自動認識整形機能をもったペンによる描画システム

†五十嵐 健夫　　‡河内谷 幸子　　†松岡 聡　　†田中 英彦
†東京大学 工学部　　‡東京大学 理学部

既存のペンによる描画システムは、基本的に長方形や楕円を基本とするオブジェクト・ベースであり、ペンの持つ本来の柔軟さを生かすことができていなかった。我々は、ペンによる図形描画と制約に基づく整形を有機的に融合することにより、ペンのもつ自由度を生かし短時間で正確な図形を描くことを可能にする描画システムを開発中である。本稿では、この描画システムの特徴である、ペンによる自然な描画技法と動的な整形結果調整機構を利用した高度な対話的処理、およびその基礎となるセグメントによる図形データ表現と知覚的制約にもとづく認識・整形アルゴリズムについて説明する。

A Pen-Based Constraint Drawing System Combining Dexterity and Precision

†Takeo Igarashi　　‡Sachiko Kawachiya　　†Satoshi Matsuoka　　†Hidehiko Tanaka
†Dept. of Information Engineering, Univ. of Tokyo
‡Dept. of Information Science, Univ. of Tokyo
takeo@mtl.t.u-tokyo.ac.jp

Traditional sketching on computers lacked the freedom of real pens. Our drawing editor combines the dexterity of real pens and computer-assisted precision based on techniques such as two-phased sketch interaction with pie menu and sliders, beautification with perceptual constraints, and segment-based drawing representation. Prototype implementation on IBM pen PC and Xerox Liveboard has shown that the system is fast and easy to use.

1 はじめに

ディスプレイ一体型のタブレットを用いたペン入力は、マウスによる入力に比べて紙とペンによる操作感にちかく、描画システムの入力手段として有効と考えられる [1]。

しかし、単なる手書きのスケッチでなく整形された図を描こうとすると、現在あるのは入力点列を長方形や楕円といったプリミティブに変換するオブジェクト・ベースなシステムが主流であり (GO's PenPoint, the Apple newton, [15][2])、限られたプリミティブにしばられない自由な描画を行なうことは難しい。

このようなシステムはペンのもつ本来の特長を生かしておらず、以下のような問題を持っていると考えられる。

- 上書きや書き足しといったペンによる直感的な操作が許されておらず、描画の際にストレスを感じる。
- 認識に失敗した場合には、基本的に入力から描き直さなくてはならない。
- 長方形や楕円といった用意されたプリミティブからはずれた図形 (星型や菱形、正三角形、L 字型の図形など) を描こうとすると手間がかかり、かつ正確に描けない。

特に最後に挙げた問題は、ペンによる描画に限らず既存の描画システムのもつ大きな問題である。実際にはポリラインや長方形を組み合わせて描くわけであるが、操作の手間が多く、位置合わせなどにも神経をつかうことになる。(位置合わせに関しては、snap dragging などの有効な手法もある。3 節を参照のこと。)

我々は以上のような問題を解消する試みとして、知覚的制約を利用した認識と整形の統合環境を提供し、さらにペンの特徴を考慮した高度な対話性を備えたシステムを開発中である。

図中の制約を推測して整形するシステムは従来から存在する [10] が、我々のシステムはペンによる図形認識と整形を有機的に統合している点、およびバッチ的に処理するのでなくユーザとの対話性を重視したリアルタイムでの整形を行なっている点でおおきく異なっている。

現在プロトタイプ・システムが実装されており、ペン・コンピュータや電子ホワイトボード上で稼働しており、実際に図 1 のような図を短時間で描くことができるようになっている。

本稿では、まず本システムにおけるスムーズで自然な描画について具体的に紹介し、その後、この自然なインタラクションを可能にしている図形の内部表現、および知覚的制約に基づく認識と整形の統合について説明する。

