対話的整形による幾何学的図形の高速描画

五十嵐 健夫[†] 松岡 聡[†††]
河内谷 幸子[††] 田中 英彦[†]

本論文では，幾何学的な図形を計算機上で手早く描くことを可能にする対話的整形という描画手法を提案し，そのアルゴリズムおよびプロトタイプシステムを用いた評価実験の結果について述べる．既存の描画システムでは，平行や対称といった性質を満たす幾何学的図形を描画するために回転や複製といった複雑な編集操作を適切に組み合わせて用いなくてはならず，特に初心者にとって短時間に正確な描画を行なうことは困難である．対話的整形とは，計算機がユーザの手書き入力を受けとり必要な幾何学的制約を自動的に推測して整形を行なうものであり，編集操作を一切使用することなく幾何学的制約を満たすことを可能にする．対話的整形では，図形全体でなくストローク一本一本に対して整形を行なうことで意図と大幅に異なる変形を防ぎ，もっともらしい整形結果の候補を複数生成してユーザに提示することで入力の曖昧性に対処している．アルゴリズムは，手書きの入力図形から必要な幾何学的制約を推測して抽出する制約抽出部と，抽出された幾何学的制約から整形図形を生成する制約解消系からなっており，実時間での効率的な動作を実現している．直線のみからなる図形を描くことのできるプロトタイプシステムが実装されており，描画実験により，既存の描画システムに比べて描画時間が短縮され，かつ図形の幾何学的制約の充足度も改善されていることを確認した．

Interactive Beautification: A Technique for Rapid Geometric Design

Takeo Igarashi,[†] Satoshi Matsuoka,[†††] Sachiko Kawachiya[††] and Hidehiko Tanaka[†]

Diagram drawing with conventional computer-assisted drawing editors often tend to take considerable amount of time despite their seeming ease of use. The causes of the problem are too many commands and unintuitive procedures to satisfy geometric constraints. To solve the problem, we propose *interactive beautification*, a technique for rapid geometric design, and developed a prototype system Pegasus to verify the efficiency of the technique. Interactive beautification system receives the user's freestroke and beautifies it considering geometric constraints among segments. Using the technique, the user can draw precise diagrams with geometric relations rapidly without using any editing commands. Current prototype system supports drawings comprised of straight lines, and a user study was performed using the prototype system, a commercial CAD, and an OO-based drawing system. The result showed that the users can draw required diagrams more *rapidly* and more *precisely* using the prototype system.

1. はじめに

MacDraw や CAD システムのような市販のオブジェクトベースの描画エディタは，複製や反転といった種々の編集コマンドやグリッドに代表される特別な描画モードを持っており，このような編集コマンドを適当に組み合わせて使用することにより幾何学的な制約を満たした図形を描くことが可能である．例えば，図形を複製し反転・位置調整を行なうことで正確な対称図形を描くことができ，斜めの線分を複製し 90 度の回転を行なうことで元の線分に直角な線分を描くことができる．また，多くの CAD システムには指定した線分に垂直な線分を描くための描画モードなどが備わっている．しかし，このような描画コマンドや描画モードを呼び出す作業は手書きでの描画と比較すると余分な手間であり，また適切なコマンドや描画モードを選択することは特に初心者にとって非常に困難である[10]．

このような問題を解消するものとして，本論文では

† 東京大学工学系研究科
Graduate School of Engineering, University of Tokyo
†† 東京大学理学系研究科
Graduate School of Science, University of Tokyo
††† 東京工業大学情報理工学研究科
Graduate School of Information Science and Engineering,

「対話的整形」と呼ぶ新しい描画手法を提案する．対話的整形とは，**図 1** のような幾何学的制約を満たした図を，編集コマンドや描画モードを一切呼び出すことなく，手早くかつ正確に描くためのインタラクション手法である．対話的整形システムは，ユーザの描いた手書きの入力ストロークからそのストロークの満たすべき幾何学的制約を自動抽出し，その幾何学的制約を満たすように整形を行なうもので，手書きによる入力ストロークのベクトル化[5] および自動図形整形システム[16] を拡張したものとみなすことができる．本手法を利用することによって，意図する図形を直接フリーストロークで描くという非常に直感的な操作によって，初心者でも正確な図形を特別な訓練を必要とすることなく短時間で描くことが可能となる．

対話的整形は以下の 3 つの点によって特徴付けられる．1) 図形全体でなくストローク毎に行われる整形処理，2) 複数の複雑な幾何学的制約の自動抽出と自動充足，3) 整形結果として複数の候補を出力し選択させる機構．これらの 3 つの機能が協調して動作することで手書き入力に特有の曖昧性の問題が解消され，直感的で自然な描画が可能になる．

現在，対話的整形機構はペガサスと呼ばれるプロトタイプ描画システムとして実装されており，このシステムを使用した実験より本手法による描画効率の向上が確認されている．本論文では，対話的整形という描画手法の内容とそれを実現するアルゴリズム，および実装されているプロトタイプシステムを紹介する．

次節では，計算機上での図形描画に関する関連研究について述べる．次に，ユーザの側から見た対話的整形の動作について幾つかの描画例を使用して説明を行ない，ついでそのような動作を実現するアルゴリズムおよび実装されているプロトタイプシステムについて説明する．さらにプロトタイプシステムを使用して行なった本手法の有効性を確認する実験の内容と結果について述べた後，現在の実装の限界と今後の課題について考察を加え，最後にまとめを述べる．

2. 関 連 研 究

計算機上で図形の描画を行なうことは古くから行なわれており，描画を効率的に行なうための手法もこれまでに数多く提案されてきている．ここでは，対話的整形に関連の深い重要な描画手法について，その特徴と本手法との違いを明確にする．

まず，今回提案する手法に非常に近い描画手法として，市販のペンコンピュータ (Apple 社の Newton や Go 社の PenPoint) に実装されているペンによる手書き図形入力の整形機構や，Smart Sketch 等の通常の描画エディタ中に見られるフリーストロークの整形モードなどがある．これらのシステムは手書きの入力を受けとった後，端点の接続および垂直化や水平化といった基本的な整形処理を行ないベクトル化する．対話的整形はこの機能を大幅に拡張したものであり，平行や対称，合同といった複雑な幾何学的制約を扱うことによってより広範な要求に応えていこうとするものである．また，本論文で提案するような複数の整形結果の候補の生成機能は，これまで見られなかったものである．

