

平成18年度

筑波大学第三学群情報学類

卒業研究論文

題目

マーキングジェスチャを利用した  
フリーハンド作図インタフェース

主専攻 情報科学主専攻

著者 成川 新吾

指導教員 田中二郎 三末和男 志築文太郎 高橋伸

## 要 旨

創造作業の初期段階において手書きは重要視されており，手書きを利用し創造作業の支援を目的とした様々なツールが開発されている．それらのツールを利用しているときには図形を多々描くが，フリーハンド図形を利用者の好きなように編集することはできないのが現状である．また，マーキングジェスチャの記号は，算数や数学の授業などで用いられる等長や平行の記号であるために，一般的によく知られていて誰にでもすぐにわかるという利点がある．本研究の目的は，創造作業における手書きの利点とマーキングジェスチャの利点の両方を生かし利用することで，創造作業の支援を行うことである．そこで，マーキングジェスチャをフリーハンド図形に行うことで，フリーハンドのストロークのままマーキングジェスチャの記号の示す通りに図形を変形させるシステムを提案し，開発した．それにより，フリーハンド図形の変形が容易に行えるようになった．

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	研究の背景	1
1.2	ペン入力の特長	2
1.3	マーキングジェスチャ	2
1.4	研究の目的	2
1.5	本論文の構成	3
<b>第2章</b>	<b>マーキングジェスチャを利用したフリーハンド作図システム</b>	<b>4</b>
2.1	マーキングジェスチャの利用	4
2.2	フリーハンド図形の変形	6
2.3	頂点の移動	6
<b>第3章</b>	<b>試作システムの設計と構成</b>	<b>8</b>
3.1	システムの設計	8
3.2	システムの構成	8
<b>第4章</b>	<b>実装</b>	<b>11</b>
4.1	マーキングジェスチャの認識	11
4.2	フリーハンド図形の変形	14
4.2.1	等長	14
4.2.2	平行	17
4.2.3	直角	19
4.3	状態の保存・解除	21
4.3.1	保存	21
4.3.2	解除	23
<b>第5章</b>	<b>考察と今後の課題</b>	<b>24</b>
5.1	考察	24
5.1.1	マーキングジェスチャの認識	24
5.1.2	フリーハンド図形の変形	24
5.2	今後の課題	24
<b>第6章</b>	<b>関連研究</b>	<b>25</b>

6.1	創造作業の支援に関する研究 . . . . .	25
6.2	ペン指向作図インタフェースに関する研究 . . . . .	25
第7章	結論	27
	謝辞	28
	参考文献	29

# 図目次

2.1	等長記号 . . . . .	4
2.2	平行記号 . . . . .	5
2.3	等角記号 . . . . .	5
2.4	直角記号 . . . . .	6
2.5	フリーハンド図形の完全整形 . . . . .	6
2.6	頂点の移動の例 . . . . .	7
3.1	システムの処理の流れ . . . . .	10
4.1	マーキングジェスチャ誤認識の例 . . . . .	12
4.2	平行記号の認識 . . . . .	12
4.3	直角記号の認識 . . . . .	13
4.4	平行記号の向きの例 . . . . .	13
4.5	等長記号の移動を行わずに発生した問題の例 . . . . .	15
4.6	等長記号による図形の変形 . . . . .	16
4.7	平行記号による図形の変形 . . . . .	18
4.8	自動的に消去された平行記号 . . . . .	19
4.9	直角記号による図形の変形 . . . . .	20
4.10	辺の状態が保存された変形 . . . . .	22
4.11	辺の状態が解除された変形 . . . . .	23

# 第1章 序論

## 1.1 研究の背景

ペンを用いて手書きで何かを書くということは、身近で自然な表現方法である。子供の頃にペンを用いて書くことを覚えて以来、常にとっていいほど鉛筆、ボールペン、筆など形は変われどもペン型のものを用いて書いている。コンピュータの世界はというと、マウスやキーボードといったペン型以外の入力デバイスが一般的に用いられている。しかし近年では、タブレットPCの普及に伴っての性能向上、低価格化や、家庭用ゲーム機、ペン入力を採用したPDA、大画面ディスプレイの普及によって、スタイラス(ペン)でコンピュータへの入力を行うペンユーザインタフェースの認知度が上がっている。

人間が創造作業を行う際、デザイナーによっては創造作業の初期段階、発想するという段階においては紙とペンを利用している[1]。また、創造作業の初期段階における手書きの良さにおいては[2]で言及されている。James A. LandayらによるSILK[3]ではGUIのデザインが行えるツールが実現されている。SILKでは認識したWidgetsを完全に変換して表示することはせず、敢えて手書きのまま表示している。ZelevnikらによるSKETCH[4]やIgarashiらによるTeddy[5]においても、3次元データを画面に表示するときに、敢えて手書き風に崩して表現する手法(Non-photorealistic rendering)が使われている。このような「崩した表現を利用することで自由な発想が促進される」という点について実際に実験で示した例も報告されている[6]。

SILKに代表されるようなデザインを行うツールを使用する際、フリーハンドの図形は重要な役割を果たす。そこで私は創造作業の初期段階におけるフリーハンド図形に着目した。それらのツールでは、描いた図形を利用者の好きなように編集することができないのが現状である。描いた後に利用者の望んだ通りに、簡単にフリーハンドの図形を変形することができれば、デザインにおいて新たな発想を生み出す手助けになるのではないかと考えた。

ここで、創造作業の初期段階を支援するツールに対する要求について[2]を参考に整理しておく。

- [簡単]: 浮かんだ発想を即座に書き留められること
- [直接]: 書き留める際に、無用な間接作業がないこと
- [柔軟]: 自由に書き留められること
- [インフォーマル]: 曖昧で不完全な情報を適切に書き留められること

## 1.2 ペン入力の特長

[簡単]を満たすために、ペン入力デバイスを利用する。コンピュータにおけるペン入力の最大の特長は紙に描くように文字や図形を手書きで描ける点だからである。

まずペンであるという点から考えると、ペンは子供の頃から使用していて、年を重ねても日常的に使用しているものであるために、非常に使うことに慣れていて、苦手であることがほとんどないことがある。また、コンピュータにおけるペン入力以外の道具と比較して考えると、コンピュータをまったくと言っていいほど使用したことのない人がマウスを使用すると、マウスカーソルをクリックしたい位置に移動させることすらできないことがある。それは、手元で操作しているにも関わらず、コンピュータの画面を見なければならず、操作する位置と実際の入力位置が違うために起こると考えられる。マウス操作に慣れていない人においても、マウスを使用し文字や図形を描くということになると驚くほどに難しい。

## 1.3 マーキングジェスチャ

ペン入力の利点を生かしたスケッチベースの様々な研究が行われている [7]。作図インタフェースにおいても、多くの研究が行われている [8][9][10][11]。加藤らのペン入力指向の図形整形インタフェース [12] では、マーキングジェスチャという対話技法を定義している。マーキングジェスチャとは、等長や平行を意味する記号や、角度や辺長を表す数値を直接図形上に描く対話技法のことである。マーキングジェスチャは、直接図形上に描くのでペン入力に適していると考えられる。また、マーキングジェスチャとして描く記号は、等長や平行の記号は、算数や数学の授業において図形の説明や状態を表すためによく用いられている。一般的によく知られている記号であるために、利用者は新たなジェスチャを覚えるという負担がなく利用することができる利点がある。

これらの利点から、マーキングジェスチャは、[簡単]、[直接]、[柔軟]を満たすことができると思われる。

## 1.4 研究の目的

フリーハンド図形に対し、マーキングジェスチャを利用して変形を指示し、フリーハンドの図形のまま変形できるようにすることで [簡単]、[直接]、[柔軟]、[インフォーマル]の全てを満たすことが可能になる。マーキングジェスチャを利用してフリーハンド図形を利用者の望むように容易に変形・整形することによって、ただ雑然としていたスケッチのときとは違った見方ができるようになる。そして新たな発想を生み出すきっかけを与え、手助けすることができるのではないかと考えた。

そこで本研究では、マーキングジェスチャを利用したフリーハンド作図インタフェースを実現することで、創造作業を支援することを目的とする。

## 1.5 本論文の構成

本論文の構成は次の通りである。本章では、創造支援の初期段階における手書きの重要性について、既存のツールや論文を参考に整理した。その上で、フリーハンド図形における操作に焦点を当てた。次に、マーキングジェスチャについて説明し、研究の目的を述べた。第2章ではマーキングジェスチャを利用した作図システムについて提案し、機能について説明する。第3章では第2章で提案したシステムにおいて試作したシステムの設計を行い、第4章で実装について説明する。第5章でシステムについて考察を行い、今後の課題を述べる。第6章で関連研究を紹介し、第7章でまとめる。

