグラフ要素相互の依存関係を明示的に整理できる。表1は個々のグラフ要素の特徴を整理している。

2.2 グラフ構造の知識表現

グラフ要素は相互に関連して1つの棒グラフを構成する。これはグラフの表現形態が変わっても相互に関連するグラフ要素によって統一的な意味解釈を可能とする。すなわち, 関連するグラフ要素はグラフ構造で配置される位置に依存関係がある。グラフ要素間の位置関係を論理的に把握することにより, レイアウト構造を定めることができる。このレイアウト構造に関する知識表現をネットワークで表し, これをレイアウト・ネットワークと呼ぶ。レイアウト・ネットワークではネットワークを解釈する手順を表すために, 節として表されたグラフ要素間の枝に方向性を与える。

レイアウト・ネットワークは図3に示すように3種のサブネットから構成され, それぞれ図3(b)のデータ・サブネット, 図3(c)の数値軸サブネット, 図3(d)

表 1 グラフ要素の定義
Table 1 Features of graph elements.

名称	役割	区分 1	区分 2
棒	対象とする項目の数値の長さの表示	図形	棒関連
棒軸	棒の基点の表示	図形	〃
補助棒軸	棒の上限を示す枠線の表示	図形	〃
数値	対応する棒の長さの文字列表現	文字列	〃
棒軸指標, 補助棒軸指標	対応する棒の項目提示	文字列	〃
棒軸指標名, 補助棒軸指標名	対象とする項目域の提示	文字列	〃
棒軸目盛, 補助棒軸目盛, 棒軸目盛線	棒と棒軸指標の対応関係, または項目間の区切り表示	図形	〃
数値軸目盛, 目盛線	棒の長さのスケールの表示	図形	数値軸関連
数値軸	数値軸目盛の基点の表示	図形	〃
数値軸指標	対応する目盛が表す数値提示	文字列	〃
数値軸指標名	対象とする項目の領域の提示	文字列	〃
凡例	棒が数種のパターンで構成されている場合, 各パターンと項目の対応を提示	文字列	その他
補助説明文	グラフの内容に対する補助的な説明表現	文字列	その他
グラフ名	グラフの名称表現	文字列	その他

の付加サブネットである。レイアウト・ネットワークを階層的な構造(サブ・ネットの集まり)として構成することにより, レイアウト構造知識を表現でき, またその解釈の効率化を図ることができる。もちろん, レイアウト・ネットワークは縦方向, 横方向の棒グラフにも対処した記述内容となっていて, 様々な表現形態のグラフのレイアウト構造を表現している。このレイアウト・ネットワークは棒グラフの特徴を最も表すグラフ要素「棒」を中心に表現され, 棒グラフの解析はこの棒(または棒グループ)から連結されるグラフ要素を照合・判定して実行される。したがって, データ・サブネットがレイアウト構造として中核的な表現となっている。データ・サブネットに連結される数値軸サブネット(棒軸が縦軸であれば左右の一方, または両方に, 横軸であれば上下の一方, または両方に存在する)は棒グラフの描写の多様性に対処するために分離して構成されている。

[枝]

図 3 で, 「右, 左, 上, 下」などの枝ラベルはグラフ要素間の物理的な位置関係でなく, 「棒軸からとらえ

た棒の方向を上」と定め, 棒軸に対して相対的に個々のグラフ要素間の位置関係を表している。たとえば, 図 4 (a) の縦方向のグラフではグラフ画像の上方向が「上」となり, 図 4 (b) の横方向のグラフではグラフ画像の右方向が「上」となる。また, 枝に付した「対応, 交互」は以下のように定義付けられる。

(a) 対応: 1 つの終点節のグラフ要素は 1 つの始点節のグラフ要素に制約される。たとえば, 棒軸目盛が棒に対応する位置に配置される場合, ある棒軸目盛の存在範囲は 1 つの棒の存在範囲内である(図 5 (a) 参照)。

(b) 交互: 1 つの終点節のグラフ要素は 2 つの始点節のグラフ要素に制約される。たとえば, 棒軸目盛が棒と交互に配置される場合, ある棒軸目盛の存在範囲は 2 つの棒の間となる(図 5 (b) 参照)。

さらに, 2 つ以上の節を関係付ける枝には, 「接触, 非接触, 交差」が付帯する。「接触」は枝で結ばれた 2 つの節に対応するグラフ要素が接触することを表し, 「非接触」はそうでないことを表す。同様に, 「交差」は 2 つのグラフ要素が交差することを表す。これらの区別は図 3 では枝の種類として示されている。

[節]

個々の節には対応するグラフ要素の名称以外に, 図 3 に示す属性が付帯し, 区別されている。「単数, 複数」は対応するグラフ要素に繰返性があるか否かを表す。これら以外に, (i) 軸と幅の比, (ii) 優先順位などがある。「軸と幅の比」は図形要素固有の属性であり, 図形要素によっては目盛や目盛線のように軸と幅がほぼ等しいもの, 棒のように幅が軸よりも広いものなどがある。したがって, 軸に対して図形要素の大きさを相対的に指示する。「優先順位」は or 節特有の属性であり, or で結ばれた節に対応するグラフ要素が同一要素であるが, 配置される位置に関連して異なることがある場合に, その出現の可能性の順位を指示する。たとえば, 数値軸指標名は数値軸に依存して配置される場合と, 数値軸指標に依存して配置される場合がある(図 3 (c) 参照)。したがって, 配置される可能性が異なるので, その可能性が高い方に高優先値を与える。さらに, グラフ要素によってはそれ固有の性質が一般に観察され, それらは以下のようなものである。

- (1) 棒軸指標: 複数行にわたって記述される場合がある。
- (2) 棒グループ: 複数の棒が接触して描かれる場合に棒グループと呼ぶ。
- (3) 棒軸グループ指標: グラフ画像内に棒グループがなければ棒軸グループ指標も存在しない。