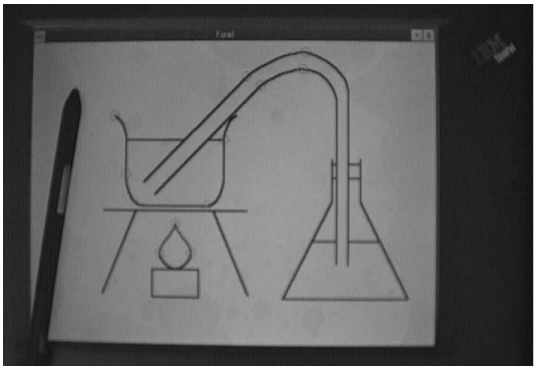

Figure 1: IBM Pen PC 上での描画例

2 本システムにおける描画操作

本節では、本システムの特徴である インタラクティブな描画操作について具体的に説明する。

2.1 ペンによる直感的な描画操作のサポート

人間がペンを用いて図を描くときには、まず 概形を簡単に描き、さらに細部を書き足したり上書きを加えたりすることにより微調整を加えていくことによって図を完成させていく。従来の オブジェクト・ベースの描画エディタでは、上書きは単なる別個の図形として扱われ、描画プロセスの中に有効に組み込まれてこなかった。描くべき形が、類形的でありマウスを使用したシステムであれば、メニューであらかじめ形状を選択しドラッグによって位置合わせをするという方法が適しているとも言えるが、ペンで自由な図形を描こうとする場合にはこのような方法は非直感的で使いにくいものである。

我々のシステムでは、上書きやスクライビングといった直感的な操作をサポートしており、これらをユーザからの重要な意味をもった入力操作とみなして認識システムのなかで適切な処理をおこなうことにより直感的で効率のより操作を実現している。

例えば、図 2 のように (a) の入力に対して (b) のような結果が出た場合、従来のシステムではこのような場合は基本的に描き直しであった。しかし我々のシステムでは、足りない部分を素直に書き足し不要な部分をスクライブして消去する (c) だけで、(d) のように意図する結果を得ることができる。

2.2 動的なメニューによる整形結果の調整

整形システムや認識システムの最大の問題は、意図しない結果が出力されることが多い (入力のもつ曖昧性故に避けられない) こと、およびその場合に結果の修正が困難であること、である。その場合には入力図形を描き

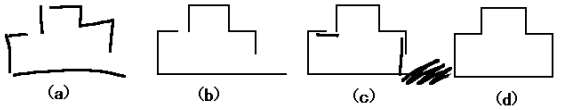

Figure 2: ペンによる直感的な描画操作のサポート

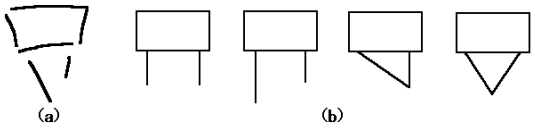

Figure 3: 動的なメニューによる整形結果の調整

直すか、パラメータを明示的に入力して整形し直すことになり、手間がかかる。

我々はこの認識誤りおよび入力の曖昧性の問題について、整形 (認識) 結果の複数の候補から意図していた結果を対話的に選べるようにすることでユーザの負担を大幅に軽減している。

例えば、図 3 で (a) のようなスケッチは、ユーザの嗜好や周囲の図の様子によって (b) のどの候補にもなり得るので、一度の認識では成功せず意図しない結果になる可能性が高い。このような場合に、ユーザはスケッチのすぐそばにメニューを開くことができる。このメニュー上では、ユーザの操作に応じて画面上の表示が次々に変化して複数の候補を見ることができるので、気に入ったところで確定することによって簡単に意図する図形を得ることができる。

候補の提示法としては、システムが自動的に入力図形を総合的に判断して可能性の高い候補から提示していくのが理想であるが、現在の実装では、ユーザが整形に使用される各種のパラメータをスライダーによって直接制御することによって整形結果の調整を実現している。

3 セグメントによる図形情報表現

ペンによる描画システムにおける図形情報の表現として通常使用されるのは

1) ビットマップ表現：入力点列をそのまま保持する表現

2) プリミティブ表現：「円」「長方形」というようなアプリケーション固有のプリミティブの組合せによる表現

のどちらかである。

1) のような形で情報を表現することにより、入力データの変換が不必要になり柔軟性は増す [7] が、見にくいといった視覚上の問題の他に、拡大や縮小・他のデータ形式への変換といった計算機の持つ図形処理能力を適用する際に困難を生じること、携帯端末などの利用を考えた場合に保持すべき情報の量が多いことなどの欠点がある。