## 第2章 マーキングジェスチャを利用したフリーハンド作図システム

### 2.1 マーキングジェスチャの利用

本研究では、フリーハンド図形の変形を行う指示としてマーキングジェスチャを利用する。加藤らによるマーキングジェスチャは、教育（特に初等教育）において利用することを目的として設計されている。本研究では創造作業の初期段階での利用を想定しているために、マーキングジェスチャとして使用する記号において、どの記号が適切かを考えなければならない。

フリーハンドの図形の変形を行うために、辺の長さや角度を細かく指定できる必要はないと考えられる。そこで、以下の4つの記号が重要であると考えた。

#### 1. 等長

図2.1の赤線が示すような記号を描くことで、等長を意味する。記号が描かれた図形の辺の長さが変化し同じ長さになる。

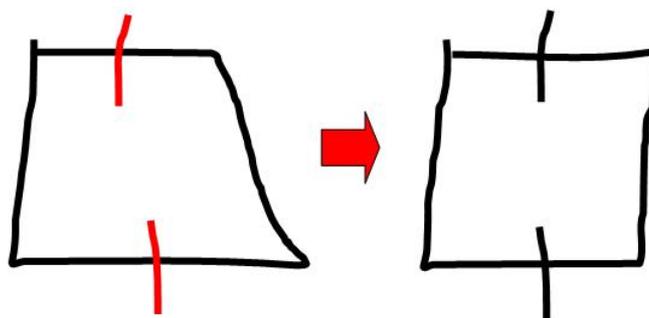


図 2.1: 等長記号

## 2. 平行

図 2.2 の赤線が示すような記号を描くことで、平行を意味する。記号が描かれた図形の辺と辺が平行になる。

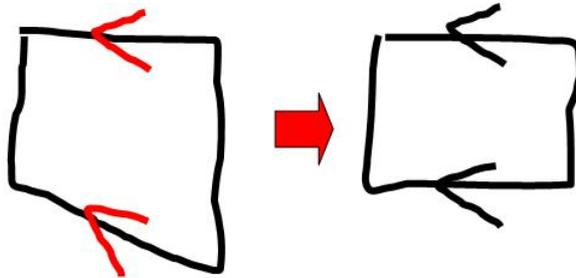


図 2.2: 平行記号

## 3. 等角

図 2.3 の赤線が示すような記号を描くことで、等角を意味する。記号が描かれた図形の角と角が同じ角度になる。

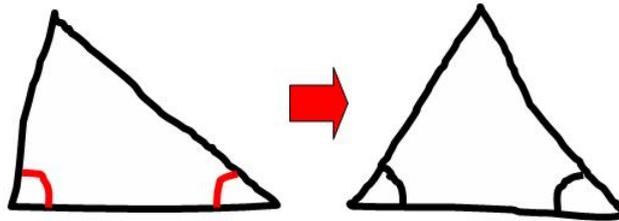


図 2.3: 等角記号

## 4. 直角

図 2.4 の赤線が示すような記号を描くことで、直角を意味する。記号が描かれた図形の角が直角になる。

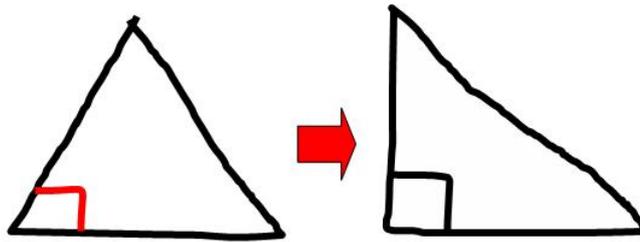


図 2.4: 直角記号

## 2.2 フリーハンド図形の変形

本システムでは、マーキングジェスチャを行うことで、マーキングジェスチャとして描かれた記号の指示に従って図形が変形する。フリーハンドの図形をマーキングジェスチャにより変形を行う際、整った線にしてしまうことは[インフォーマル]であることを失わせてしまうことになる(図 2.5)。そのため、フリーハンドのまま変形を行う。変形する図形はフリーハンドで描かれていて、変形後もフリーハンドで描かれたストロークの状態が保たれている。

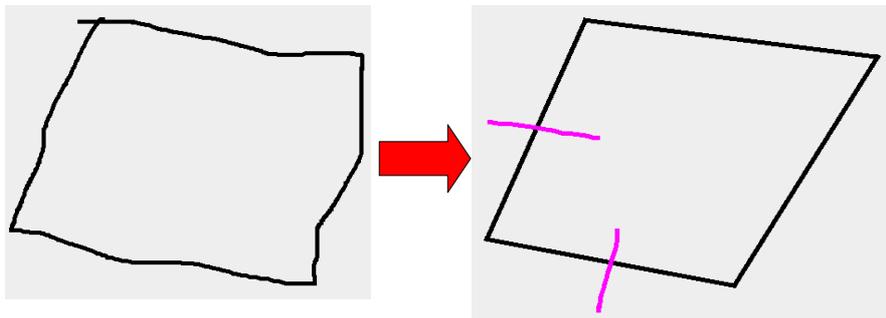


図 2.5: フリーハンド図形の完全整形

変形を行うために、基準を設定しなくてはならない。等長や平行記号においては2つ以上描かれて意味のある記号である。そこで、1つ目に描かれた記号のある辺を基準に変形を行うことにする。

## 2.3 頂点の移動

マーキングジェスチャにより、辺・図形の状態を指示し、指示された状態が保持される。保持されていることを生かし、頂点の移動が行えるようにすることで、利用者の好きなように

変形することができるようになる。そこで、マーキングジェスチャによって指示された図形の状態を保存したまま、頂点の移動を行えるようにすることを考えた。

頂点の移動とは、例えば図 2.6 に示すような動作を指す。三角形 ABC において頂点 A の移動を行っている。等長記号を描いていない、上の三角形は A をどこにでも動かすことができる。等長記号を描いている、下の三角形は等長である状態を保存しているので、保ったままの移動（ここでは上下動作）のみ可能である。

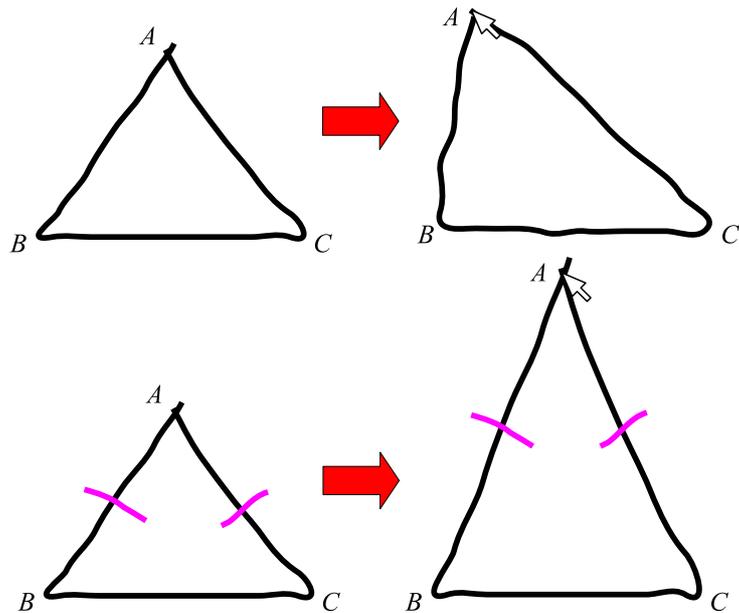


図 2.6: 頂点の移動の例

## 第3章 試作システムの設計と構成

### 3.1 システムの設計

マーキングジェスチャを利用したフリーハンド作図システムの理想形は、まず第2章で示した機能を全て満たすことである。さらに、必要であると思われる機能を評価実験などにより調査し、付加することで、さらに利用者の望む図形の変形を容易に行えるようにする。そして、既存の他のツールに組み込めるようにすることである。

本論文では、2章のアイデアの中核を実現することで、アイデアの有効性を立証するために、以下のような制約を持ったシステムの設計を行った。

1. 図形は四角形において行う：

創造作業の初期段階において、四角形を描くことは大変多い。また、四角形は自由度が高く、他の図形の基本となり得る図形であるために、四角形において実装を行うことで三角形など他の図形に容易に応用が可能である。

2. 四角形の底辺を水平にし、固定する：

底辺を水平にし固定することで、変形が直感的に理解しやすくなる。

3. マーキングジェスチャの記号は、等長・平行・直角とする：

等長・平行・直角は、記号の中において基本であり、また平行においては四角形において重要な記号である。

これらの特徴を持ったシステムを実現することで、第2章で示した機能の実現への応用がしやすくなると考えられる。

### 3.2 システムの構成

本システムの構成は主に入力部・処理部・出力部の3つからなる。それぞれの役割について、UMLのアクティビティ図を用いて記述した図3.1に示し、以下で説明する。

- 入力部

入力部では、利用者によってツールのキャンバス上で行われたマウスイベントを取得し、マウスイベントによって分岐する。左クリック（スタイラスでは通常入力）では図形入力として、右クリック（スタイラスではバレルボタンを押しながらの）ではマーキングジェスチャの入力として、それぞれ処理部へ渡す。