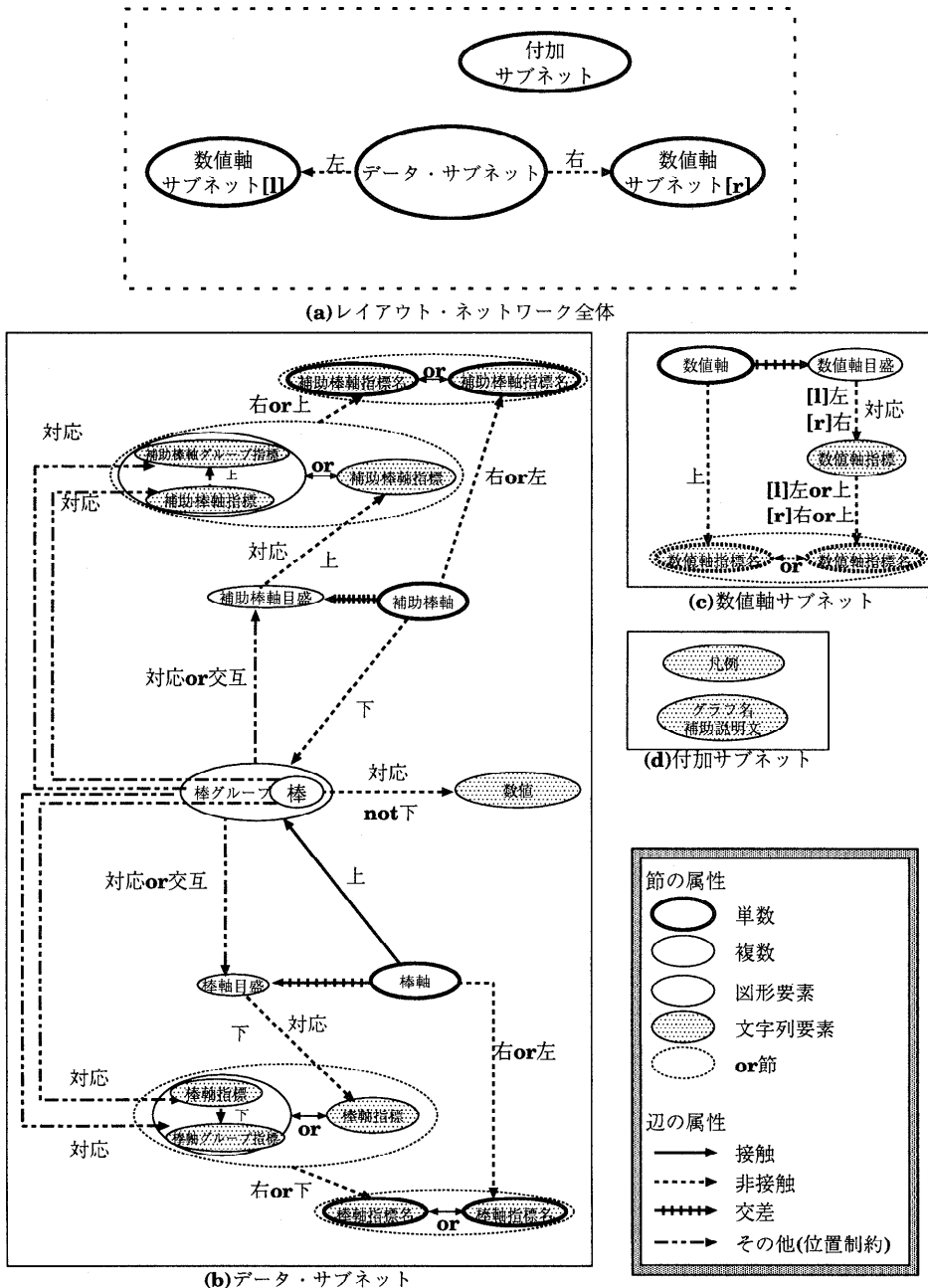


図3 レイアウト・ネットワーク
Fig. 3 Layout network.

このような節と枝の種別によって、グラフ構造に対する論理的、物理的な構成をレイアウト・ネットワークとして表現している。ここで、節や枝に付帯するすべての情報は、位置や閾値などの物理的な値ではなく、関係などの論理的な値となっていて、様々な形状、構成にも対処可能である。

3. グラフ構造認識処理の概要

グラフ構造認識処理は構成要素抽出処理と構造解析処理に分けられる(図6参照)。構成要素抽出処理は入力グラフ画像からその構成要素を抽出する。構造解析処理は抽出された構成要素からレイアウト構造知識を適用し、グラフ要素を識別し、対象グラフの構造を

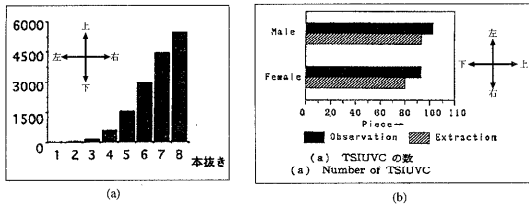


図4 グラフの方向
Fig. 4 Direction of bar-graphs.

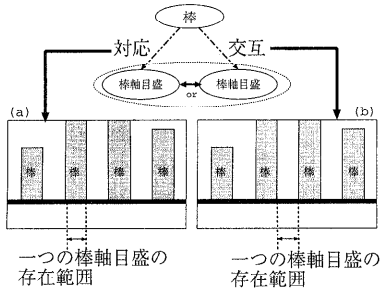


図5 位置制約
Fig. 5 Constraints for graph elements.

解析する。構成要素は画像処理的に抽出されたグラフ要素の候補である。一方、グラフ要素は表1に対応している。構成要素の抽出処理はグラフの画像特徴に従ったボトム・アップ処理であるのに対し、構造解析処理はレイアウト構造知識に従ったトップ・ダウン処理である。

[角度補正処理]

構成要素抽出処理は入力されたグラフ画像内の横軸・縦軸が水平か垂直方向に対して傾いていないことを前提としている。そこで、前処理として入力画像の角度補正を行う。棒グラフ内の文字はすべて水平か垂直方向に並べられ、文字列を形成している。もし入力画像が水平方向に対して傾いている場合、グラフ内の文字列も水平か垂直方向に対して傾いている。したがって、O'Gormanの手法¹²⁾を用いて入力画像内の文字列の傾きを求め、入力画像の傾きを修正する。

[構成要素抽出処理]

構成要素抽出処理は入力画像の物理的特徴に注目して、まず最初に入力画像内の黒画素域を図形要素候補と文字列要素候補に分け、それぞれについて構成要素を抽出する(図7参照)。

図2から分かるように、棒グラフ内の図形要素はすべて接触し、直線または矩形で表現されている。直線は中身が塗りつぶされた矩形と見なすと、図形要素はすべて矩形で表現される。したがって、図形要素の抽出処理は図形要素候補を矩形に分割する処理となる。

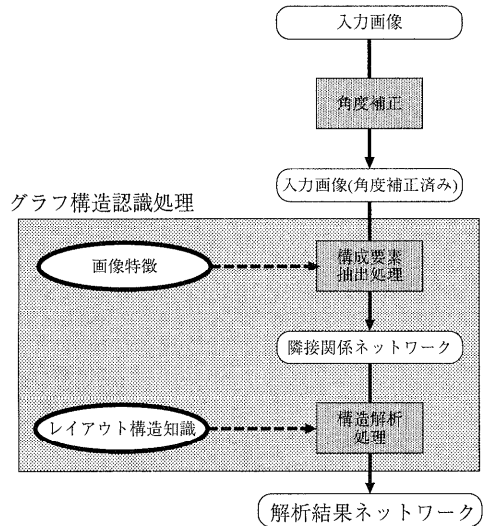


図6 グラフ構造認識処理の流れ
Fig. 6 Processing flow.