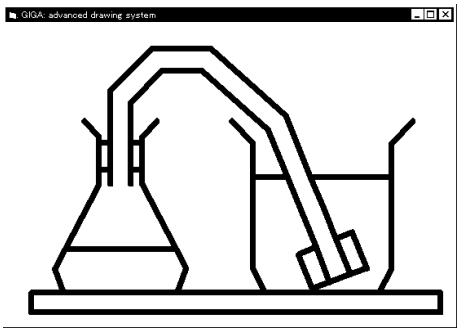


図 1 プロトタイプシステム (Pegasus) を使用した描画例
Fig. 1 A diagram drawn on the prototype system Pegasus

ジェスチャーによるシステム[1)20)17)14)] もまた手書き入力を扱うが，これらのシステムは入力されたストロークの形状とテンプレートとして記憶しておいたストロークの形状を比較してもっとも近いものを独立した記号として返すものであり，周囲の図形との幾何学的制約を満たすように整形を行なう対話的整形とは異なるものである．Gross らは 文献 7) において，ジェスチャーを認識する際の手書き入力の曖昧性を取り除くためには周囲の状況を考慮にいれた処理を行なうことが重要であると指摘しており，これは対話的整形の設計に関して重要な示唆を与えている．

計算機による図形の自動的な整形処理として幾つかの研究例が認められる[16)12)] が，これらは主に既に描き終った図形の全体を受け取り一括処理として整形を行なうものであり，ユーザの入力のもつ曖昧性や多義性の影響により結果として意図しない図形が返される可能性が大きいといえる．対話的整形は，一ストローク毎に整形を行ない複数候補を提示して確認を求めることで，このような不具合を防いでいる．

対話的整形が入力ストロークの始点と終点という二つの点の位置の計算を同時に行うのに対し，多くの描画システムはマウスポインタの動きを制御することによって一点の位置の指定を支援している．グリッドはポインタの移動を特定の位置 (通常は一定間隔をもった格子) に

制限し，重力関数がポインタを一定の特徴を持った位置(線分や交点など)へと自動的に誘導する．例えば，アドビ社の Intelidraw エディタ[15]は画面上に存在するオブジェクトと縦や横方向の位置がそろうようにマウスポインタを誘導する．このような方式に対して対話的整形には以下のような利点がある．1) 手書き入力によって意図する線分の概形を描く操作は，ポインタの位置を注意深く指定する操作に比べ，特にマウスでなくペンを利用した描画の場合非常に親しみやすく自然である．2) マウスの位置という一点だけからよりも，フリーストロークの始点と終点という二点からの方が，より多くの情報を取り出すことができ，より高度な処理が可能となる．例えば平行線分の幅の制御などは，マウスポインタの動作の操作からでは困難であるが，フリーストロークを整形線分に置き換える操作では自然に実現することができる．

Bier の提案した Snap Dragging[2] は，マウスの動きを制御して様々な幾何学的制約を満たす図形を簡便に描けるようにしたもので，対話的整形と同様の目的をもっているが，Snap Dragging ではユーザはどのような動作を行なうべきなのか事前にメニューによって選択しなければならず，負荷が大きいと考えられる．Saund らの研究[18] は，手書き図形からユーザが自然に知覚する構造を計算機にも認識させるというもので，本手法と似た考え方に基づいているが幾何学的制約を用いた整形処理までは考慮されていない．

制約を明示的に指定することによって正確な描画を行なうシステム[8)3)19)4)] は，入り組んだ幾何学的制約を正確に満たす必要がある場合に適したシステムであるが，必要な制約を正確に指定することは非常に手間のかかる作業であり，比較的簡単な幾何学的制約を持った図形の描画には適さないといえる．

3. 対話的整形

対話的整形は基本的には手書き入力のベクトル化を行なう方式であり，手書きの入力を受け取り幾何学的制約を満たすようなベクトル表現の線分に変換する．始めに，ユーザはペンあるいはマウスを使用して目的とする図形(線分)の概形を直接入力する(**図 2**a)．次に，システムはこの入力ストロークと周囲の図形との位置関係を調べることによって，ユーザが意図したと考えられる幾何学的制約を自動的に推測する(図 2b)．最後に，推測された幾何学的制約を連立方程式として解くことにより，最終的なベクトル表現の線分の始点と終点の座標を計算し，ユーザに結果を提示する(図 2c)．また，手書き入力の持つ本質的な曖昧性に対処するために複数の候補を生成し提示する(図 2d)．

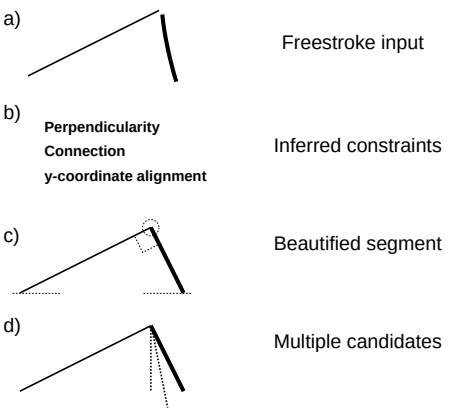


図 2 対話的整形システムの基本動作
Fig. 2 Basic operation of interactive beautification

対話的整形の特徴は大きく，ストローク毎の整形を行なうことにより合同や平行あるいは対称といった複雑な幾何学的制約を満たす図形を生成する機構(1)と，曖昧性の問題を解消するために複数の候補を生成しユーザに提示する機構(2)に分けることができる．以下では，このそれぞれについて例を用いて詳しく説明する．

3.1 幾何学的制約を充足するストローク毎の整形

ここでは，ストローク毎の整形処理によって幾何学的制約を満たす図形が描かれる過程を説明する．実際のシステムでは既に述べたように整形結果として複数候補が生成されるのであるが，この項では簡単のため整形処理の結果としてただひとつの結果しか返さないものとして説明を行なう．複数候補の生成については次の項で詳しく述べる．

図 3 に，現在実装されている幾何学的制約および入力ストロークと整形結果の例の一覧を示す．図 3a,b は，端点接続の例を説明している．ユーザの描いた手書き入力線分の端点が既に描かれている線分あるいはその端点の付近にあった場合に，システムは自動的にその近接性を認識し，整形結果として対応する線分に正確に継った線分を返す．図 3c,d は平行および垂直制約の例を示している．システムは入力ストロークの傾きと既に描かれている全線分の傾きとの比較を行ない，ほぼ同じ傾きの線分が存在する場合には正確に平行な線分を生成し，またほぼ垂直なものがあれば正確に垂直な線分を生成する．図 3e は水平および鉛直方向の位置揃えの例を示している．フリーストロークが入力されると，システムはそれぞれの端点の x 座標および y 座標を，画面上の図形の端点の座標と比較し近いものがあればそれらと一致するように整形を行なう．