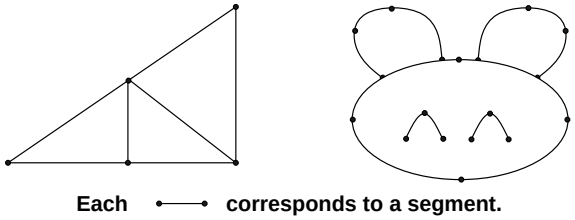

Figure 4: プロトタイプシステムにおけるセグメントの表現

一方、ジェスチャーを認識することにより、オブジェクトに変換する 2) の場合には、整形された図形が使用されるために見やすくまた拡大縮小やコピーや移動などの操作ができるメリットがある。しかし、用意された個別の認識ルーチンに見合うような図形を常に描かなければならないことになり、図的表現（ペン入力）のもつ柔軟性は失われ、またそれらの用意されたプリミティブからはずれた図を描く場合に非常な手間が必要になる。

我々はこの両者の中間に位置する図形表現がペン入力には適すると考える。すなわち、人間の書く図形を「円」や「長方形」のようなアプリケーション固有の限られたプリミティブではなく、線分や曲線部といった知覚的な図形のプリミティブの集合として表現するべきである。これによって、図形の内部表現を意識しない、自然で自由な処理が可能になる。このような考え方に基づくものとして Moran らの研究 [12] がある。彼らのシステムは、スケッチとして入力された図形を内部で自動的に分断して管理することで、実際に人間の目に知覚される図形の構成を表現できるようにしようというものであるが、整形図形を意図したものではない。

セグメントを基本とする表現によって、ビットマップの欠点である情報量の多さや粗さを解決しながら限られたプリミティブに捕らわれない柔軟な操作を実現することができる。具体的に現在の実装では、直線に関しては屈曲点や交点で区切られる各線分、曲線に関しては x,y 方向での極大極小といった一定の特徴を持った点で切った曲線片、を基本的な単位としている (図 4)。これらのセグメントの上に次節で説明する制約を課することによりさまざまな図形表現が実現される。

4 制約に基づく認識と整形

ペンコンピュータにおけるオブジェクトベースのシステムでの認識 [15][2] は、長方形や菱形・円や楕円といっ

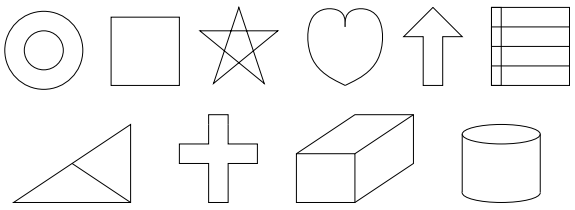

Figure 5: 簡単な制約で表現できるさまざまな図形の例

た限定されたプリミティブへの変換であるため、認識に失敗した場合には基本的に描き直しが必要であり、また用意された図形と違う図形の描画には非常に手間がかかる。

しかし、人間が描こうとする図形のほとんどは、少数の制約を用いることで実現することが可能であると考えられる。たとえば、図5に示したような図形は、対称性や類似部分の合同性、並行や垂直といった知覚的な制約を用いることで一意に定めることが可能である。(制約の働かない部分はユーザの入力データをそのままもちいることで、入力ストロークによって表現されたユーザの意図をそこなわずに望まれる図形を得ることができる。)

我々の方針は、この制約による整形を従来のペンによる描画システムにおける認識の代わりに利用しようというものである。すなわち、オブジェクトベースの描画エディタで入力から特徴ベクトルを計算して長方形や菱形・楕円へと直接変換していたところを、本システムでは、スケッチ入力から特徴をとりだして分節しその上に知覚的に自然な制約による整形をほどこすことによって同様の結果をえようする。例えば、長方形の場合には、各角の角度を直角にする、あるいは各辺を水平あるいは垂直する、という制約が使われ、菱形の場合には 上下左右に対称にするという制約が使われる。

このようにして、一度セグメントに落してから制約による整形をすることにより、限定された図形にしばられずに簡単に必要な図を得ること、およびまた 2.1 節で説明したようなペンによる自然な操作が可能になる。

利用される制約は、図5に示したような個々のオブジェクト内で閉じているものの他に、上下左右位置の一致や類似部分の合同性というような、周囲にすでに描いてある図形から得られる制約がある。この周囲の図形から生じる制約については、新しい図形を描く以前に計算して保持しておくことができるので計算時間を押えることができる。