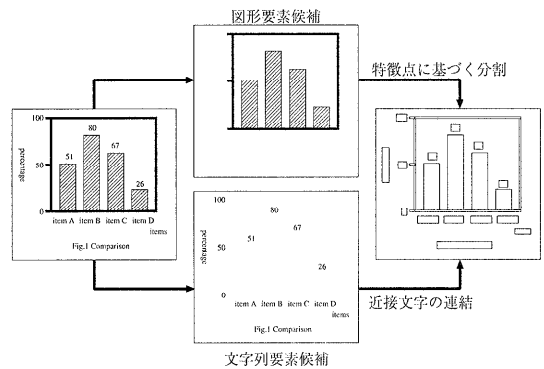


図7 構成要素抽出処理の概要
Fig. 7 Extraction of composite elements.

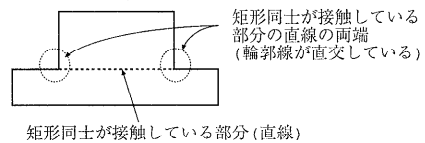


図8 矩形同士が接触する部分
Fig. 8 Interconnected parts between different graphic elements.

矩形同士が接触している部分は直線であり、この直線の両端に相当する点では、輪郭線が直角に交わっている(図8参照)。本処理では、図形要素候補の輪郭線が直角に交わる点を見つけ、その点を両端とする直線によって図形要素候補を、構成要素に切り分ける(図9参照)。

一方、棒グラフ内の文字はすべて水平か垂直方向に並べられ、文字列を形成している。水平か垂直方向に近接している文字同士を連結することにより、文字列

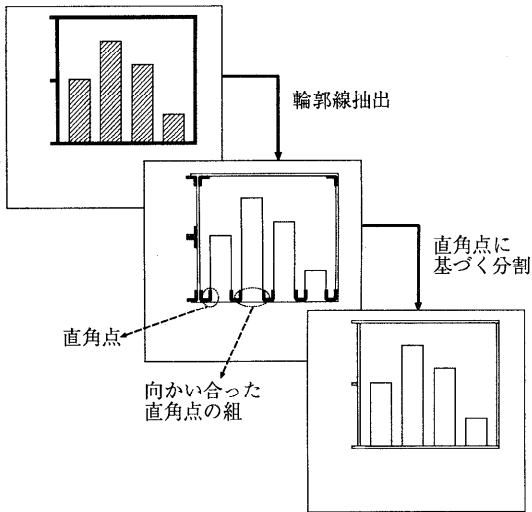


図 9 図形要素候補の抽出

Fig. 9 Extraction of candidates of graphic elements.

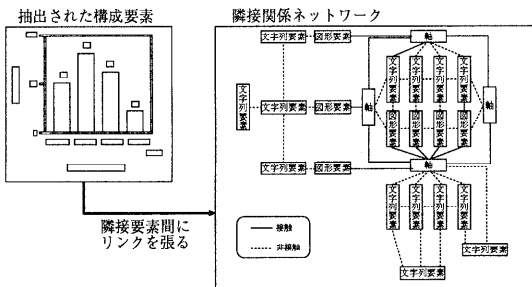


図 10 隣接関係ネットワークの生成

Fig. 10 Generation of neighboring relationship network.

要素候補を抽出できる。ただし、水平か垂直方向の文字列が入り混じっているため、単に近接している文字同士を連結するのでは、誤抽出が発生する。そこで、各文字について、近接する文字を水平か垂直方向の両方に探し、1つの文字列を生成する可能性の高い方向に文字を連結する。

構成要素抽出処理では構成要素の抽出後、レイアウト・ネットワークとの比較のために、抽出した構成要素から隣接関係ネットワークを抽出結果として構成する(図 10 参照)。隣接関係ネットワークは抽出された構成要素を節とし、棒グラフ画像上で最も隣接した構成要素間を枝で連結したネットワークである。隣接関係ネットワークの節は、属性として構成要素の位置および種類(図形要素/文字列要素)を、枝は属性として2つの構成要素が接触しているか否かおよび枝の向き(縦方向または横方向)を持つ。

[構造解析処理]

構造解析処理は構成要素抽出処理の結果として得ら

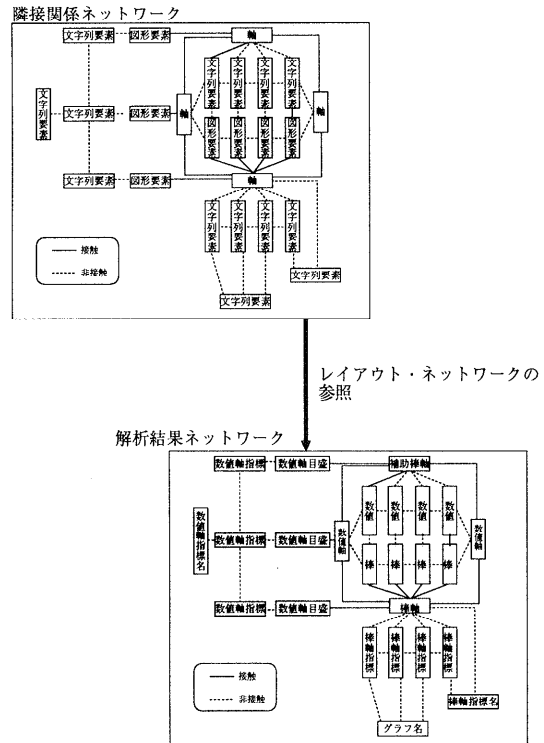


図 11 構造解析処理

Fig. 11 Structure analysis.