図 3f,g は，合同線分や対称線分の描画例である．シ

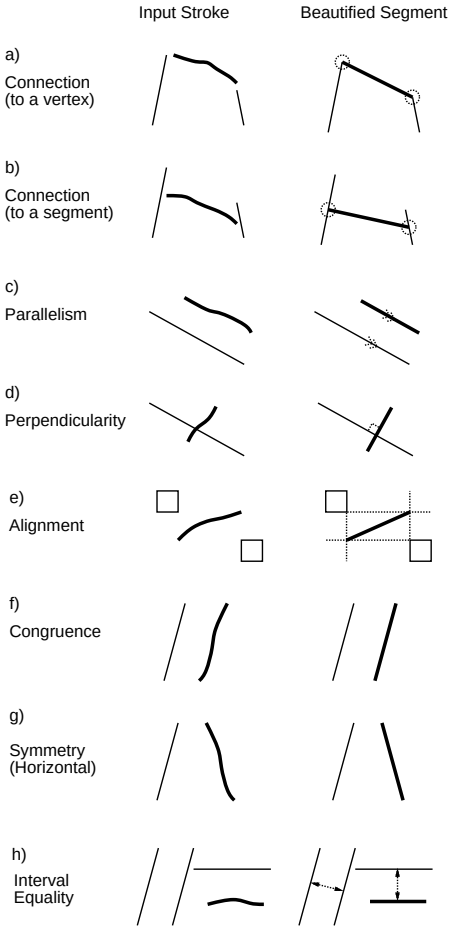


図 3 現在実装されている幾何学的制約
Fig. 3 Supported geometric relations

ステムは，入力ストロークと画面上のすべての線分の形状 (傾きと長さ) の比較を行ない，ほぼ同じであった場合には整形結果として正確に合同な線分を返す．入力ストロークを水平および垂直方向に反転した成分についても同様に類似線分の検索を行ない，反転したものとほぼ合同な線分が画面上にあれば整形結果としてこれらと正確に対称な線分を生成する．

図 3h は，平行線分間の幅の一致の例を示している．このような関係は，入力ストロークとそれにほぼ平行な画面上の線分との間の距離を，既に画面上に存在している平行線分の間の距離と比較することによって検出される．この機能によって，**図 4** に示すような一定の幅をもったパイプのような図や縦横の等間隔でならぶ格子状の図を簡単に描くことが可能となる．既存の描画システム上での幅の等しい平行線分の描画では，あらかじめ描いた平行線分を複製して回転および位置合わせといった操作を行なうことが必要とされ，非常に非直感的で困難

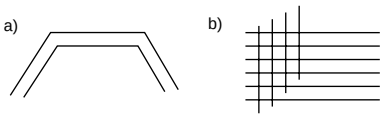


図 4 平行線分間の幅の一致を利用した描画例
Fig. 4 Example use of interval equality among segments

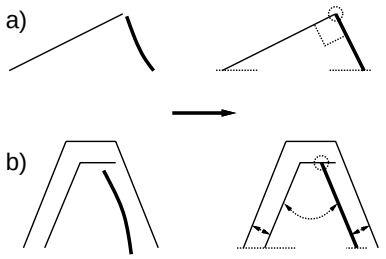


図 5 複数の幾何学的制約を同時に満たす整形の例
Fig. 5 Construction of a diagram with many constraints

であった．

実際の描画では，以上で述べたような複数種類の幾何学的制約が同時に認識され充足されることで，目的とする図形が生成される．**図 5**a では，端点の接続や線分の角度の垂直関係，および水平方向の位置揃えといった幾何学的制約が同時に満たされている．一方，図 5b では，平行線分の幅の一致と水平方向の位置揃え，および反転図形の合同性 (対称性) といった幾何学的制約を使用することにより左右対称なアーチ型の図が描かれている．(整形の結果として生じる不要な線分は，後で説明するような消去ジェスチャーによって簡単に取り除くことができる).

3.2 複数候補の自動生成と選択

手書き入力インタフェースの持つ本質的な問題として，手書き入力の曖昧さあるいは不正確さを挙げることができる．ユーザは何らかの図形形状を頭の中に描いており，それを手書き入力の形でシステムに伝える．システムはその手書き入力から元の図形形状を推測し再構成するのであるが，曖昧で不正確な手書き入力から正確にユーザの意図を推測することは一般に非常に困難である．例えば，**図 6**a に示すような入力が与えられた場合，同図 b にあるような複数の可能性のうちのどれがユーザの意図するものであるのかを言い当てることは難しい．通常の手書き入力による描画システムは，このような複数の候補を考慮せずに単一の結果のみを生成する．もし，その変換結果がユーザの意図と異なっていた場合には，その入力を取消してもう一度書き直さなければならない．

この問題に対する解決法として，対話的整形では，もっともらしいと推測されるすべての整形結果を複数候

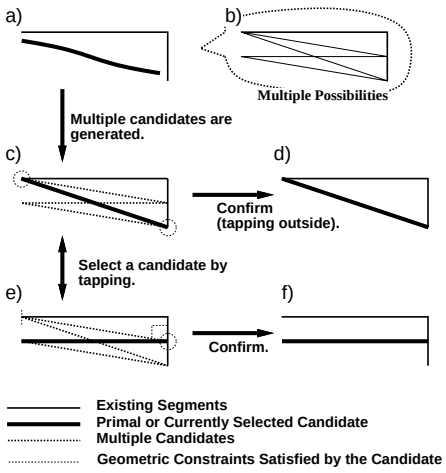


図 6 複数候補の生成と選択
Fig. 6 Interaction with multiple candidates

補として生成して提示し，ユーザに気に入ったものを選択させる (図 6c)．もしシステムが提示した第一候補が気に入らない場合には，直接ペンでタップ (クリック) することによって別の候補を選ぶことができる (図 6e)．複数候補からの選択を行なっている際には，選択されている候補が満たしている幾何学的制約が画面上に明示されるため，ユーザは確実に必要とする制約を満たした図を得ることができる．候補の選択は，候補以外の領域をタップするか次の手書き入力ストロークを描いた時点で終了し，その時点で選択されていた候補以外の候補は消去される (図 6d,f)．

複数候補の自動生成と選択，および充足された幾何学的制約の視覚的な提示機構によって，手書き入力から確実に必要な図形を再現することが可能になる．その結果，従来の手書き整形システムでは非常に困難であった図 1 のようなある程度複雑な幾何学的図形の短時間での描画が実現されている．また，システムの提示する第一候補が正しい場合には一切余分な操作を行なうことなく次の線分の描画に移ることができるため，複数候補の生成によって生じる手間は最小限に抑えられているといえる．