- 処理部

まず入力部からの入力が左クリックによるものであった場合の説明を行う。マウスイベントが `mouseClicked` であった場合、ダブルクリックであるかを判定し、ダブルクリックであった場合は全ての情報を消去し、初期化する。 `mousePressed` であった場合はドラッグによる座標を用いて、リリースされた時点で角の検出を行い図形の認識を行う。認識した図形が四角形であればストローク情報を登録し、四角形以外であったらドラッグ中に登録した情報を消去する。左クリックによる処理が終了し、出力部へ渡す。

次に入力部からの入力が右クリックによるものであった場合の説明を行う。マウスイベントが `mouseClicked` であった場合、ダブルクリックであるかを判定し、ダブルクリックであった場合はマーキングジェスチャとして登録されている情報を消去する。 `mousePressed` であった場合は、ドラッグによる座標を用いて、リリースされた時点でマーキングジェスチャの認識を行う。認識によって「等長」・「平行」・「直角」に分類される。現在の図形の状態を確認し、分類された記号と図形の状態を基に変形を行う。変形によって新たな座標を求め、座標の情報を出力部へ渡す。

- 出力部

処理部から渡された座標の情報を基に、ストロークを描き描画処理を行い画面に出力する。

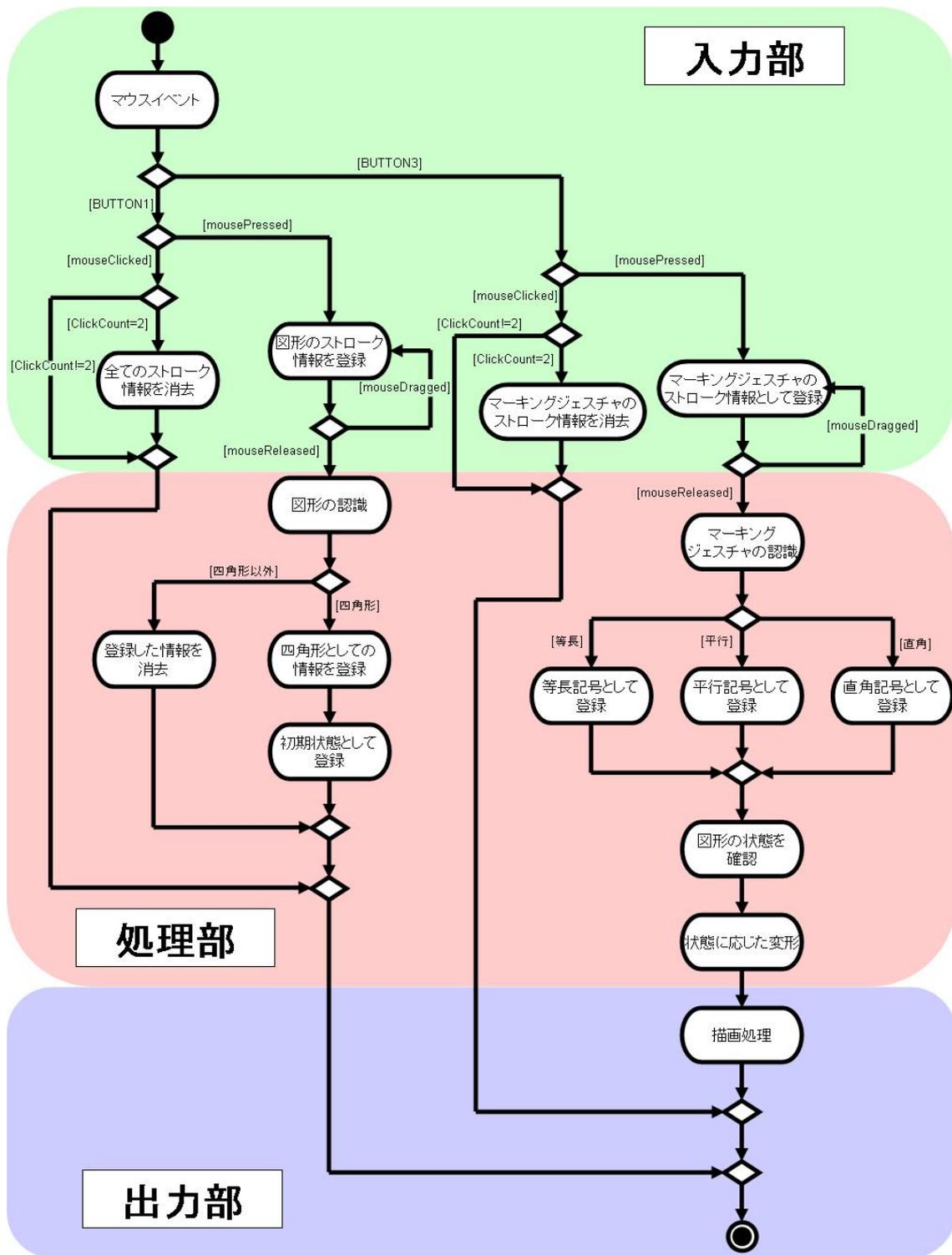


図 3.1: システムの処理の流れ

## 第4章 実装

システムの実装には Java<sup>TM</sup> Platform Standard Edition 5.0 を用いた。

### 4.1 マーキングジェスチャの認識

ジェスチャ形状の認識アルゴリズムとして、Rubine のアルゴリズム [13] が非常に多くのシステムで利用されている。しかし本研究で使用するマーキングジェスチャとしての記号は、種類があまり多くなく、図形に対するジェスチャである。そのため記号の特徴を考え実装を行うだけで十分であると考え、実装を行った。

「等長」「平行」「直角」の記号の認識において実装を行った。

ここで、等長、平行と直角記号の特徴について簡単に整理する。

- 等長：1. 一本の直線である（角がない）  
2. 辺をまたいでいる
- 平行：1. 二本の直線で形成されている（角が1つである）  
2. 辺をまたいでいる
- 直角：1. 二本の直線で形成されている（角が1つである）  
2. 角に対する指示であるので、角に近い

本研究ではフリーハンドのストロークを扱う。そのため直線とは、ストロークの始点から終点まで、もしくは角が認識された場合は、始点から角、角から角、角から終点までなどを直線であると定義する。

上記に示す特徴を元に実装を行ったが、これらの特徴だけでは図 4.1 に示すような誤認識を代表して、他にも多々起きてしまうという問題があった。図 4.1 では、平行の記号を描いたとは考えにくいですが、角が一つ存在するために平行の記号であると認識されてしまった。そこで、もう一度詳しくそれらの記号の特徴について考え、整理した。

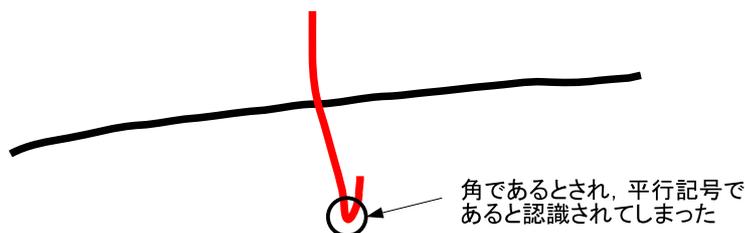


図 4.1: マーキングジェスチャ誤認識の例

- 等長 : 1. 一本の直線である (角がない)  
2. 辺と直線の交点が 1 つである
- 平行 : 1. 二本の直線で形成されている (角が 1 つである ( $P_2$ ))  
2. 二本の直線の長さ  $L_1, L_2$  において,  $L_1 < L_2$  のとき  $2 \times L_1 > L_2$  となる  
3. 角の点 ( $P_2$ ) と, 図形の辺との距離  $L_3$  において,  $L_3 < L_1/2$  となる  
4. 辺と二本の直線との交点 ( $P_4$ ) が, 合計 1 つである