なお、さまざまな知覚的制約の中でも、特に対称性は通常の認識アルゴリズムではあまり利用されないが、我々のアルゴリズムでは主要な役割を果たすという意味で重要である。対称性を利用することによって、特別な認識ルールに頼ることなく、 長方形や菱形・楕円や同心円といったよく使われるプリミティブを抽出することが可能となる。また、対称性は大局的な判断基準であり、局所的な類似性などにくらべて頑健な処理が行なえる点も重要である。

一般に入力図形は曖昧で認識は一意には決まらず、また速度的問題から一度に同時には解けない制約が生じることが多い。しかし、我々のシステムでは 2 節で説明したように対話に整形を進めていくことができるようになっており、より短い時間で確実に求める解へとたどり着くことが可能である。

5 プロトタイプシステム

現在、我々のプロトタイプシステムは Windows 上 Visual Basic で開発されている (現在高速化のため C++ へ移行中)。本システムは IBM のペンコンピュータ ThinkPad 360PE (図 1) や XEROX 社の電子ホワイトボード である Liveboard 上 (図 7) で稼働する。図形を描いてから整形結果を得るまでの反応時間は、図 2 程度の図で 1 秒以下であり対話的な作業が可能である。

ここでは、現在のプロトタイプシステムで実際に実装されているアルゴリズムについて詳しく説明する。

まず、ユーザは上書きやスクライブを含んだ自由なスケッチを描き、適当なところで整形 (認識) 処理を要求する。システムは描画結果を返しユーザに提示するが、ユーザがその結果に納得しない場合には、メニューによってパラメータを変更し再び整形をおこない別の候補を得るか、一旦確定して図のようにペンで書き足す、あるいは入力を取消して描き直す (2 節を参照)。

整形ルーチンでは、まず入力点列を屈曲点で区切る (実際にはペンの速度の最小値を検出し、これを屈曲点とみなす)。曲線と思われる場合には 3 節で説明したような特徴点で切断してセグメントとする。上書きされた部分古い部分やスクライブで消去された部分セグメントを取り除いたのち、実際の制約の抽出と整形に入る。この部分は現在の実装では、近い点同士を結び付けたり、対称と思われる場合に完全な対称にするといった、各制約を抽出し充足させる複数のフィルターの中を入力データが通過する形式になっており、まだ抽出された制約を総合的に統合して結果を返すようにはなっていないため不完全な結果になる可能性がある。しかし実際には、結果が即座に帰ってくるので、複数回対話的に整形操作を行なうことにより望みの形状を短時間で得ることが可能である。

現在実際に抽出し処理を行なっている制約としては、セグメントの端点同士および端点と他のセグメントとを完全につなげる制約、曲線部と直線部の接点における滑らかさを保証する制約、水平垂直方向に近い直線の傾き

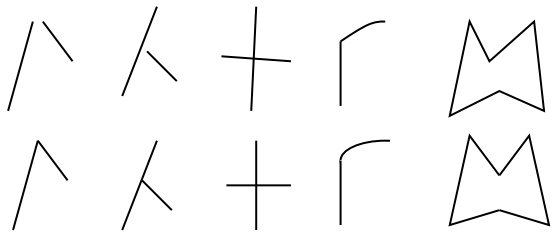

Figure 6: 現在実装されている制約と整形処理

を完全にする制約およびセグメント集合のもつ対称性などがある (図 6)。近い部分をくっつける場合には、ある閾値 (メニューで調整可能) 以下の距離にある点のペアや点とセグメントを検出して座標を一致させる。傾きについても、水平垂直からのずれが閾値以下のセグメントを検出してずれがなくなるように座標を変更する。対称性に関しては、入力図形の重心を求めた後、各セグメントをその重心をとおる軸 (中心軸) に対して反転させて、他のセグメントとある程度以上重なるかを順番にしらべていき、対応するセグメントがそれぞれ 1 対 1 にみつかった場合に対称と判断して、各セグメントの対に対して求めた中心軸が中心にくるように座標を変更する。まだ実装されていないセグメントの位置関係および形状の類似性、直線間の並行性、等角度や等間隔といった制約についても将来実装する予定である。

メニューでは、整形結果の確定、削除、作業の取消 (undo)、整形無し、整形結果の微調整などを選ぶことができ、微調整を行なうメニューでは、対称性を明示的に指定したり、近い点を結びつけたり水平垂直に近い直線の傾きを完全にしたりする場合のずれの許容範囲を調整したりすることができる。