3.3 補助的なインタフェースについて

手書き入力による描画とタップによる候補選択に加えて，現在の実装では基本的な機能を実現する移動式のメニューと消去用ジェスチャーを利用している．移動式メニューは画面上におかれたボタンで，ユーザはその位置を自由に変更することができる．ボタンを押すとパイメニュー[9] と同様にして操作メニューが周囲に表示される．現在は，全画面消去 (Clear) と前操作の取消 (Undo) 操作のみが実装されている．

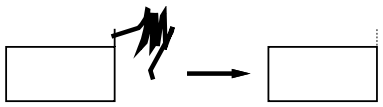


図 7 消去用ジェスチャーとトリミング操作
Fig. 7 Erasing gesture and trimming operation

消去用ジェスチャーは，一定回数以上の往復を伴った手書きストロークとして実装されている．ペン先がタッチパネルから離れた時点で通常の入力ストロークか消去ジェスチャーかの判断を行ない，消去ジェスチャーであった場合にストロークの始点にもっとも近い線分を消去する．画面上の線分は交点および接点であらかじめ区切られているため，非常に簡単な操作で不要なごみのトリミングを行なうことができる (図 7)．一般的にトリミング操作は幾何学的図形を描く際に頻繁に実行される操作であり，この例で示されたような簡単なトリミング機構は複雑な幾何学的図形の効率的な描画の実現に大きく貢献する．

4. 対話的整形のアルゴリズム

ここでは対話的整形のアルゴリズムを詳しく説明する．システムを実装する立場から見た場合，対話的整形システムは以下のように動作する (図 8)．

1) ストロークによる描画が終了しペンがタブレットから離れた時点で，システムはまずそのストロークが通常の入力なのか消去ジェスチャーなのかの分類を行なう．具体的には，フリーストロークの全体の長さを始点から終点までの距離で割ったものが，ある閾値を越えていた場合に消去ジェスチャーとする．

2) 消去ジェスチャーでないと判断された場合に，図形整形ルーチンが呼ばれる．図形整形ルーチンは，手書きのストロークと画面上の全線分の位置情報を入力として受けとり，複数候補を整形結果としてインタフェース部へ返す．インタフェース部は整形結果を画面上に提示し，ユーザからの指示を待つ．

3) 複数候補からの選択が終了した時点，すなわち候補以外の領域がタップされたか次のストロークが描かれた時点で候補図形の確定処理が行なわれる．確定処理により，第一候補あるいは選択された候補が最終的な整形結果として採用され，それ以外の候補は捨てられる．

4) 入力ストロークが消去ジェスチャーであると判断された場合，システムは消去すべき線分を見つけこれを画面上から取り除く．確定処理を行なう部分では，整形処理の負担を軽くするために画面上の線分集合に対して端点座標のソーティングといった基本的な計算処理が行なわれる．

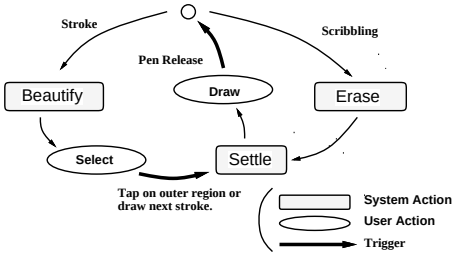


図 8 対話的整形システムの動作モデル
Fig. 8 Operational model of interactive beautification

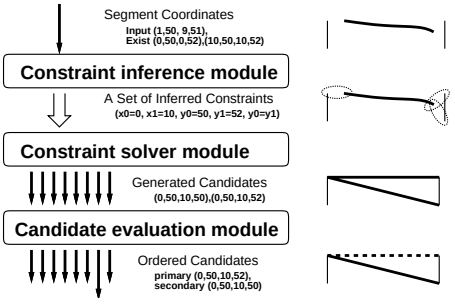


図 9 整形処理部の処理の流れ
Fig. 9 Sturcture of the beautification routine

以下では，整形処理部の内部について詳しく述べる．整形処理は，3 つの処理モジュールが入力データに対して順番に処理を行なうことで実現される (図 9)．まず，制約抽出モジュールにおいて，入力ストロークと画面上の図形との位置関係を調べることにより満たされるべき幾何学的制約が抽出される．次に，制約解消モジュールにおいて，抽出された幾何学的制約を解くことによって複数候補が生成される．最後に，評価モジュールで生成された各候補のもっともらしさが評価され，もっとも確信度の高い候補が第一候補として選択される．制約抽出処理と制約解消処理を分離することによって，各画面上の線分と入力ストロークとの位置関係を調べるというもっとも時間のかかる処理を，各候補毎に行なうことなく一度で済ますことが可能になり，処理時間の大幅な短縮が実現される．

評価モジュールでは，制約解消系での計算結果を元にして，より入力ストロークに近い位置にある候補に高い確信度を与える．制約を抽出する時点で各制約にそのもっともらしさに応じて確信度を与えることも考えられるが，確信度の高い幾何学的制約を組み合わせて計算を行なった結果，もとの入力ストロークからかけ離れた候補が得られる可能性があるため，最終的な整形図形の位置から確信度を求める必要がある．

抽出される幾何学的制約は，整形の結果として生成される線分の端点の座標に相当する四つの変数を拘束する等式の集合として表現され，制約解消系ではこれらを連立方程式として解くことで整形処理を実現している．図 10 に，現在実装されている幾何学的制約と対応する等式の一覧を示す．

Geometric Relations	Corresponding Equalities
Connection (start point on a vertex)	x0 = const y0 = const
Connection (end point on a vertex)	x1 = const y1 = const
Connection (start point on a line)	y0 = const * x0 + const
Connection (end point on a line)	y1 = const * x1 + const
Alignment (start -x)	x0 = const
Alignment (start -y)	y0 = const
Alignment (end -x)	x1 = const
Alignment (end -y)	y1 = const
Vertical line	x0 = x1
Horizontal line	y0 = y1
Congruence (Symmetry)	x1 - x0 = cosnt y1 - y0 = const
Parallelism (Perpendicularity)	y1 - y0 = const * (x1 - x0)
Interval equality	y0 = const * x0 + const y1 = const * x1 + const

図 10 幾何学的制約と等式の対応表
Fig. 10 Relation between geometric relations and equalities

4.1 制約抽出モジュール

始めに，線分の傾きと端点の座標について，入力ストロークの値に近いものを画面上の全線分の値をまとめておいた表から探す．近い値があった場合に入力ストロークの値を対応する画面上の線分の値と等しくするような制約を生成する．この処理によって，水平および垂直方向の位置揃え，平行および垂直といった幾何学的制約が抽出される．端点の x,y 座標および傾きは予めソートしてあるので，この部分の処理の計算量は $O(\log n)$ (n は画面上の線分の数) である．垂直制約は，画面上の線分の傾きを記録する際に，元の傾きに加えて 90 度回転したものを記録しておくことで発見される．