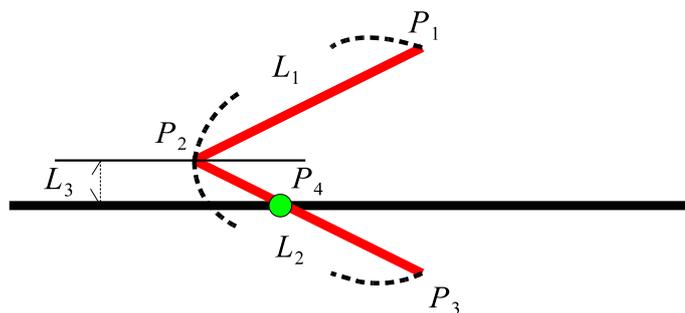


図 4.2: 平行記号の認識

- 直角 : 1. 二本の直線で形成されている (角が 1 つである ( $P_2$ ))  
2. 二本の直線の長さ  $L_1, L_2$  において,  $L_1 < L_2$  のとき  $2 \times L_1 > L_2$  となる  
3. 二本の直線が, それぞれ隣り合った辺と交差している。  
4. 直線との交点が, 一本の直線に 1 つずつである ( $P_1, P_3$ )

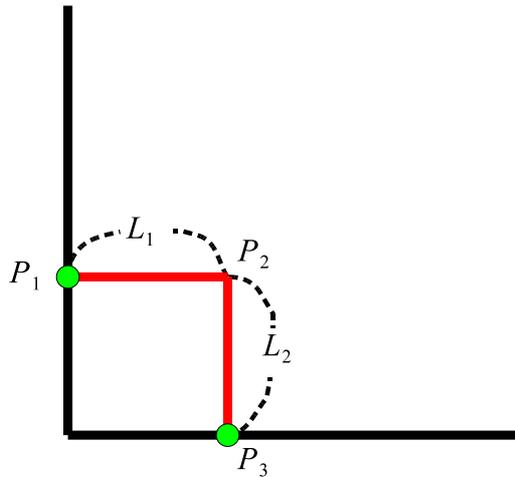


図 4.3: 直角記号の認識

記号それぞれに，上記の条件を加えた．ここでの注目点は，辺と記号との交わり方である．平行記号においては，「向き」が重要で，図 4.4 のように描いた記号を平行記号とは言わないために，これを除去しなければならない．直角記号においては，隣り合う 2 つの辺と，直角記号の二本の直線がそれぞれ交わる．これらの条件を加えることで，認識率の向上を図ることができる．

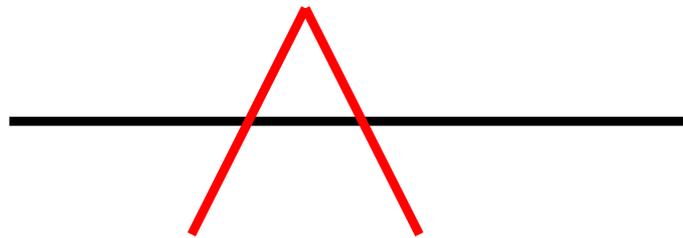


図 4.4: 平行記号の向きの例

また，図形を描いたストロークなのか，マーキングジェスチャを行ったストロークなのかの区別をしなければならない．そこで，マウスの右クリックに相当するスタイラスのパレルボタンを利用し，パレルボタンを押しながら描いたストロークはマーキングジェスチャであるとした．スタイラスのパレルボタンを押しながら描くことは，パレルボタンが押しやすい位置にあるために利用者の負担になるとは考えにくい．さらに，誤認識を防ぐ効果もあると考えられる．

## 4.2 フリーハンド図形の変形

フリーハンド図形の変形の実装について説明する．図形の中でも非常に使用頻度の高い四角形において実現を試みた．

マーキングジェスチャとして記号が描かれる度に，描かれている四角形のどの辺と交わっているかを調べる．交わっている辺が存在すれば，その辺に対する指示であるとし，認識されたマーキングジェスチャの通りに，辺に状態を記録する．変形において，底辺以外の3辺を優先して変形することとし，変形の基本的な基準は底辺としている．

また，記号がどのように描かれるかによって四角形の状態をパターンに分け，記号が描かれる度に四角形の状態を更新していく．状態によって次になることのできる状態が限定される．四角形に対し，このような監視を続ける．

### 4.2.1 等長

マーキングジェスチャとして描かれた記号が等長記号であった場合の変形を図 4.6 に示し，図 4.6 を例に実装の説明を行う．

動作の手順としては，以下の3つの段階に分かれている．

第一段階：辺  $P_1-P_2$  の変形

1. (a) のようなマーキングジェスチャの描かれていない四角形  $P_1P_2P_3P_4$  に対し，(b) のように辺  $P_2-P_3$  に等長記号を描くと，長さ  $L_0$  が記録される
2. 次に (c) のように，辺  $P_1-P_2$  に等長記号を描くと，変形を行うために  $L_1, L_2, L_3$  が記録される
3. 辺  $P_1-P_2$  の点列全てに対し， $P_2$  からの距離が  $L_0/L_1$  倍となるように平行移動させる
4.  $L_0 = L_4$  となり，辺  $P_1-P_2$  の変形が完了する

第二段階：辺  $P_1-P_4$  の変形

1.  $P_1, P_4, P'_1$  から  $\theta$  を求める
2.  $P'_1$  から  $P_4$  の距離  $L_6$  を求める
3. 辺  $P_1-P_4$  の点列全てに対し， $P_4$  を中心に  $\theta$  だけ回転移動させる
4. 回転移動させた点列に対し， $P_4$  からの距離が  $L_6/L_3$  倍になるように平行移動させる
5. 辺  $P_1-P_4$  の変形が完了する

### 第三段階：等長記号の移動

1.  $proportion = L_2/L_1$  とする
2. 辺  $P_1-P_2$  に描いた等長記号の点列全てに対し,  $P_2$  からの距離が  $proportion$  倍になるように平行移動させる
3.  $L_5/L_2 = proportion$  となり, 等長記号の移動が完了する

第一段階で, 等長記号が描かれた辺同士の長さを同じにする. 辺の長さを同じにしたことによって, 等長記号が描かれていない辺  $P_1-P_4$  も変形しなければならない. そこで, 第二段階で辺  $P_1-P_4$  の変形を行う. 以上で辺の移動は完了であるが, この状態ではマーキングジェスチャとして描いた記号がそのままの位置に残ってしまい, 図 4.5 のような状態になってしまう場合がある. そこで, 第三段階で等長記号の移動を行う.