れた隣接関係ネットワークをレイアウト・ネットワークと比較し、隣接関係ネットワークの節に対応する構成要素をグラフ要素として対応付ける(図 11 参照)。この結果、隣接関係ネットワークは対象とするグラフの構造を表した解析結果ネットワークへと変換される。

4. 構成要素抽出処理

図形要素と文字列要素では抽出方法が異なるため、まず最初に入力画像内の黒画素を図形要素候補と文字列要素候補に分割する。

4.1 図形要素の抽出

図形要素の抽出手順は以下のとおりである(図 12 参照)。

- (1) 図形要素候補の輪郭線を抽出する。
- (2) 輪郭線が直交する点を抽出する。
- (3) 輪郭線の直交点を2点1組とし、図形要素候補を分割する境界線を決める。
- (4) 境界線に従って図形要素候補を分割し、構成要素を抽出する。

(3)の処理では、輪郭線の直交点の属性を利用して直交点の組を決める。輪郭線の直交点の属性を図 13 に示す。たとえば、属性 *right-upper* は黒画素の右端にあたる輪郭線と、黒画素の上端にあたる輪郭線が

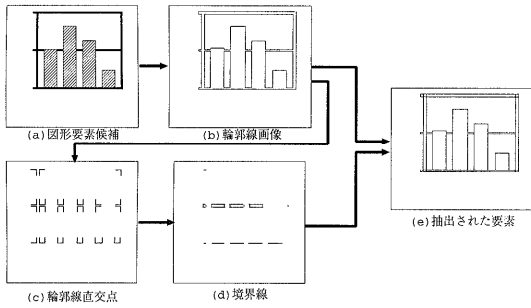


図 12 図形要素の抽出処理

Fig. 12 Processing of graphic elements.

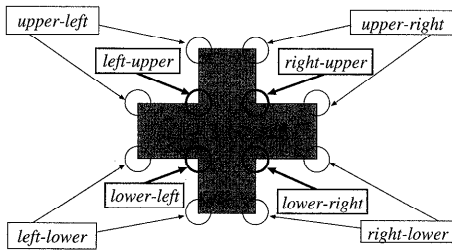


図 13 輪郭線が直交する点の属性

Fig. 13 Attributes of points intersected by two different line segments.

直交する点で、さらに右端輪郭線が上端輪郭線よりも上側にある点を示す。矩形と矩形が交わっている直線の両端に相当する点は輪郭線が凹型に交わり、つまり *right-upper*, *lower-right*, *lower-left*, *left-upper* を属性として持つ (図 13 の太線で囲まれた点)。これらの属性を持つ点を効率良く抽出するために、輪郭線を上端輪郭線、下端輪郭線、左端輪郭線、右端輪郭線に分けて抽出した後、属性 *right-upper*, *lower-right*, *lower-left*, *left-upper* を抽出する。

4.1.1 輪郭線の抽出

輪郭線は図 14 に示すフィルタを利用して抽出する。このフィルタと照合された図形要素候補画像において、A に対応する画素が黒となるのは次の 2 つの条件が満たされたときである。

[条件]

- (1) A に対応する 3 画素がすべて黒である。
- (2) C に対応する 3 画素のうち、少なくとも 1 画素が白である。

このフィルタリング処理を図形要素候補画像の左上から 1 画素ずつ右方向に調べ、1 列ごとに走査する。得られた結果の黒画素にはどのフィルタで得られた黒画素か、つまり上端、下端、左端、右端のどの輪郭線であるかを属性として与える。

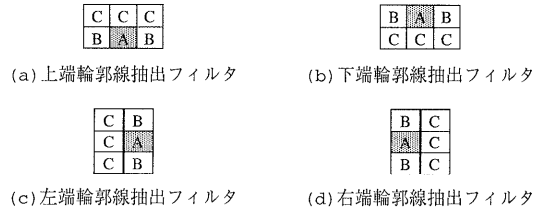
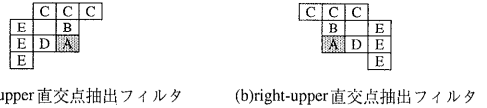
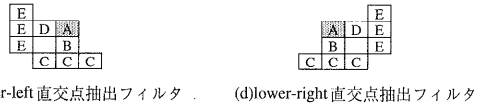


図 14 輪郭線抽出フィルタ

Fig. 14 Filters for extracting contour line segments.



(a)left-upper 直交点抽出フィルタ (b)right-upper 直交点抽出フィルタ



(c)lower-left 直交点抽出フィルタ (d)lower-right 直交点抽出フィルタ

図 15 輪郭線直交点抽出フィルタ

Fig. 15 Filters for extracting intersections of contour line segments.

4.1.2 輪郭線直交点の抽出

属性 *right-upper*, *lower-right*, *lower-left*, *left-upper* の輪郭線直交点は図 15 のフィルタを利用して抽出する。このフィルタと照合された輪郭線画像において、A に対応する画素が *left-upper* の直交点と判断されるのは次の条件が満たされたときである。

[条件]

- (1) B に対応する画素、および C に対応する 3 画素のうちの 1 画素が左端輪郭線である。
- (2) D に対応する画素、および E に対応する 3 画素のうちの 1 画素が上端輪郭線である。

他の輪郭直交点についても同様に判断する。

4.1.3 境界線の決定

接触している矩形同士を分割する境界線は、1 組の輪郭線直交点から得られる。輪郭線直交点の組合せは以下の 4 通りである (図 16 参照)。

- *left-upper* と *right-upper* (upper line)
- *left-upper* と *lower-left* (left line)
- *right-upper* と *lower-right* (right line)
- *lower-left* と *lower-right* (lower line)

隣り合っている輪郭線直交点の組のうち、以上の組合せの組を端点とする直線を生成する。

4.1.4 図形要素候補の分割

軸は棒グラフ内で最も重要なグラフ要素であり、図 17 に示すように軸の位置はヒストグラムのピークとなって現れる。まず軸を優先的に切り出し、その後に残りの要素を切り分ける。図形要素候補の切出し

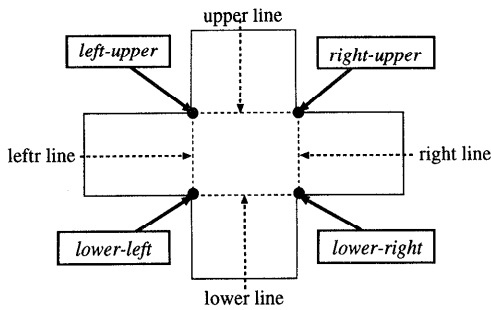


図 16 輪郭線直交点の組合せ

Fig. 16 Combination for intersections of contour line segments.

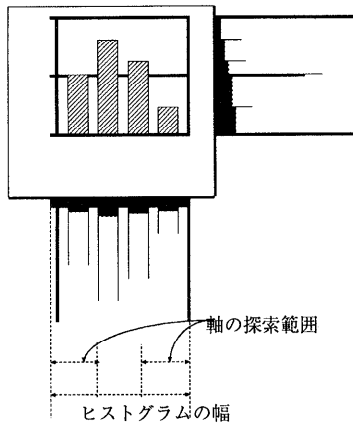


図 17 図形要素候補のヒストグラム

Fig. 17 Histogram of candidates of graphic elements.