次に，画面上の各線分と入力ストロークとを比較し，端点の接続性や合同性および対称性といった幾何学的制約について調べる．さらに，入力ストロークとほぼ平行な線分が画面上にあった場合には，予め記録してあった既存の平行線分対の間隔と比較し，必要に応じて平行線分間の幅の一致制約を生成する．平行線分間隔は予めソートしてあるため，この部分の処理の計算量は $O(n \log n)$ である．

以上のような 2 段階の制約抽出処理によって，入力ストロークの満たすべき幾何学的制約の集合が生成される．制約解消系での処理の無駄を防ぐため，制約が新しく生成される度に，既に同一の制約が生成されていない

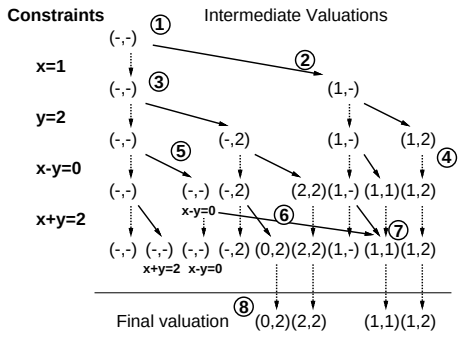


図 11 制約解消アルゴリズム
Fig. 11 Algorithm for constraint solving

かのチェックを行なう.

4.2 制約解消モジュール

ここでは，制約抽出部で生成された幾何学的制約を解くことによって，整形結果の候補の位置 (端点の座標) を計算する．生成される制約は基本的に過制約 (over-constrained) の状態にある☆ので，制約解消モジュールでは互いに矛盾しない制約を組み合わることにより複数の候補を生成する.

実装されている制約解消系は CLP($\Re$)[11] の等式制約解消系を，過制約状態から複数解を生成するように拡張したものである．CLP($\Re$) と同様に，始めはすべての変数の値が未定義状態である初期変数割当が用意され，この変数割当に対して制約を順番に適用していくことで，変数の値が計算される．ただし，本制約解消系は，制約が適用された時点で適用前の変数割当を捨てて新しい変数割当に置き換えるのでなく，適用前の変数割当を保存しつつ，新しい変数割当を変数割当集合に加えるという点で CLP($\Re$) の等式制約解消系と異なっている．また矛盾した制約が発見された場合，CLP($\Re$) では処理が停止するが，本処理系では単に無視されるだけである.

図 11 で，変数の数を 2 つに簡略化した例 (本来は 4 変数) を用いて制約解消系の動作の具体的な説明を行なう.

(1) 始めに空の変数割当が生成され，次に最初の制約 (`x=1`) が適用される.

(2) この制約は矛盾なく適用することができ，新しい変数割当 (`1,-`) が生成される.

(3) この時，最初の空の変数割当は破棄されずに保存される．(`x-y=0`) という制約を (`1,2`) という変数割当に

(4) 適用しようとすると，これは矛盾した操作であるので失敗し，新しい変数割当は生成されない.

(5) 一方，この制約は空の変数割当 (`-,-`) に矛盾なく適用が可能であるが，この時点では値が確定されないのでサスペンドされた制約をもった変数割当が生成される.

(6) サスペンドされた制約については，十分な数の変数が確定するか連立方程式として十分な数の制約が追加された時点で計算が行なわれる.

(7) 無駄な計算を防ぐため，同一の変数割当が生成された場合には一方を破棄する.

(8) 最後に，すべての制約についての処理が終った時点ですべての変数が確定している変数割当を最終的な解として返す.

処理を効率的に行なうため，計算途中の変数割当は空の変数割当を根とする木構造の形で保存される．すべての変数割当は，空の変数割当を起源とするいずれかの変数割当から制約を追加することによって派生したものであり，木構造はこの親子関係を表現したものである．木構造を利用すると，ある変数割当に対する制約の適用が矛盾を起こして失敗した場合，この変数割当を親とするすべての変数割当に対してこの制約の適用が失敗することが明らかなので，計算量を大幅に減らすことができる.

現在の実装は線形の制約のみを扱っているので，連立方程式を解く際の基本的なアルゴリズムとしてガウスの消去法を利用している．線分の長さの一致や円と線分の接触といった非線形の制約を扱うためには，ニュートン法[6)8)] のような別のアルゴリズムを利用する必要がある．端点同士の接続，合同と対称，および幅の一致といった制約は，2 つの方程式の組み合わせとして表現される (図 10 参照) が，これらは and 条件で結ばれているものと解釈される．すなわち，ある変数割当に対して一方の方程式が適用不能であった場合には，もう一方の方程式も自動的に無視される.

5. プロトタイプシステム Pegasus

プロトタイプシステム Pegasus は，Microsoft Visual Basic および Visual C++ を用いて実装されおり，Microsoft Windows 上で動作する．ユーザ・インタフェース部は Visual Basic で，整形処理部は実行速度向上のため Visual C++ で記述されている.

本システムは Windows 95 および 3.1 の動作するすべての PC 互換機上で動作するが，主にペンによる入力をターゲットとしたシステムであり，携帯型のペンコンピュータ (三菱電機 AMiTY) や電子黒板 (Xerox Liveboard) などを使用してテストを行なっている.

次に，ペガサスを利用して描かれた図の例を幾つか示す．**図 12** は，対話的整形を授業で利用した場合を想定

☆ 各変数を入力座標そのものに結び付ける制約が自動的に生成されるので制約が不足する (under-constrained) ことはない

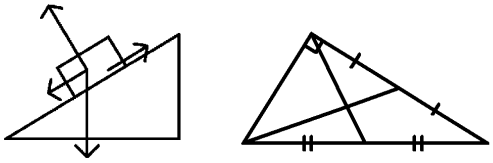


図 12 物理や数学の授業で使われる図の描画例
Fig. 12 Diagrams for physics and mathematics

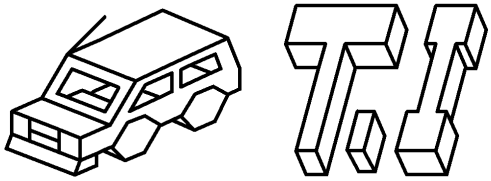


図 13 平行投影による三次元の図の例
Fig. 13 Three-dimensional illustrations

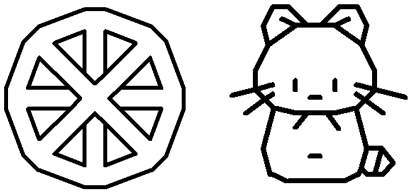


図 14 幾何学的なイラストレーションの例
Fig. 14 Geometric illustrations

した例である．従来のメニューに基づく描画方式は，操作に手間がかかるために学校での授業のような対話的な場面での利用が困難であった．対話的整形は，チョークで黒板に図を描くのとほとんど変わらない操作で電子黒板上に正確な図形を描くことを可能にするものであり，その効果は非常に大きいと期待される．**図 13** は，3 次元表現の図の例である．このような平行投影の図の描画は，線分の平行化および合同化，端点接続といった処理によって可能になる．特殊な処理や操作を使用せず，平行や合同といった一般的な制約のみでこのような図が正確に描かれている点に注目されたい．**図 14** 左は，幾何学的図形の例を示している．ここでは，回りの輪の幅とスポークの幅が等しくなっているが，このような図形の描画は従来の描画システムでは非常に困難である．図 14 右は，左右対称なイラストの例である．特別なコマンドを使用することなく対称性が自動的に満たされていくため，デザイナはコマンド操作に気をとられることなくデザイン自体に集中することができる．