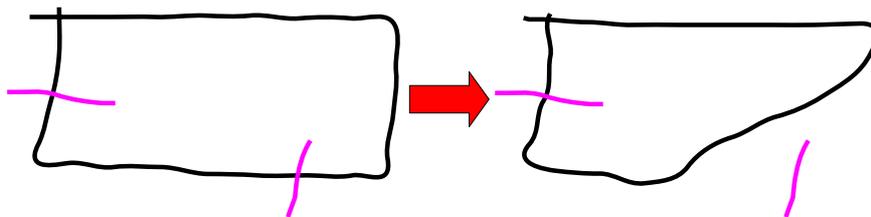


図 4.5: 等長記号の移動を行わずに発生した問題の例

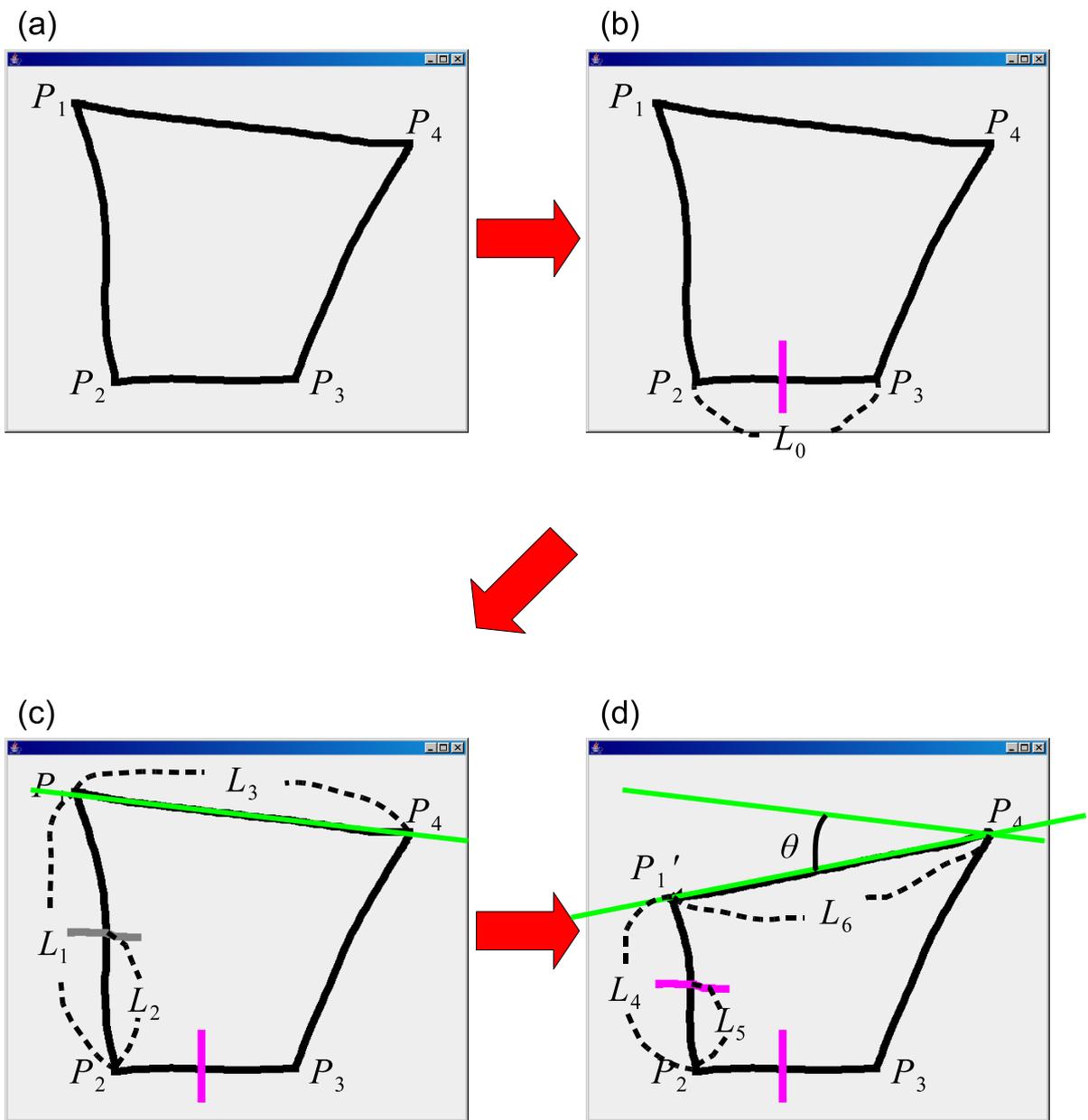


図 4.6: 等長記号による図形の変形

## 4.2.2 平行

マーキングジェスチャとして描かれた記号が平行記号であった場合の変形を図 4.7 に示し、図 4.7 を例に実装の説明を行う。

動作の手順としては、3つの段階に分かれる。

第一段階：辺  $P_1-P_4$  の変形

1. (a) のようなマーキングジェスチャの描かれていない四角形  $P_1P_2P_3P_4$  に対し、(b) のように辺  $P_2-P_3$  に平行記号を描くと、辺  $P_2-P_3$  が変形の基準となる
2. 次に(c)のように、辺  $P_1-P_4$  に平行記号が描かれると、辺  $P_2-P_3$  と平行な直線と辺  $P_1-P_4$  の角度  $\theta_1$  を求める
3. 辺  $P_1-P_2$  の点列全てに対し、 $P_1$  を中心に  $\theta_1$  だけ回転移動させる
4. 辺  $P_1-P'_4$  と辺  $P_2-P_3$  が平行となり、辺  $P_1-P_4$  の変形が完了する

第二段階：辺  $P_3-P_4$  の変形

1.  $P_4, P_3, P'_4$  から  $\theta_2$  を求める
2.  $P_3$  から  $P_4$  の距離  $L_1$  を求める
3. 辺  $P_3-P_4$  の点列全てに対し、 $P_3$  を中心に  $\theta_2$  だけ回転移動させる
4. 回転移動させた点列に対し、 $P_3$  からの距離が  $L_1/L_2$  倍になるように平行移動させる
5. 辺  $P_3-P_4$  の変形が完了する

第三段階：平行記号の移動

1. 辺  $P_1-P_4$  に描いた等長記号の点列全てに対し、 $P_1$  を中心に  $\theta_1$  だけ回転移動させる
2. 等長記号の移動が完了する

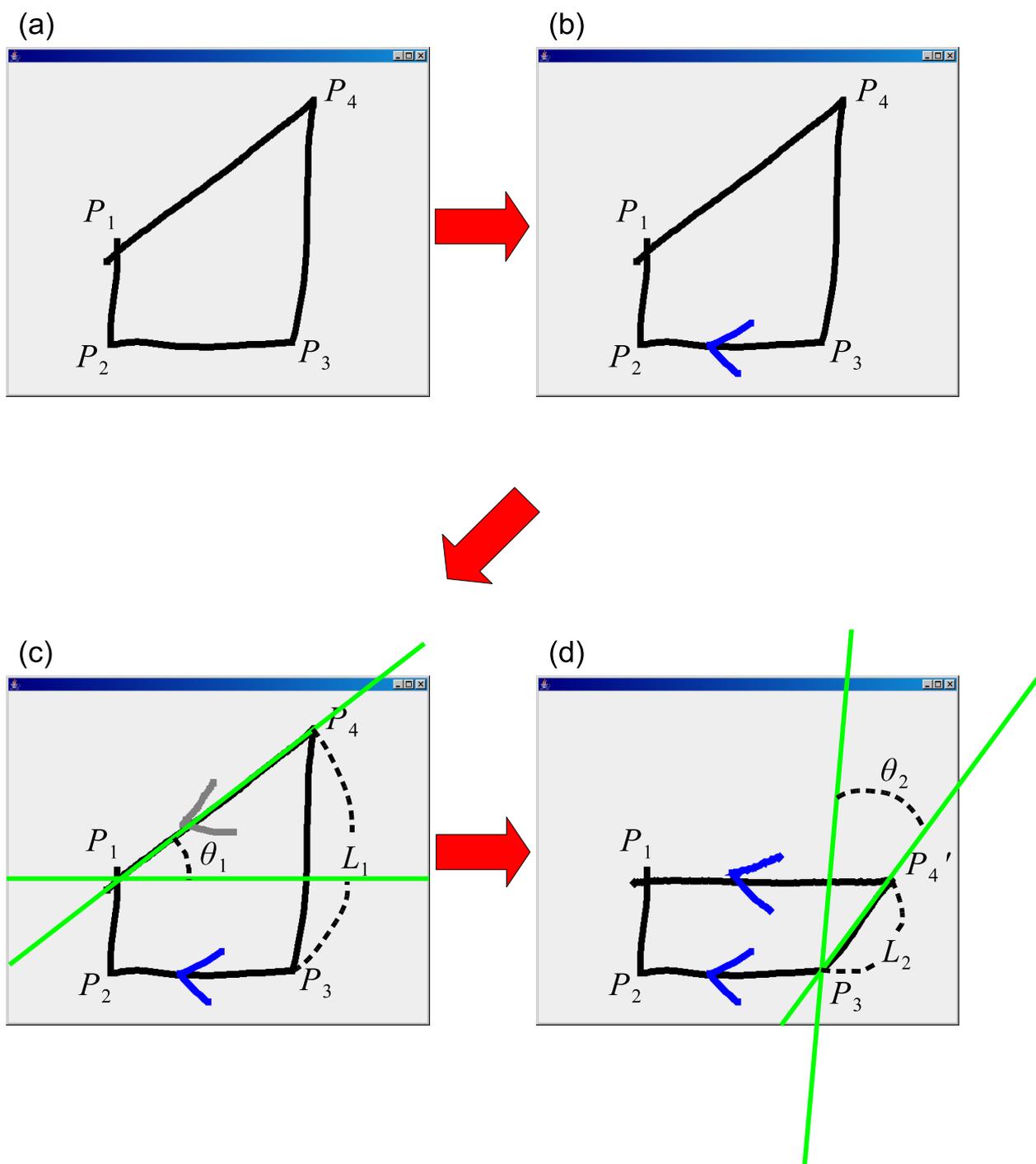


図 4.7: 平行記号による図形の変形

四角形に平行記号描く場合，向かい合った辺同士にしか平行記号を描くことができない．そこで，向かい合った辺同士以外に平行記号を描いた場合は，自動的に記号を消去し描けないようにしている（図 4.8）．