手順は次のようになる。

- (1) 図形要素候補画像の垂直（水平）方向のヒストグラムを求める。
- (2) ヒストグラムの両端からそれぞれヒストグラムの幅（ヒストグラムで0以外の値を持っている幅）の3分の1以内の範囲（図17参照）で最も長いピークの位置を軸の位置とする。
- (3) 軸の位置の上下（左右）で最も近い垂直（水平）方向の境界線を求め、軸を切り出す。
- (4) 軸と重なっている境界線を取り除く。

軸を切り出した後、残っている境界線は長さが軸の幅とほぼ等しいか、長いかに分類できる。長さが軸の幅とほぼ等しい線は棒が目盛線を分断する線に相当し、長い線は目盛線が棒を分断する線に相当する。目盛線と棒が重なっている場合、目盛線が棒に隠され、分断される。したがって、軸を切り出した後に、必要な境界線は軸の幅とほぼ等しい長さを持つ境界線のみである。残りの要素は、軸とほぼ等しい長さを持つ境界線にしたがって切り分けられる。

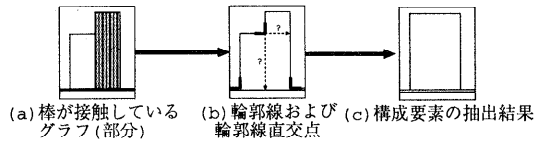


図 18 接触した棒

Fig. 18 Interconnected bars.

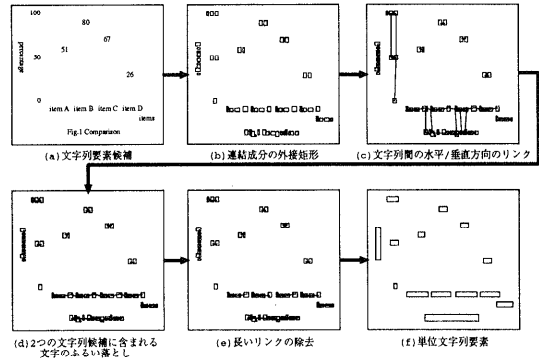


図 19 文字列要素の抽出処理

Fig. 19 Processing of character string elements.

棒が接触している場合、その接触部には輪郭線直交点の組が現れないので、接触している棒同士を切り分けることができない（図18参照）。1つの輪郭線直交点を基準に、図形要素候補を切り分けることもできるが、構成要素抽出処理ではその構成要素が棒であることを確定できないので、どの方向に切るかを決定することは困難である。したがって、構造解析処理において棒であることが認識された後に、それ以上分割できるかどうかを調べて対処する。

4.2 文字列要素の抽出

文字列要素の抽出手順は以下のとおりである（図19参照）。

- (1) すでに抽出されている図形要素と重なっている画素を除去する。
- (2) 各連結成分の外接矩形を求め、すべての矩形について垂直か水平方向に最も近い矩形とリンクを張る。1方向の連続したリンクで結ばれた文字の集まりを1つの文字列要素候補とする（図20参照）。
- (3) 2つの文字列要素候補に含まれる文字をふるい落とす。
- (4) 文字間のリンクのうち、比較的長いものを除去する。

4.2.1 不要な画素の除去

文字列要素候補として得られた画素のうち、図形要素と重なっているものを文字以外棒のハッチングの模

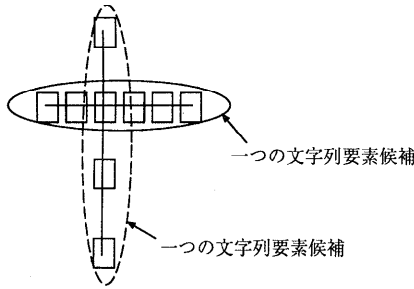


図 20 文字列要素候補

Fig. 20 Candidates of character string elements.

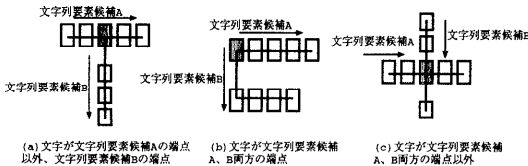


図 21 2つの文字列要素候補に含まれる文字のパターン

Fig. 21 Patterns among two different character string elements.

様と考え、除去する。

4.2.2 隣接文字間のリンク

棒グラフの文字は水平か垂直方向に配置されて文字列を構成する。前処理により画像の角度は補正されているので、同じ文字列に属する文字は水平または垂直方向に隣り合っている。したがって、各文字について水平または垂直方向に隣接している文字を求め、リンクを張る。ただし、文字が軸をはさんで連結されることはないで、軸をはさんだ文字は隣接していないものと見なす。同方向に連続したリンクによって結ばれた文字の集まりを文字列要素候補とする。

4.2.3 2つの文字列要素候補に含まれる文字のふり落し

すべての文字について、水平か垂直方向に隣接した文字を求めているので、1つの文字が2つの文字列要素候補に含まれる場合がある。この場合、どちらの文字列に含まれる確率が高いかを判断し、確率が低いリンクを除去する。2つの文字列要素候補に1つの文字が含まれる場合には次の3つのパターンがある(図21参照)。

- 文字が文字列要素候補 A の端点以外、文字列要素候補 B の端点の場合 (図 21 (a) 参照) : その文字と B のリンクを除去する。
- 文字が文字列要素候補 A, B の端点 (図 21 (b) 参照) : A の隣接する文字との文字間距離と、B の隣接する文字との文字間距離を求め、長い方の文字列のリンクを除去する。

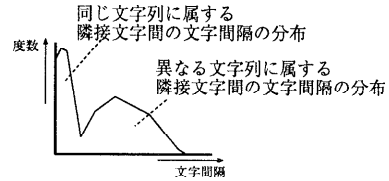


図 22 連結された文字の間隔のヒストグラム

Fig. 22 Histogram of gaps between linked characters.