6. 評価実験

ここでは，実装したプロトタイプシステムと市販の

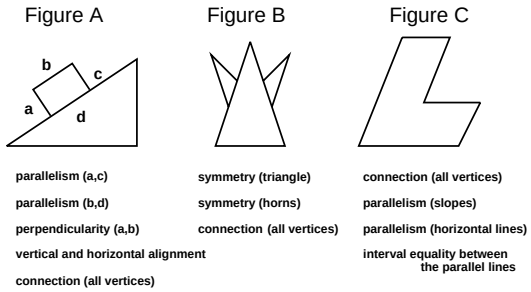


図 15 実験で使用した 3 種類の図と幾何学的制約
Fig. 15 The diagrams used in the experiment, and required geometric relations

CAD ソフトウェアおよびドロー系描画ソフトウェアを使用して行なった対話的整形の評価実験について述べる．今回の実験では特に，対話的整形によって図形を描くのに必要な時間がどの程度短縮されるか (描画の速さ)，および図形ごとに指定した幾何学的制約がどの程度忠実に充足されているか (描画の正確さ)，について注目する．

6.1 方　　法

6.1.1 システム

実験は，三菱電機のペンコンピュータである AMiTY SP (i486DX4 75MHz, Windows95) を利用して行なった．ソフトウェアとして，前述のプロトタイプシステムに加え，市販の CAD ソフトウェア (AutoDesk 社 AutoSketch, 以下 CAD) とドロー系描画ソフトウェア (FutureWave Software 社 SmartSketch, 以下 Draw) を使用した．ここで，CAD は正確な幾何学的図形を描くために一般に使用されているソフトウェアの例として，一方 Draw は特に幾何学的制約を必要としない手軽な描画ソフトウェアの例として使用している．

6.1.2 タスク

各被験者は，以上の 3 つのソフトウェアを使用して，**図 15** に示す 3 つの図形を描くように指示される．また，1) 指定された幾何学的関係を出来るだけ満たしつつ，できるだけ短時間に描画を終了すること，2) 描画時間の上限を 5 分とし，これを越えたら描画を中断すること，3) 幾何学的関係を満たすことが非常に困難であると感じた場合には，無理をせずに大体の形を再現すればよいこと，といった指示が与えられる．

6.1.3 被験者

18 名の学生が被験者として実験に参加した．計算機環境一般および各ソフトウェアの使用経験は各人によって大きく異なっている．被験者のうち 8 人は計算機の使用に慣れ親しんでおり一般的な描画ソフトウェアの操作経験があるが，残りの被験者は計算機の使用経験がほと

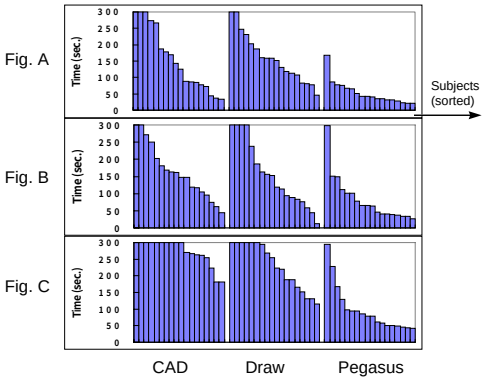


図 16 各被験者の描画時間
Fig. 16 Drawing time required for each task

んどない.

6.1.4 手 続 き

学習による実験データの偏りを避けるため，各ソフトウェアの使用順序を影響が相殺されるように被験者の間で規則的に変化させる．実験全体で，18 (被験者) × 3 (ソフトウェア) × 3 (図) = 162 の描画プロセスを観察する．各描画プロセスは 5 分以下の長さであり，すべてビデオテープに記録する．

各ソフトウェアを使用した本実験に先立って，各被験者にそれぞれのソフトウェアの使用法の簡単な説明と練習課題を与える．この説明と練習に必要な時間は被験者およびソフトウェアにより多少変化するが，一般的に，CAD ソフトウェアの説明に多くの時間が必要となる．

6.2 実験の結果と考察

6.2.1 描画の速さについて

図 16 に，各被験者が各描画タスクを終了するのにかかった時間を示す．各カラムは一人の被験者が各描画エディタを使って各図を描くのに要した時間に対応する．被験者は描画時間の順に並べられている．描画時間の上限は前述のように 300 秒に制限されているので，これを越えたものについては 300 秒として示した．プロトタイプシステムを利用した場合は，他のソフトウェアを利用した場合に比べて，明らかに描画時間が短くなっていることがわかる．

図 17 に，各描画ソフトウェアごとの制限時間内に終了した描画タスクの割合を示す．CAD を使用した場合で 28 %，Draw では 20 % の被験者が制限時間内に描画を終えることができなかったのに対し，プロトタイプシステムでの描画はすべて 300 秒以内に終了している．ここでの描画終了の基準は，被験者本人が終了と申告したものであり，必ずしも指定された幾何学的制約がすべて満たされているとは限らない．

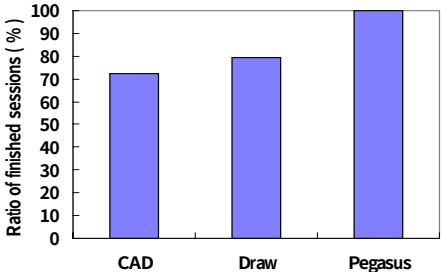


図 17 制限時間内に終了した描画タスクの割合
Fig. 17 The ratio of finished sessions

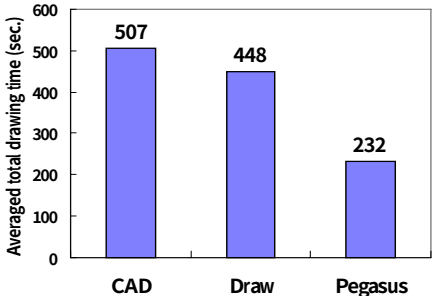


図 18 3 つの図を描いた場合の推定合計描画時間
Fig. 18 Estimation for time required for a subject to draw the three diagrams