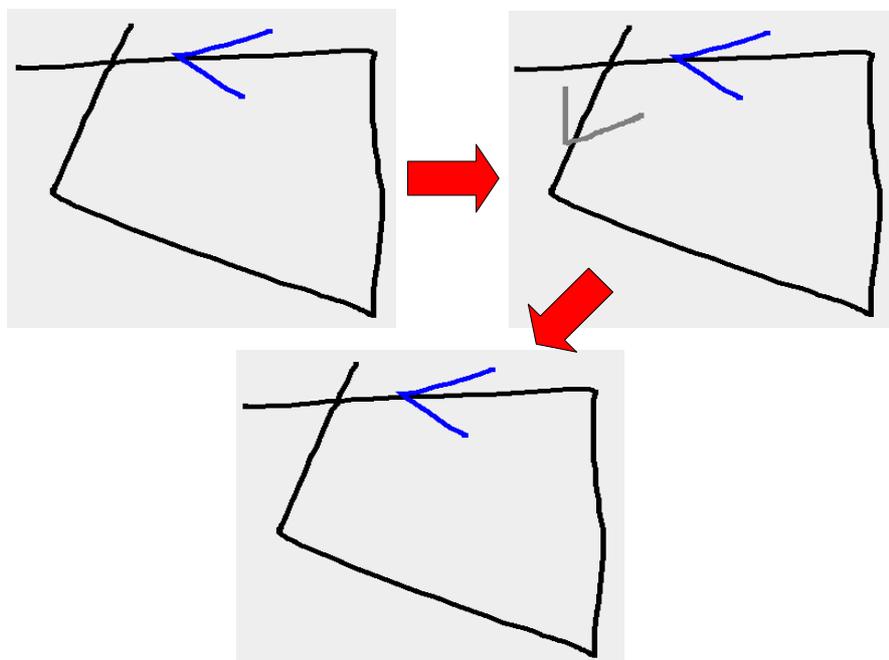


図 4.8: 自動的に消去された平行記号

### 4.2.3 直角

マーキングジェスチャとして描かれた記号が直角記号であった場合の変形を図 4.9 に示し，図 4.9 を例に実装の説明を行う．

動作の手順を以下に示す．

1. 角  $P_2$  に対し直角記号を描くと，底辺が優先的に基準となるので底辺が変形の基準となる
2. 底辺と直角の直線と，辺  $P_1-P_2$  との角度  $\theta_1$  を求める
3.  $P_1$  を  $P_2$  を中心に  $\theta_1$  だけ回転移動した座標を  $P'_1$  とする
4.  $P_1, P_4, P'_1$  から  $\theta_2$  を求める
5. 辺  $P_1-P_2$  の点列全てを， $P_2$  を中心に  $\theta_1$  だけ回転移動する
6. 辺  $P_1-P_4$  の点列全てを， $P_4$  を中心に  $\theta_2$  だけ回転移動する

7. 回転移動した辺  $P_1-P_4$  の点列全てを,  $P_4$  からの距離が  $L_2/L_1$  倍となるように平行移動する

8. 辺  $P_1-P_2$  と辺  $P_2-P_3$  が直角となり, 変形が完了する

記号が描かれた辺は  $P_1-P_2$  と  $P_1-P_4$  である. それに従って辺  $P_1-P_2$  の移動を行う. そこで変形を終了すると, 辺  $P_1-P_4$  の点列がそのままになってしまい四角形の形が崩れてしまう. そこで, 6の動作を行うことで正確な変形になる.

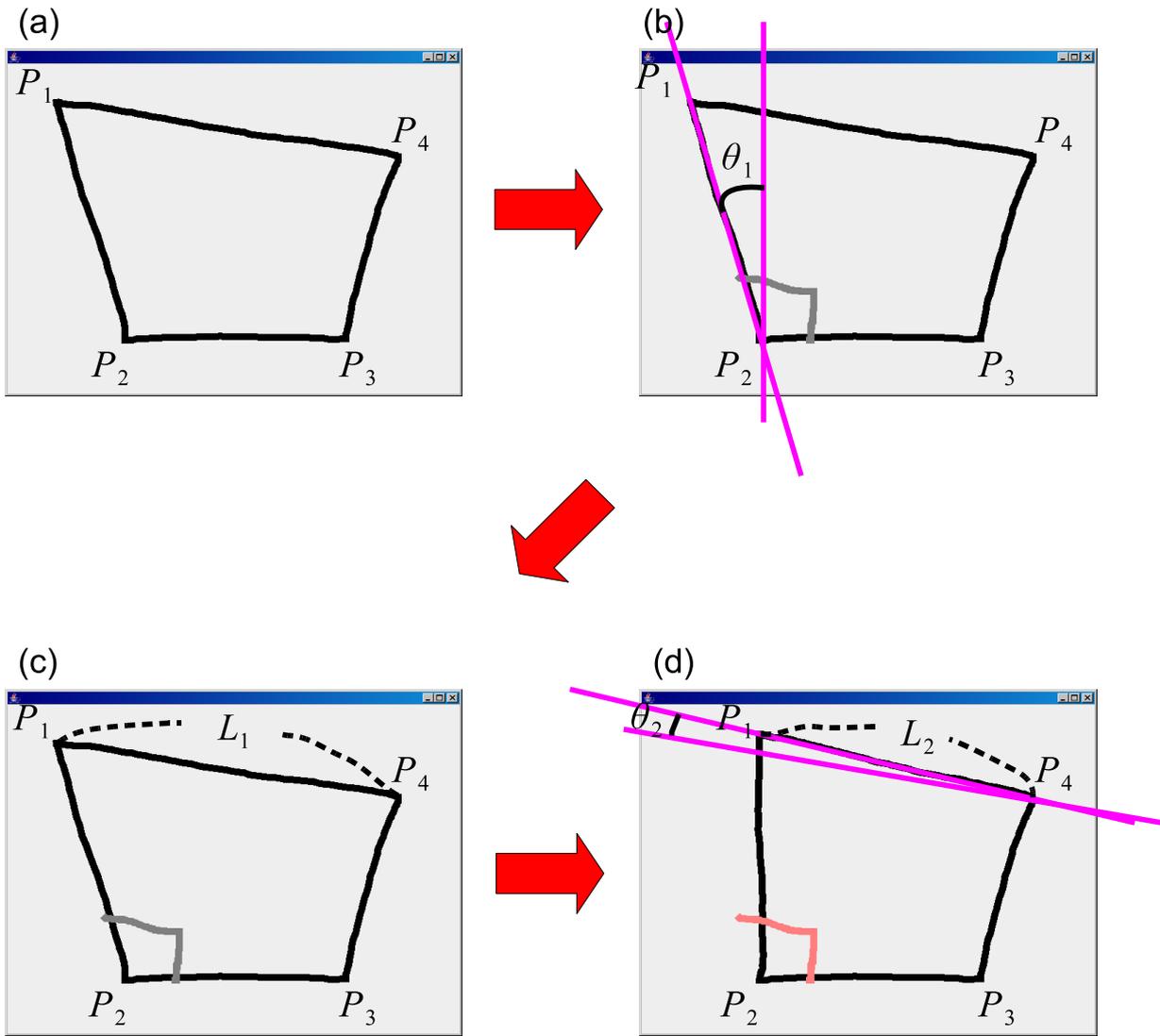


図 4.9: 直角記号による図形の変形

## 4.3 状態の保存・解除

### 4.3.1 保存

マーキングジェスチャにより，辺の状態が決定されて図形の変形が行われる．マーキングジェスチャによる記号は，通常描かれたまま残る．記号が描かれた状態のまま，また別の記号を描くと辺の状態が保存されたまま図形の変形が起こるようになっている．実行例を図 4.10 に示し，例と共に実装について述べる．

動作の手順を以下に示す．

1. (a) で，辺  $P_1-P_2$  と辺  $P_2-P_3$  の長さが，等長記号によって揃っている
2. 次に辺  $P_3-P_4$  に描くことで，(b) のように長さが揃う
3. 上記の 3 辺が揃っている状態で，辺  $P_1-P_4$  に等長記号を描くと辺  $P_1-P_4$  の長さが同じになるだけでなく，辺  $P_3-P_4$  の長さが保たれたまま (c) のように変形され，ひし形になる．
4. 全ての辺が揃った状態で，角  $P_2$  に直角記号を描くと，辺の長さが揃ったまま直角になるので (d) のように正方形になる．

3 について詳しく述べる．3 では記号が描かれた後，角  $P_1P_3P_2$  の角度を求め， $\theta_1$  とおく． $\theta_2 = 2 \times \theta_1$  とする．角  $P_2$  を  $\theta_2$  だけ回転移動した点を  $P_4''$  とする． $P_4'$ ， $P_3$ ， $P_4''$  から  $\theta_3$  を求める．辺  $P_3-P_4$  の点列全てに対し， $\theta_3$  だけ回転移動させる．回転移動前の辺  $P_1-P_4'$  の長さを  $L_1$ ，回転移動後の辺  $P_1-P_4''$  の長さを  $L_2$  とする．辺  $P_1-P_4''$  に対し，回転移動を行い， $P_1$  からの長さが  $L_1/L_2$  倍になるように平行移動させる．これで (c) の状態になる．

4 について述べる．4 では直角記号を描くことでまず，辺  $P_1-P_2$  を回転移動させ，同様に辺  $P_3-P_4''$  も回転移動させる．最後に辺  $P_1-P_4''$  に対し平行させることで，(d) の状態になる．

本研究における実装では，四角形の状態を常に監視するようにしている．そのため，(b) の状態から，残りの辺に等長記号を描いた場合は，ひし形になると決められており，また (c) の状態からどこかの角に直角記号を描いた場合は正方形になると決められている．故に，(c) から (d) への変形は，どの角に直角記号を描いたとしても，一つのアルゴリズムで変形することが可能となっている．