- 文字が文字列要素候補 A, B の端点以外 (図 21 (c) 参照) : A の隣接する2つの文字との文字間距離の和と、B の隣接する2つの文字との文字間距離の和を求め、長い方の文字列のリンクを除去する。

4.2.4 長いリンクの除去

各文字について水平か垂直方向に隣接した文字を求める際、文字間の角度のみを調べているため、同じ文字列を構成しない文字同士が接続される場合がある。そこで、不必要なリンクを除去する。外接矩形間の距離、つまり文字と文字の間隔は、同じ文字列に属する文字間では比較的狭く、異なる文字列に属する文字間では比較的広がっている。また、同じ文字列に属する文字間隔はどの文字列についてもほとんど等しい。したがって、連結されている文字の間隔のヒストグラムを求めると、図 22 のような分布になる。ここで、「同じ文字列に属する文字間隔」と「異なる文字列に属する文字間隔」の閾値を求めるために判別分析法を用いる。

連結された文字の間隔のヒストグラムを求め、度数を正規化する。間隔 i であるリンクの数を n_i とすると、正規化された値は次式で表され、文字間隔のヒストグラムは確率分布と見なされる。

$$P_i = n_i / N; N = \sum_{i=0}^L n_i$$

(L は画像内の連結された文字間の最大文字間隔)

ある確率分布を2つのクラスに分けるときの最適閾値 t^* は次式で表される。

$$\sigma_B^2(t^*) = \max_{1 \leq t < L} \sigma_B^2(t)$$

ただし、 $\sigma_B^2(t)$, μ_T , $\omega(t)$, $\mu(t)$ はそれぞれ

$$\sigma_B^2(t) = \frac{[\mu_T \omega(t) - \mu(t)]^2}{\omega(t)[1 - \omega(t)]}, \mu_T = \sum_{i=1}^L iP_i,$$

$$\omega(t) = \sum_{i=1}^t P_i, \mu(t) = \sum_{i=1}^t iP_i$$

である。この式によって求めた閾値よりも大きい間隔

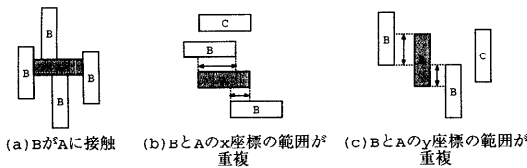


図 23 構成要素の隣接条件

Fig. 23 Condition for neighborhood among composite elements.

を持つリンクを除去する。最終的に残ったリンクで結ばれた文字の集まりを1つの文字列とし、それらの文字の集まりの外接矩形を文字列要素として抽出する。しかし、最終的に得られた文字列要素でも、複数の文字列が誤って1つの文字列として連結される場合がある。そこで、抽出された文字列要素には、その文字列がどのような文字の配置から構成されているかの情報を与える。後の処理で誤抽出の可能性が出てきた場合に、誤抽出の箇所を特定するためである。

4.3 隣接関係ネットワーク

隣接関係ネットワークの生成処理では、構成要素抽出処理で得られたすべての構成要素について、上下左右の隣接関係を求めてリンクを張る。構成要素 A に対して構成要素 B が次の条件の1つを満たしている場合、B は A に隣接している (図 23 参照)。

[条件]

- B は A に接触している。
- B は A と x 座標の範囲が重複し、最も近い。
- B は A と y 座標の範囲が重複し、最も近い。

隣接関係ネットワークの節は対応する構成要素の位置 (矩形の左上および右下座標) と種類 (軸/図形/文字列) を、枝は2つの構成要素が接触しているか否かと枝の向き (縦方向/横方向) を情報として付帯する。

5. 構造解析処理

構造解析処理では、構成要素抽出処理の結果として得られた隣接関係ネットワークと、レイアウト構造知識であるレイアウト・ネットワークを照合して、グラフ構造およびグラフ要素を識別する。処理手順は以下になる。

- (1) 軸を棒軸、補助棒軸、数値軸として認識する。これは、棒軸、補助棒軸、数値軸に対応する節がレイアウト・ネットワーク全体の開始節となっているためである。
- (2) レイアウト・ネットワークにおいて親にあたる節がすべて探索済みである節を注目節とし、その注目節に対応する構成要素に対して条件を生成する。

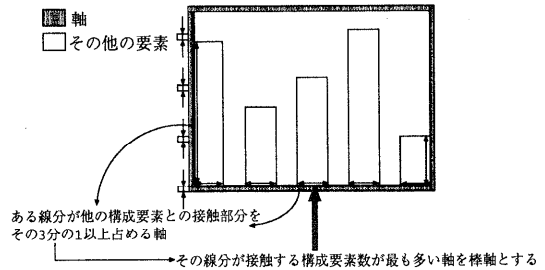


図 24 棒軸の条件

Fig. 24 Condition for bar-axis.

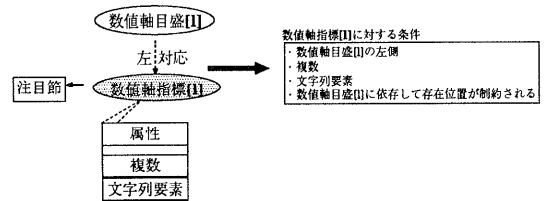


図 25 注目節に対応する条件

Fig. 25 Corresponding condition for focused node.

- (3) 条件を満たす構成要素を隣接関係ネットワーク内で探す。
- (4) 条件を満たす隣接関係ネットワーク内の節にあたる構成要素に対し、注目節にあたるグラフ要素の名前を与える。
- (5) (2) から (5) を隣接関係ネットワーク内のすべての構成要素が識別できるまで繰り返す。

5.1 軸の認識

棒軸、補助棒軸、および数値軸は構成要素抽出の段階で軸としてすでに認識されている。しかし、軸として抽出された構成要素が、実際の棒グラフにおいてどの軸に対応するかはまだ分かっていないので、構造解析処理では初めに軸を認識する。まず、棒軸に相当する軸を決め、それに平行な軸を補助棒軸、垂直な軸を数値軸として認識する。棒軸に対する条件は次のとおりである。

[条件]

ある線分が他の構成要素とその接触部分の長さを3分の1以上占めるものうち、その線分に接触している構成要素の数が最も多いものを棒軸とする (図 24 参照)。

5.2 注目節に対応する条件

注目節に対応する条件とは、注目節に対応する構成要素の存在位置に関する条件である。この条件は、注目節の親節 (注目節に入る枝の開始節) に対応する構成要素、注目節に入る枝の属性、注目節の属性から構成される (図 25 参照)。

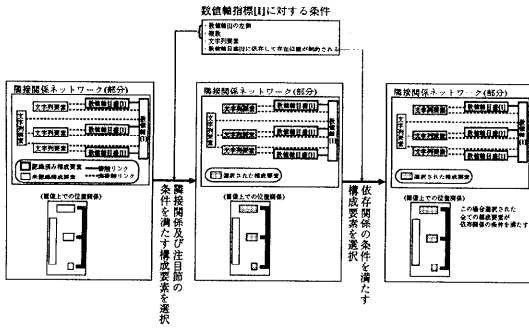


図 26 条件を満たす構成要素の探索
 Fig. 26 Search of composite elements.