300 秒で打ち切っているために描画時間の平均を求めることは不可能であるが，参考として**図 18** に，平均的な被験者が各エディタを使用して 3 つの図を描いた場合の描画時間の推定値を示す．この値は，各エディタでの各図の描画時間のうち 300 秒以内のものの平均をすべての被験者に渡ってとり，さらに各エディタごとに 3 つの図の描画時間を合計したものである．この計算結果によると，プロトタイプでの描画は平均的に Draw での描画時間の 52 %，CAD での描画時間の 46 % で終了していると言える．この計算は 300 秒を越えるものを含んでいないため，実際の描画時間の差はこれよりも大きなものになると予想される．

6.2.2 描画の正確さについて

描画時間が短縮されたとしても，描かれた図の正確さが著しく損なわれるようでは意味がない．そこで，各描画ソフトウェアを利用して描かれた図のうち，図 15 に指定された幾何学的制約をすべて満たしているものの割合を調べた．その結果を**図 19** に示す．被験者が 300 秒以内に描画を終了したが制約が完全に満たされていないものは含まない．同図と図 18 を比べると，Draw は描画時間で CAD を上回っていたが，制約の充足度では

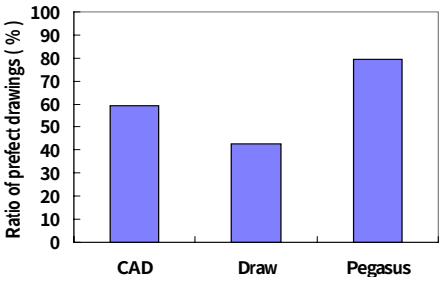


図 19 要求した幾何学的制約が完全に満たされた図の割合
Fig. 19 The ratio of diagrams where required constraints are perfectly satisfied

CAD の方が上回っている様子が示されており，それぞれのソフトウェアの特徴がよくでていると言える．しかし，プロトタイプシステムはその双方に対して，描画時間と制約充足度の両方の面でより優位な結果を示している．

この実験では，テキストや曲線を含まない図のみを対象にするなど描画エディタに関する多様な要素が捨象されており，得られた結果はあくまでも予備的なものである．しかし，このような限界があるとはいえ，多くの被験者にとって対話的整形が未知の描画手法であったことを考慮すると，本実験を通して対話的整形という描画手法の可能性は十分に示されたと言えよう．特に，実験で用いたような簡単な幾何学的な図形の描画に関しては，描画時間および描画の正確さの両点において，既存の描画手法に対する本手法の明らかな優位性が認められる．

7. 今後の課題

現在のプロトタイプシステムの重大な欠点として，数多くの候補が生成されて提示された場合に選択が困難になるという問題がある．この問題は，複雑な図を描く場合に深刻になる．解決法としては，生成される候補の数を減らす方法，および選択のためのインタフェースを工夫する方法が考えられる．候補数を絞り込むための手法としては，空間的に近い線分や，時間的に近い (最近描いた) 線分のみを考慮するようにしたり，頻繁に使用する幾何学的制約のみを扱い，めったに使われない幾何学的制約を無視するということが考えられる．インタフェースの改良としては，ユーザに整形処理において考慮する必要のない領域を指定させたり，逆に幾何学的関係を考慮すべき線分を直接指示させたりするといったことが可能である．

また，実用化に当たって必要と考えられる機能に，一度確定した図形の移動や回転といった編集操作が挙げられる．対話的整形の基本的な設計方針は，認知的な負荷の高い編集操作を避け，単純なフリーストロークと整形で描画を完結させようというもので (図形を移動させたい場合には消して書き直す)，簡単な図形の描画においては描画効率の向上につながっている．しかし，より複雑な応用を考えた場合には移動や拡大縮小といった操作も当然必要であると考えられ，将来的には対話的整形と編集操作との自然な融合の方法を探っていく必要がある．

今後は，現在のプロトタイプを拡張し，曲線やテキスト，および描画パターンを実装していく予定である．円や曲線の導入は難しい課題であるが，既存の描画エディタで曲線に関する幾何学的制約を満たすことは特に困難であり，対話的整形に対する期待は大きい．

また，対話的整形の 3 次元モデル生成[21] への拡張も検討している．3 次元モデルに関しては，3 次元空間に広がる複数の候補をどのようにして見やすく提示するかが問題となるであろう．

認識アルゴリズム自体に関しては，現在固定的に使用している様々な閾値をユーザに応じて動的に変化させていくことも検討している．現在までの実験においてはユーザの側による適応が十分働いており重大な問題とはなっていないが，より長期的には取り組むべき課題である．

8. おわりに

本論文では，幾何学的制約を満たす図形を手早く描くための描画手法である対話的整形について説明した．対話的整形システムは，基本的には整形システムであり，フリーストロークで描かれた線分を幾何学的制約を満たすベクトル表現の線分に置き換える．対話的整形の特徴は，一ストローク毎の整形と大局的な幾何学的制約の抽出と充足，および複数候補の生成機構にある．これらの機構によって，初心者でも様々な幾何学的制約を満たす図を手早く確実に描くことが可能になる．整形処理部は，制約抽出モジュール，制約解消モジュールおよび候補の評価モジュールによって構成されており，効率的に処理が行なわれる．プロトタイプシステム (Pegasus) がペンコンピュータ上に実装されており，実時間での処理が可能となっている．本システムおよび市販のソフトウェアを使用した評価実験を行ない，対話的整形によって幾何学的図形の描画時間と描画の正確さが共に改善されることを確認した．

対話的整形は，通常の CAD システム上での幾何学的形状のモデリングに使用することも考えられるが，特にペンコンピュータでの簡単な図形の描画に最適な手法である．具体的には，携帯型のペン入力 PDA でのメモ書

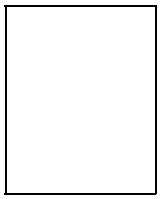
きや，授業や会議で使用する電子黒板上での描画などに有効と考えられる．また，本手法はコマンドによらない描画を実現しており，従来複雑なコマンド操作を避けて紙の上で行なわれていたデザイン作業を計算機上で行なうことを可能にするもの[13]と期待される．

謝辞 本論文をまとめるに当たり有益なアドバイスを下さった田中研究室 および TRIP グループのメンバーに感謝する．また，実験の被験者となってくださった多くの方々に感謝の意を表する．