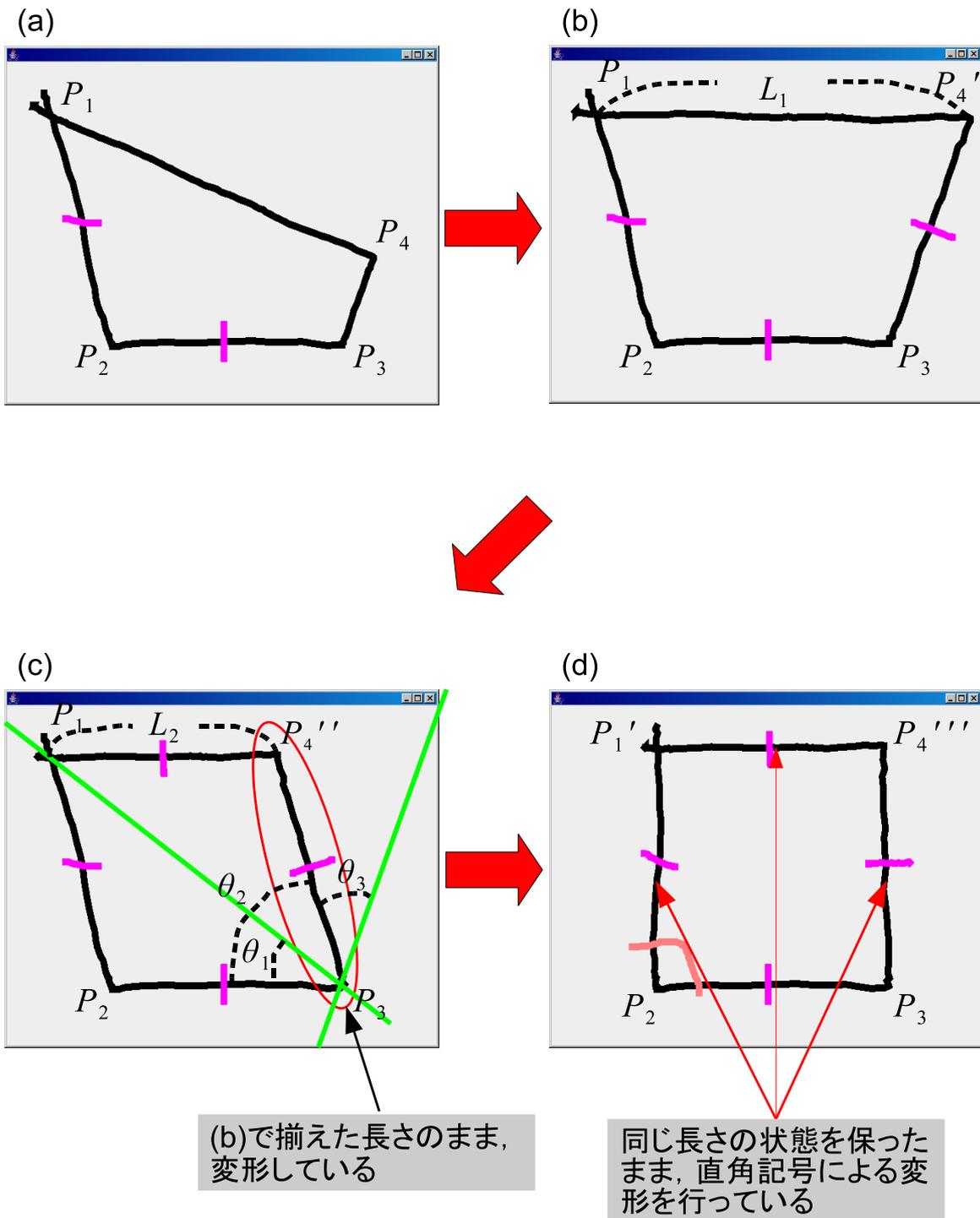


図 4.10: 辺の状態が保存された変形

### 4.3.2 解除

マーキングジェスチャにより描かれた記号は、右クリックをダブルクリック（スタイラスではバレルボタンを押しながら2回タップ）することで、記号を消去することができる。記号が消去されると、状態が解除され次に記号が描かれた場合は、その記号の示す状態に変形する。右クリックのダブルクリックが行われると、四角形のストロークは残されたまま、記号のストローク情報が全て消去される。また、四角形の状態を初期の状態に戻すことで、状態の解除を行っている。動作の例を図4.11に示す。

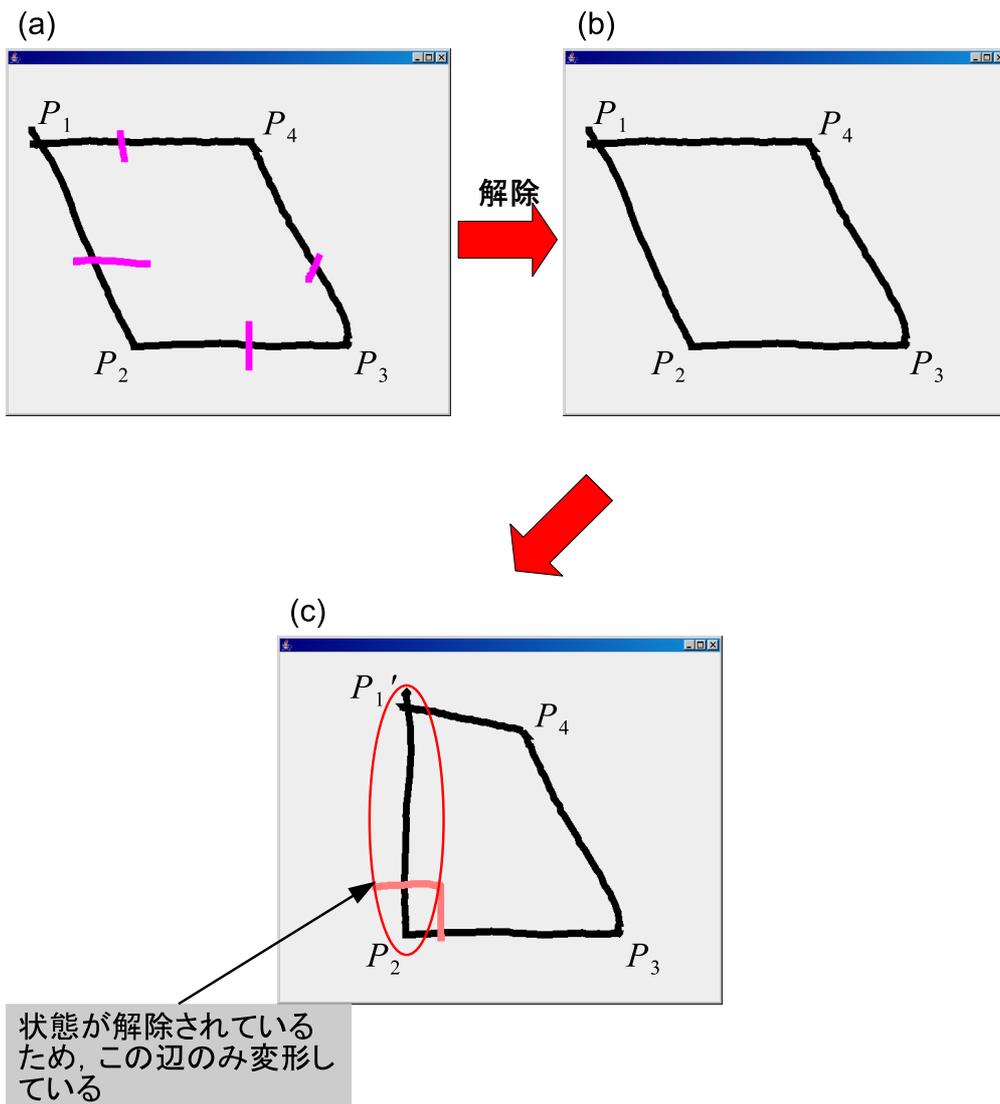


図 4.11: 辺の状態が解除された変形

## 第5章 考察と今後の課題

### 5.1 考察

#### 5.1.1 マーキングジェスチャの認識

本論文では3つのマーキングジェスチャの記号の認識を行った。どの記号であるかという条件を厳しくすることで、記号でないものを記号であるといった誤認識は減らすことができる。しかし、本研究ではフリーハンドで描いた記号をフリーハンドのまま残すので、条件を厳しくすることは利用者にとってフラストレーションとなってしまう場合が考えられる。また、創造作業において必要であると述べた、[簡単]、[柔軟]の2つを満たせなくなってしまう。そのため、条件の設定に問題があった。