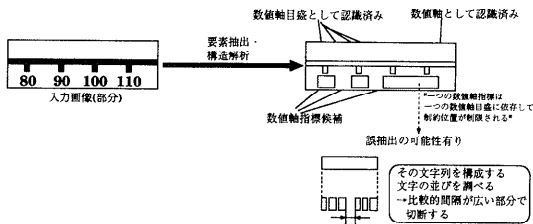


図 27 文字列の誤抽出
 Fig. 27 Mis-extraction of character strings.

5.3 条件を満たす構成要素の探索

条件を満たす構成要素は次の順番で探索される(図 26 参照).

- (1) 隣接関係ネットワークの中から隣接関係および注目節の属性の条件を満たす構成要素を選択
- (2) 選択された構成要素の中から依存関係の条件を満たす構成要素を選択

(2) の依存関係の条件を満たす構成要素の選択では図 5 の制約を満たすように構成要素を選択する. たとえば, 図 25 に示すように, 数値軸指標は数値軸目盛に依存して存在位置が制約される. 図 27 のような場合, 誤抽出された構成要素は正しく数値軸指標として認識されない. そこで, 文字列要素の場合は, もとの文字の情報を持っているので, このような誤抽出の可能性がある場合に文字間隔の情報を利用して誤抽出を訂正する.

5.4 グラフ要素の識別

選択された構成要素に, 注目節に対応するグラフ要素の名前を与えることにより, 認識が完了する. 注目する構成要素が or 節の場合は, or で結ばれたすべての節について上記の処理を行い, 対応する構成要素を選択した後, 節に与えられた優先順位や選択された構成要素の数に従って, グラフ要素を選択する.

6. 実験結果および考察

以上に述べた方法に従って, 実際のグラフ画像 10 枚に対してグラフ構造認識処理を行った. グラフ画像は 400 dpi, 2 階調でスキャナにより入力した. これらのグラフ画像は電気情報通信学会, 情報処理学会, IEEE, ACM などの論文誌と, IPPS, ICCP, ICDCS などの会議録から収集した. 試作システムの実装は Sun SparcStation 上で C 言語により行った. 図 28 は入力画像の例, 図 29 はその入力画像に対する角度補正の結果, 図 30 は構成要素の抽出結果, 図 31 は構造解析処理の結果である. 図 31 では, 黒く塗りつぶされた部分がグラフ要素として認識されたことを示す. 表 2, 表 3 に各処理の結果をまとめた.

角度補正処理の失敗例は, 棒に対するハッチングの点が互いに接触していたため, それらの点が小成分として除去できず, 文字として扱われ, 文字列の正しい角度を抽出できず, 誤った角度を画像の傾きとして得てしまった結果である. これを防ぐには, 文字と推定できる構成要素のみを抽出できるように工夫する必要がある.

構成要素抽出処理では, 全要素数の 93.0% を正しく抽出することができた. これは抽出した構成要素全体の 96.1% に相当する. 誤抽出の中には誤って連結された文字列も含まれる. また, 棒に対するハッチングが斜線として描かれていた場合に, 斜線と棒の輪郭線の交点が, 誤って輪郭線の直交点として抽出されることにより, 棒が寸断されてしまうという誤抽出も見られた. 構成要素に対する制約を工夫するか, 抽出アルゴリズムを再検討する必要がある.

構造解析処理では, 全要素数の 71.9% を正しく認識することができた. これは認識対象となったグラフ要素の 80.1% に相当する. この処理では, 構成要素抽出処理において, 誤って 1 つの文字列として抽出された複数の文字列が正しく分割されている (たとえば, 図 30, 図 31 の点線で囲まれた部分). 構造解析処理における誤りの特徴として, 構成要素の抽出処理の失敗による影響を受けやすいことと, 1 つのグラフ要素の認識を誤るとその誤りが他のグラフ要素にも伝搬することがあげられる (図 32 参照). したがって, 構成要素の抽出誤りを減らすことと, 抽出結果を検証的に構造解析処理において照合し, 訂正することが重要となってくる. そのためには, グラフ要素間の論理的相互関係をレイアウト・ネットワークの枝に与え, 照合時にチェックするか, またはルールとして独立に用意し, 検証的に処理する必要がある.

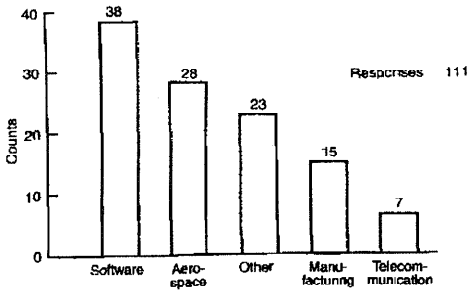


図 28 入力画像の例
Fig. 28 Original image.

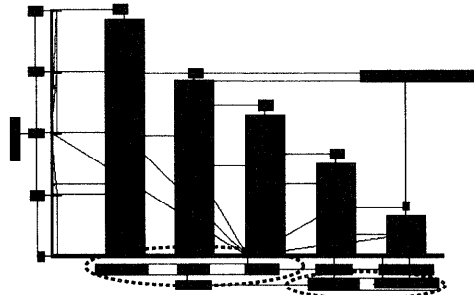


図 31 構造解析処理の結果
Fig. 31 Result in structure analysis.

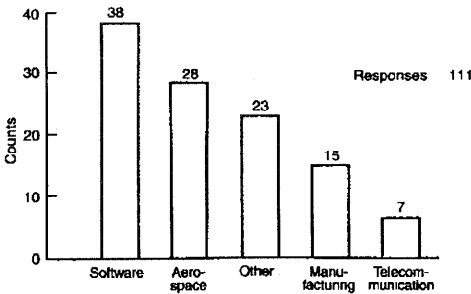


図 29 角度補正処理の結果
Fig. 29 Image modified by angle correction.

表 2 角度補正処理の結果

Table 2 Summary about results in angle correction.

	対象枚数	正補正枚数
角度補正処理	10	9

表 3 構造認識処理の結果

Table 3 Summary about results in structure analysis.

	構成要素抽出処理	構造解析処理
全要素数 (A)	374	374
全抽出/認識数 (B)	362	336
正抽出/認識数 (C)	348	268
(C/A) (%)	93.0	71.9
(C/B) (%)	96.1	80.1

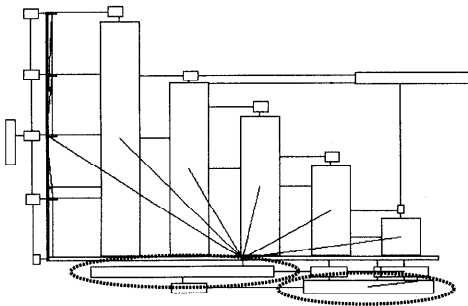


図 30 構成要素抽出処理の結果 (隣接関係ネットワーク)
Fig. 30 Result in extraction of composite elements: neighborhood network.