参 考 文 献

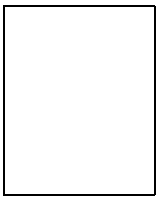
1) Apte,A., Vo,V., Kimura,T.D. : Recognizing Multistroke Geometric Shapes: An Experimental Evaluation, *Proc. of UIST'93*, pp. 121-128, (1993).
2) Bier,E.A., Stone,M.C. : Snap Dragging, *Proc. of SIGGRAPH '86*, pp. 233-240, (1986).
3) Borning,A. : The Programming Language Aspects of ThingLab, A constraint-Oriented Simulation Laboratory, *ACM Trans. on Program. Lang. Syst.*, Vol.3, No.4, pp.353-387, (1981).
4) Bouma,W., Fudos,I., Hoffman.D., Cai,J., Paige,R. : Geometric constraint solver, *Computer Aided Design*, Vol.27, No.6, pp. 487-501, (1995).
5) Chen,C.L.P., Xie,S. : Freehand drawing system using a fuzzy logic concept, *Computer Aided Design*, Vol.28, No.2, pp.77-89, (1996).
6) Conte,S.D., Boor,d.C. : Elementary Numerical Analysis, McGraw-Hill, (1972).
7) Gross,M.D., Do,E.Y. : Ambiguous Intentions: A Paper-like Interface for Creative Design, *Proc. of UIST'96*, pp. 183-192, (1996).
8) Heydon,A., Nelson,G. : The Juno-2 Constraint-Based Drawing Editor, SRC Research Report 131a, System Research Center, Digital Equipment Corporation, Palo Alto (1994).
9) Hopkins,D. : The design and implemetation of pie menus, *Dr.Dobb's Journal* 1, Vol.6, No.12, pp.16-26 (1991).
10) Igarashi,T., Kawachiya,S., Matsuoka,S., Tanaka,H. : In Search for an Ideal Computer-Assisted Drawing System *Proc. of INTERACT'97*, pp.104-111 (1997).
11) Jaffar,J., Michaylov,S., Stuckey,P.J., Yap,R.H.C. : The CLP(ℜ) Language and System, *ACM Trans. on Program. Lang. Syst.*, Vol.14, No.3, pp. 339-395 (1992).
12) Kurlander,D., Feiner,S. : Interactive Constraint-Based Serach and Replace, *Proc. of CHI'92*, pp.609-618 (1992).
13) Lakin,F., Wambaugh,J., Leifer,S., Cannon,D., Steward,C. : The electronic notebook: performing medium and processing medium, *Visual Computer*, Vol.5, pp.214-226 (1989).
14) Landay,J.A., Myers,B.A. : Interactive Sketching for Early Stages of User Interface Design, *Proc. of CHI'95* , pp. 43-50 (1995).
15) 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).
16) Pavlidis,T., VanWyk,C.J. : An Automatic Beautifier for Drawings and Illustrations, *Proc. of SIGGRAPH '85* , pp. 225-234 (1985).
17) Rubine,D. : Combining Gestures and Direct Manipulation, *Proc. of CHI'92*, pp.659-660 (1992).
18) Saund,E., Moran,T.P. : A Perceptually Supported Sketch Editor, *Proc. of UIST'94*, pp. 175-184 (1994).
19) Sutherland,I.E. : Sketchpad: A Man-Machine Graphical Communication System, *Proc. of Spring Joint Computer Conf.*, No.23, pp.329-346 (1963).
20) Zao,R. : Incremental Recognition in Gesture-Based and Syntax-Directed Diagram Editors, *Proc. of InterCHI'93*, pp. 95-100 (1993).
21) Zeleznik,R.C., Herndon,K.P., Hughes,J.F. : SKETCH: An Interface for Sketching 3D Scenes, *Proc. of SIGGRAPH '96* , pp. 163-170 (1996).

(平成?年?月?日受付)
(平成?年?月?日採録)



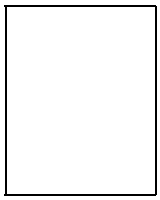
五十嵐健夫（学生会員）

1973 年生．1995 年東京大学工学部計数工学科卒．1997 年同大学大学院工学系研究科情報工学専攻修士課程修了．現在，同博士課程在学中．1997 年 XEROX PARC summter student, ユーザインタフェース全般，CAD システム，ソフトウェアエージェントなどに興味を持つ．情報処理学会，人工知能学会，日本ソフトウェア科学会，各会員．



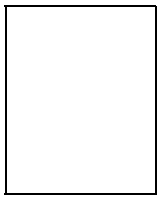
松岡　聡（正会員）

1963 年生.1986 年東京大学理学部情報科学科卒,1989 年同大学大学院博士課程中退. 同大学情報科学科助手, 情報工学専攻講師を経て,1996 年より東京工業大学情報理工学研究科数理・計算科学専攻助教授. 理学博士. オブジェクト指向言語, 並列システム, リフレクティブ言語, 制約言語, ユーザ・インターフェースソフトウェアなどの研究に従事. 現在進行中の代表的プロジェクトは, 世界規模の高性能計算環境を構築する Ninf プロジェクト, 計算環境に適合・最適化を目指す Java 言語の開放型 Just-In-Time コンパイラ OpenJIT, 制約ベースの TRIP ユーザインターフェースなど. 並列自己反映型オブジェクト指向言語 ABCL/R2 の研究で 1996 年度情報処理学会論文賞受賞. 1997 年はオブジェクト指向の国際学会 ECOOP'97 のプログラム委員長を務める. 情報処理学会, ソフトウェア科学会, ACM, IEEE-CS 各会員.



河内谷幸子（正会員）

1963 年生. 1986 年東京大学理学部情報科学科卒. 1987 〜 1990 年日立製作所システム開発研究所, 1990 〜 1993 年青山学院大学理工学部助手. 1994 年東京大学大学院理学系研究科情報科学専攻修士課程修了, 同大学博士課程進学. コンピュータ・グラフィックス, 文書画像処理, ユーザ・インタフェースに興味を持つ. 電子情報通信学会, 情報処理学会, 日本ソフトウェア科学会, 各会員.



田中　英彦（正会員）

1943 年生. 1965 年東京大学工学部電子工学科卒業. 1970 年同大学院博士課程終了. 工学博士. 同年東京大学工学部講師.1971 年助教授,1978 年〜1979 年ニューヨーク市立大学客員教授,1987 年教授現在に至る. 計算機アーキテクチャ, 並列処理, 人工知能, 自然言語処理, 分散処理,CAD 等に興味を持っている. 「非ノイマンコンピュータ」,「情報通信システム」著,「計算機アーキテクチャ」,「VLSI コンピュータ I,II」,「ソフトウエア指向アーキテクチャ」共著,New Generation Computing 編集長, 情報処理学会, 電子情報通信学会, 人工知能学会, ソフトウエア科学会,IEEE, ACM, 各会員.