#### 5.1.2 フリーハンド図形の変形

フリーハンドで図形を描く場合、図形の向きや大きさ、書き順は人によって、また時によって様々である。特に向きと書き順において自由に描かれた図形のまま変形を行うためには、回転移動を行う際の移動する方向などにおいて、細かく条件を設定しなくてはならない。そのため、非常に困難であった。また、変形を行うときには基準が必要である。例えば等長記号であれば、どの辺を基準にするのか、どの角を基準にするのかといったことを決定しなければならない。

### 5.2 今後の課題

まず、扱うことのできる図形・マーキングジェスチャの記号を増やすことである。次に、底辺を水平にしなくても変形が行えるようにする。そして評価実験を行い、マーキングジェスチャの認識の精度は利用者にとってどうか、図形の変形の基準はどうかといったことを調査する。また、マーキングジェスチャの記号が残ることを利用し、初期段階の次のステップに生かすことである。次のステップに進んだときに、認識されている記号の通りに図形を綺麗に整形されれば、初期段階から清書までを網羅することができるようになる。

## 第6章 関連研究

ペン入力指向図形整形インタフェース [12] は、マーキングジェスチャ (MG: Marking Gestures) という対話技法を定義し、詳細設計を述べている。実装は三角形のみで行っている。描かれた三角形を完全に整形する。三角形の底辺は必ず水平とし、実装を簡単にしている。三角形を、正三角形、角度固定二等辺三角形、二等辺三角形、3角度固定、2辺長比固定、2辺長比1角度固定、2角度固定、2辺長比固定、1角度固定、条件なしのどの状態にあるかを常に監視し、マーキングジェスチャが行われるたびに条件が適用可能かを判別し、可能であれば整形するという手順を繰り返すようにしている。この研究は、算数の授業で利用することを前提とし、教育効果の向上を目的としている。そのため、描かれた図形、マーキングジェスチャは全て整形している。しかし本研究は、マーキングジェスチャを利用している点に関しては同様であるが、創造作業の支援を目的としており、描かれた図形、マーキングジェスチャ共にフリーハンドのストロークを保ったまま変形を行っている。そこが大きく異なる部分である。

### 6.1 創造作業の支援に関する研究

序論で既に述べた文献 [4][5] 以外にはまず DENIM [1] がある。DENIM はスケッチ画を利用して Web サイトのデザインを行うシステムである。Tivoli [14] は Xerox Liveboard 上で動かすことをターゲットとし、インフォーマルなワークグループのミーティングを支援するためにデザインされた電子ホワイトボードシステムである。Cocktail Napkin [15][16] は、手書きスケッチのデザインを支援することを目的に作られたペンベース描画環境である。Misue らによる研究 [17] は、創造作業を支援することを目的に試作ツールを開発している。試作ツールでは、手書き図の論理構造をツールが自動的に解釈し、構造を保存したままオブジェクトの移動ができる。

### 6.2 ペン指向作図インタフェースに関する研究

Kurtenbach らによる研究 [8] では、図形に対し横線を引いたり、丸で囲むなどをすることで、オブジェクトの削除や移動ができる。Baudel による研究 [9] では、自由曲線の整形手法について研究されている。五十嵐らの Pegasus [11] では、算数や理科の授業で出てくるような図の板所を補助するものとして、手書きで描かれたストロークを、平行や直角などの幾何学的制約に基づきいくつか候補を挙げるなどして整形・清書するシステムである。また、Sutherland

による SketchPad[18] では，入力デバイスとしてライトペンを採用し，整形したい部分と整形の種類を指し示し，線分を平行にしたり角度を直角にできるシステムである．

## 第7章 結論

本研究ではまず，創造作業の初期段階における手書きの良さについて検討し，創造作業の支援を行うツールに対する要求について整理した．そしてペン入力の利点について検討し，研究の目的を述べた．次に，手書き図の編集においてマーキングジェスチャを利用することを提案した．提案に基づき，マーキングジェスチャを利用したフリーハンド作図システムを設計し、Java を用いて実装を行った．本システムでは，マーキングジェスチャとしての記号である等長・平行・直角の記号の認識を行うことができるようになった．結果，利用者はその記号を利用して，フリーハンド図形にマーキングジェスチャを行うことで，フリーハンドのストロークのまま図形の変形を行うことができるようになった．本システムの実現により，ペン入力の良さを生かしてフリーハンド図形の変形が容易に行えるようになった．

## 謝辞

本論文の執筆にあたり、指導教員である田中二郎教授ならびに、三末和男助教授、講師の志築文太郎先生と高橋伸先生には、大変貴重な意見やご指導、激励を頂きました。本論文の執筆に至るまでにも色々とお世話になったことを含め、この場を借りて深く御礼を申し上げたいと思います。また、田中研究室の皆様にも、ゼミ等を通じて多くの意見、アドバイスを頂きました。最後に、自分を支えてくれた両親や、全ての友人にも感謝申し上げたいと思います。本当にありがとうございました。

## 参考文献

- [1] J. Lin, M. W. Newman, J. I. Hong, and J. A. Landay . DENIM : Finding a Tighter Fit Between Tools and Practice for Web Site Design . in *Proceedings of the SIGCHI conference on Human factors in computing systems ( CHI '00 )* , pp.510-517 , 2000 .
- [2] 三末和男 . ペン、スケッチ、フリーハンド図 ~ 創造支援への手書きの利用 . 第二回知識創造支援システムシンポジウム予稿集 , pp. 40-47 , 2005 .
- [3] J. A. Landay and B. A. Myers . Interactive Sketching for the Early Stages of User Interface Design . in *Proceedings of the SIGCHI conference on Human factors in computing systems ( CHI '95 )* , pp.43-50 , 1995 .
- [4] S. P. Zeleznik , K. P. Herndon and J. F. Hughes . SKETCH : An interface for sketching 3D scenes . in *Proceedings of the 23th annual conference on Computer graphics and interactive techniques ( SIGGRAPH '96 )* , pp.163-170 , 1996 .
- [5] T. Igarashi , S. Matsuoka and H. Tanaka . Teddy : A Sketching Interface for 3D Freeform Design . in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques ( SIGGRAPH '99 )* , pp.409-416 , 1999 .
- [6] J. Schumann , T. Strothotte , S. Laser and A. Raab . Assessing the Effect of Nonphotorealistic Rendered Images in CAD . in *Proceedings of the SIGCHI conference on Human factors in computing systems ( CHI '96 )* , pp.35-41 , 1996 .
- [7] 五十嵐健夫 . スケッチインタフェースの研究動向 . コンピュータソフトウェア , Vol.23 , No.4 , pp.3-13 , 2006 .
- [8] G. Kurtenbach and W. Buxton . Issues in combining marking and direct manipulation techniques . in *Proceedings of the 4th annual ACM symposium on User interface software and technology ( UIST '91 )* , pp.185-192 , 1991 .
- [9] T. Baudel . A mark-based interaction Pradigm for free-hand drawing . in *Proceedings of the 7th annual ACM symposium on User interface software and technology ( UIST '94 )* , pp.185-192 , 1994 .
- [10] 松田浩一 , 近藤邦雄 . 手書き図形入力のための時系列情報を利用した逐次清書法 . 情報処理学会論文誌 , Vol.40 , No.2 , pp.594-601 , 1999 .

- [11] 五十嵐健夫, 松岡聡, 河内谷幸子, 田中英彦. 対話的整形による幾何学的図形の高速度描画. 情報処理学会論文誌, Vol.39, No.5, pp.1373-1384, 1998.
- [12] 加藤直樹, 中川正樹. ペン入力指向図形整形インタフェース. 情報処理学会第63回全国大会講演論文集, Vol.4, pp.31-32, 2001.
- [13] D. Rubine. Specifying gestures by example. in *Proceedings of the 18th annual conference on Computer graphics and interactive techniques (SIGGRAPH '91)*, pp.329-337, 1991.
- [14] E. R. Pedersen, K. McCall, T. P. Moran and F. G. Halasz. Tivoli: an electronic whiteboard for informal workgroup meetings. in *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '93)*, pp.391-398, 1993.
- [15] M. D. Gross and E. Y.-L. Do. Ambiguous Intentions: a Paper-like Interface for Creative Design. in *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)*, pp.183-192, 1996.
- [16] M. A. Hearst, M. D. Gross, J. A. Landay and T. E. Stahovich. Sketching Intelligent Systems. in *Proceedings of IEEE Intelligent Systems*, Vol.13, No.3, pp.10-19, 1998.
- [17] K. Misue and J. Tanaka. A Handwriting Tool to Support Creative Activities. in *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES '05)*, pp.423-429, 2005.
- [18] I. E. Sutherland. SketchPad a man-machine graphical communication system. in *Proceedings of the SHARE design automation workshop (DAC '64)*, pp.6329-6346, 1964.