7. おわりに

本稿では、ビジネス・グラフ、特に棒グラフが独自のレイアウト構造を持つことに着目し、それを知識として利用した棒グラフの構造認識の手法について述べた。棒グラフを構成する様々なグラフ要素は互いに何らかの規則を持って配置され、この規則をレイアウト構造知識としてグラフ画像の構造認識に利用する手法を提案した。そして、レイアウト構造知識はグラフ要素を節、グラフ要素間の位置関係を枝としたネット

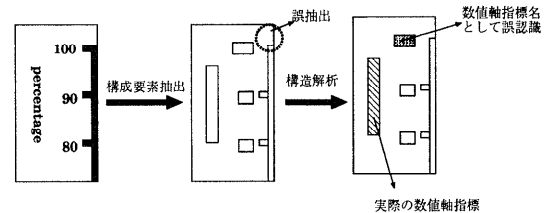


図 32 構造解析処理における誤り
Fig. 32 Errors in structure analysis.

ワークで表現され、それをレイアウト・ネットワークと呼んだ。

本グラフ構造認識処理は構成要素抽出処理と構造解析処理の 2 段階に分けられる。構成要素抽出処理では画像の特徴に従って入力画像内の黒画素を、グラフを構成する要素に分割する。この処理では、さらに抽出された構成要素の位置関係を調べ、隣接関係ネットワークを生成する。一方、構造解析処理では得られた隣接関係ネットワークと、レイアウト構造知識であるレイアウト・ネットワークを比較・照合しながら、グラフ要素および対象グラフの構造を認識する。構成要素抽出処理は、画像の特徴に基づいたボトム・アップ処

理であり、構造解析処理は知識を参照するトップ・ダウン処理である。提案した手法を実装して実験し、実験例は少ないが、構成要素抽出処理の抽出率が93.0%、構造解析処理の認識率が71.9%という結果を得た。

今後の課題として、まず実験結果の考察にあげた問題点の解決がある。角度補正処理と構成要素抽出処理の画像処理では、グラフ内の棒のハッチングに影響を受けた誤りが見られた。様々な棒グラフに対処するためには、棒に対するハッチングとは無関係に処理できる手法を検討しなければならない。構造解析処理では構成要素の抽出失敗の影響を受けやすいことと、1つのグラフ要素の認識誤りが周りに伝搬することが原因で、誤りが発生した。グラフ要素の認識誤りを減らすためには、まず構成要素の抽出誤りを減らすこと、そして誤った抽出結果を構造解析処理において訂正する手法を検討することが重要となる。さらに、得られたグラフの構造を参照した入力グラフ内容理解があげられる。そのためにはグラフ構造から内容を解釈するための知識や手法、さらに内容をどのような形式で表現するかについて検討する必要がある。

謝辞 日々ご教授いただいている中京大学大学院情報科学研究科・福村晃夫教授、本学大学院工学研究科・稲垣康善教授、鳥脇純一郎教授に深く感謝するとともに、討論していただいた研究室の皆様にご感謝いたします。

参 考 文 献

- 1) O'Gorman, L. and Kasturi, R. (Eds.): *Document Image Analysis*, IEEE Computer Society Press (1995).
- 2) O'Gorman, L. and Kasturi, R.: *Image Analysis Systems*, *IEEE Computer*, Vol.25, No.7, pp.5-8 (1992).
- 3) Watanabe, T., Luo, Q. and Sugie, N.: *Structure Recognition Methods for Various Types of Documents*, *Int'l J. of Machine Vision and Applications*, Vol.6, pp.163-176 (1993).
- 4) 渡邊豊英: 図面・文書画像処理技術の現状, 画像ラボ, pp.19-23 (Apr. 1998).
- 5) Watanabe, T.: *Document Analysis and Recognition*, *IEICE Trans. Inf. & Syst.*, Vol.E82-D, No.3, pp.601-610 (1999).
- 6) Futrelle, R.P., et al.: *Understanding Diagrams in Technical Documents*, *IEEE Computer*, Vol.25, No.7, pp.75-78 (1992).
- 7) Lee, M.H., Babaguchi, N. and Kitahashi, T.: *Symbolization and Presentation of Graph Images for Intelligent Communication of Document Images*, *Proc. ACCV'95*, Vol.3, pp.680-684 (1995).
- 8) Yokokura, N. and Watanabe, T.: *Reading Bar Graphs*, *Proc. VSMM'96*, pp.508-512 (1996).
- 9) Yokokura, N. and Watanabe, T.: *Recognition of Composite Elements in Bar Graphs*, *Proc. MVA'96*, pp.348-351 (1996).
- 10) Yokokura, N. and Watanabe, T.: *Layout-based Approach for Extracting Constructive Elements of Bar Graphs*, *Proc. GREC'97*, pp.119-126 (1997). *Graphics Recognition*, Tombre, K. and Chhabra, A.K. (Eds.), *Lecture Notes in Computer Science*, Vol.1389, pp.163-174 (1998).
- 11) Yokokura, N. and Watanabe, T.: *Recognition of Various Bar-graph Structures Based on Layout Model*, *Proc. ACCV'98*, Vol.2, *Lecture Notes in Computer Science*, Vol.1352, pp.113-120 (1998).
- 12) O'Gorman, L.: *The Document Spectrum for Page Layout Analysis*, *IEEE Trans. PAMI*, Vol.PAMI-15, No.11, pp.1162-1173 (1993).

(平成 10 年 10 月 2 日受付)

(平成 11 年 5 月 7 日採録)



横倉 直子 (正会員)

1996 年名古屋大学工学部情報工学科卒業。1998 年同大学院工学研究科情報工学専攻博士前期課程修了。同年日本総合研究所入社、現在に至る。グラフの構造認識の研究に従事。



渡邊 豊英 (正会員)

1948 年生。1972 年京都大学理学部卒業。1974 年同大学院工学研究科数理工学専攻修士課程修了。1975 年同博士課程中途退学。同年京都大学大型計算機センター助手。1987 年名古屋大学工学部情報工学科助教授。現在、同大学院工学研究科情報工学専攻教授。京都大学工学博士。統合化環境、分散協調環境、データベース環境、データベースの高度インタフェース、知的 CAI、文書理解、地図理解に興味を持つ。電子情報通信学会、日本ソフトウェア科学会、人工知能学会、ACM、IEEE Computer Society、AAAI 各会員。