6 関連研究

計算機上での図形処理に制約を利用する試みは数多くみられる。[14] [5] [9] [6] などシステムでは制約を明示的に与えることにより要素図形 (オブジェクト) を移動した場合にその制約を維持するような機構や、宣言的に指定した制約を与えることにより複雑な図形を計算して描く機能などが実現されている。しかし、これらのシステムは制約の指定に手間がかかるために、時間をかけて複雑な図を描く場合には適しているが、ペンを利用した簡単な描画作業には向いていない。

Snap dragging[3] や Aldus IntelliDraw[8] などは、図形のなかからヒューリスティクスによって制約を抽出し、描画操作を助ける機能がある。たとえば、長方形をドラッグするときに、別の長方形と水平あるいは垂直方向に近い位置にくると、その位置で長方形が一時的にとまり (ひっかかり)、正確な図形を描くことを助けてくれる。これらの機能は、ユーザが明示的に制約を指定する必要がなく、システムが積極的に正確な図になるように支援するという点で我々の考え方と共通するものである。我々のシステムは、ダイレクトマニピュレーションの範疇で利用されていたこれらの機能をスケッチによる描画システムに応用したものといえる。

図形の自動整形に関しても研究が行なわれているが [10][4]、これらの研究では、描画のフェーズと整形のフェーズが完全にわかれてしまっている点で我々のシステムと異なっている。[10] では、既存図形からの制約の抽出と制約解消による整形について考察がなされているが、あくまでもバッチ的に処理することを前提としており対話的な処理については言及されていない。一方、[10] は整形に使われるパラメータを調整するインターフェースを備えており整形結果を自由に修正することが可能となっているが、オブジェクトベースでありまた非常に低レベルな整形 (接触すべき点や直線の間のずれ、水平垂直方向からの微小な傾きの解消) しか扱っておらずペンによる図の描画には適していない。

ペンによる入力に関しては、ほとんどの研究・システムが入力からの特徴ベクトルの抽出とパターンマッチングによる用意されたオブジェクト要素への変換を行なっている。描画エディタにおけるジェスチャーの利用に関しては、[11] のような研究があるが、ジェスチャーの利用に関しては、ジェスチャーを覚えるのが大変であること、通常の紙の上での描画作業と大きく異なるために不自然な操作感になることなどの問題がある。Apple 社の Newton や [15][2] のシステムなどは、ジェスチャーというよりも描きたい形状を描くとそのまま整形して表示してくれる点でより自然であるといえる。しかし、結果として出力されるものは制限されたカテゴリに属するもののみであり、フローチャートのような決まった要素からなる図の描画には適するものの、1 章で挙げたような理由から 我々の目指す一般的な図形描画のためのシステムとしては不都合が多い。

7 まとめ

ペンによる入力データを、自動的に抽出される知覚的制約によって整形する描画システムについて概説した。このシステムは 高度な対話性を提供することよって、従来のプリミティブの認識方式のペンによる描画エディタのもつ限界を克服した、より柔軟かつ強力な図形描画を可能にしている。

本稿で提案したようなシステムは、素早く正確な図を描くことが可能な点が最大の特徴であり、携帯用のペンコンピュータで簡単な図をメモしたり (小さい計算機で

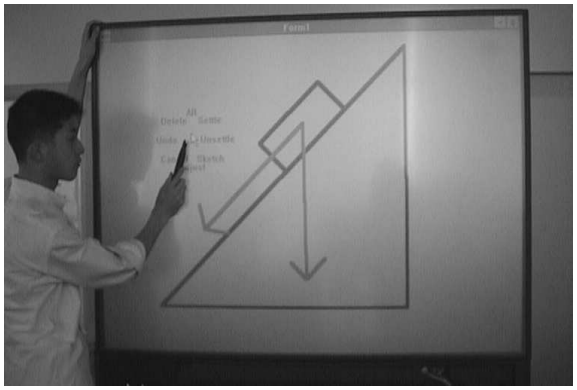

Figure 7: XEROX Liveboard での描画例

は整形を行なわないとドットの粗さが目につくために見にくく、また限られた記憶容量を無駄にするという点で整形が要求される)、電子的なホワイトボードを利用したプレゼンテーションや講義の最中にすばやくきれいな図を描いたりする時に特に有効であると考えられる。特に後者の場合、既存の描画システムでは描画に手間がかかりすぎるために主に手書きそのままの図が用いられてきたが、整形された図の方が考えをはっきり示すことができる場合もおおいはずであり、また入力される図形も単純であるため、我々の手法が効果を発揮するものと期待される。

状態遷移図やフローチャートのように決まった要素のみから構成される図の描画では、従来のメニューとダイレクトマニピュレーションを利用したマウスによる描画手法が有効であると考えられるが、自由な曲線の含まれる微妙な表現やアイデアからまとめていくような作業にはマウスよりも操作性の高いペンによる描画が有効であると考えられるので [13]、今後より複雑な図面への本手法の適用なども検討していきたい。

今後の課題としては、大局的な類似性といった高度な知覚的特徴を扱えるようにする認識アルゴリズムの改良、困難ではあるがより大きな可能性を持つ曲線を適切に扱える手法の開発、地図エディタのようなアプリケーションへの組み込み、およびユーザスタディによる我々の手法の有効性の評価などがあげられる。

謝辞

本研究の実装にあたり、XEROX 社の Liveboard を貸してくださった慶応大学環境情報学部の徳田英幸教授に感謝する。

References

[1] Apte,A., Kimura.D., "A Comparison Study of the Pen and the Mouse in Editing Graphic Diagrams", *Proc. of Visual Languages '93*, pp. 352-357, 1993.

[2] Apte, A., Vo, V., and Kimura, T.D., "Recognizing Multistroke Geometric Shapes: An Experimental Evaluation," *Proc. of UIST'93*, pp. 121 - 128. 1993.

[3] Bier,E.A., Stone,M.C., "Snap Dragging", Computer Graphics, Vol.20, No.4, pp. 233-240, 1986.

[4] Bolz, D., "Some Aspects of the User Interface of a Knowledge Based Beautifier for Drawings", Proc. of 1993 Int'l Workshop on Intelligent User Interfaces, Gray,W.D., Hefley,W.E., Murray,D., Eds. , ACM Press, New York, 1993.

[5] Borning,A., "The Programming Language Aspects of ThingLab, A constraint-Oriented Simulation Laboratory", ACM Trans, on Program. Lang. Syst., Vol.3, No.4, pp.353-387. 1981.

[6] Bouma,W., Fudos,I., Hoffman.D., Cai,J., Paige,R., "Geometric constraint solver", Computer Aided Design, Vol.27, No.6, pp. 487-501, 1995.

[7] Lopresti,D., Tomkins,A., "Computing in the Ink Domain" , Symbiosis of Human and Artifact *Proc. of HCI'95* , Vol.1, pp. 543-548, 1995

[8] Myers,B.A., Wolf,R., Potosnak,K., Graham,C., "Huristics in Real User Interfaces", INTERCHI'93 Panel, *Proc. of InterCHI'93*, pp.304-307, 1993.

[9] Nelson,G., "Juno, A Constraint-based Graphics System", Computer Graphics, Vol.19, No3, pp. 235-243, 1985.

[10] Pavlidis,T., VanWyk,C.J., "An Aotomatic Beautifier for Drawings and Illustrations", Computer Graphics, Vol.19, No.3, Proc. of SIGGRAPH '85 , pp. 225-234, 1985.

[11] Rubine,D., "Combining Gestures and Direct Manipulation", *Proc. of CHI'92*, pp.659-660, 1992.

[12] Saund,E., Moran,T.P, "A Perceptually Supported Sketch Editor", *Proc. of UIST'94*, pp. 175-184, 1994.

[13] Sugishita.S., Kondo.K., Sato.H., Shimada.S., Kimura.F., "Interactive Freehand Sketch Interpreter for Geometric Modelling", Symbiosis of Human and Artifact *Proc. of HCI'95*, Vol.1, pp. 543-548, 1995

[14] Sutherland,I.E., "Sketchpad: A Man-Machine Graphical Communication System", *Proc. of Spring Jint Computer Conf.*, No.23, pp.329-346, 1963.

[15] Zao,R., "Incremental Recognition in Gesture-Based and Sytax-Directed Diagram Editors", *Proc. of InterCHI'93*, pp. 95-100, 1993.