

キーボードとペンを併用するノートエディタの研究

辰川 肇, Nigel WARD
東京大学大学院 工学系研究科
〒113-8656 東京都文京区本郷 7-3-1
03-5841-6282
E-Mail: tatukawa@sanpo.t.u-tokyo.ac.jp

概要 パソコンが普及し、情報教育が盛んになった現在でも、ノートは未だ手書きで書かれている。これはノートに対する文書作成速度と多様な図形の描画の要求がパソコン上でのノート作成を難しくしているためである。そこで我々は、それらの要求を解決してパソコン上でも手書きと同じようにノートを書くことのできるアプリケーションを開発することを目指し、テキスト入力をキーボード、図形の描画をペンで行う“ノートエディタ”を開発した。

本論文ではノートエディタの概要、文書作成速度を向上させる3つの機能、ノートエディタの総合評価実験について説明する。総合評価実験では4名の被験者が継続して講義ノートを作成し、十分なパソコンスキルを持つ2名のユーザがノートエディタは便利であると評価した。

Towards a Tool for Taking Lecture Notes

Hajime TATSUKAWA, Nigel WARD
University of Tokyo
Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
03-5841-6282
E-Mail: tatukawa@sanpo.t.u-tokyo.ac.jp

Abstract Students still take lecture notes using pencil and paper — although digital documents are more legible, easier to search in and easier to edit — in part because of the lack of software to support note-taking.

Lecture notes are characterized by free spatial organization, many small chunks of text, a dense mix of text and graphic elements, and general sloppiness.

These characteristics imply that a note-taking system should use pen and keyboard; allow the swift entry of text at any desired position; and minimize the need to switch between input tools.

Four students used the system to take notes in class over several months; two rated it preferable to pencil and paper, suggesting that taking lecture-notes with the computer is feasible.

1 序論

情報教育が盛んになりパソコン使用経験の豊富な学生が多くなった現在でも、講義ノートや研究ノートは紙に手書きで書かれている。パソコンを使ってノートを書くことが可能になれば、読み易さや編集・検索し易さ、あるいはインターネットとのリンク付けなど紙のノートに無い様々なメリットをノートに与えることが可能になる。

レポートや論文など既にデジタル化されている文書の多くは他者に読ませることを第一目的としていて、読み易さや編集し易さといった要素が重視される。これに対して、ノートにはより重要度の高い2つの要求がある。

ノートは通常、講義や会議といった時間の制約がある場面で書かれる。研究ノートは書き手の思考速度を妨げない速度で書けることが好まれる。ノートに対する要求の1つ目はノート作成が素早く行えることである。

ノートに対する要求の2つ目は、多様な図形の描画が可能なことである。ノートにはテキストの他に関係線、強調線、矢印、グラフ、絵など様々な図形が書かれる。それらノートの構成要素(本論文では“オブジェクト”と呼ぶことにする)の位置関係はそれぞれの関係性や意味に結びつく重要な情報である [1]。

現在普及している文書作成用アプリケーションのドロー機能は多様な図形に対応しているものの、従来の GUI

の操作などに時間を取られるためにノートに対する2つの要求を同時に解決するには至っていない。

計算機を使ってノートを作成することを目的とした研究は環境構築型と端末型の2種類に分けることができる。

環境構築型とは、特に対象を講義ノートに限定し、電子黒板やマイク、ビデオカメラなど教室に設置される機器と受講者に用意された端末を連携させるというアプローチをとった研究である。[2]は教室全体をシステムとし、画像と音声、電子ホワイトボードを組み合わせたマルチメディアノートを作成する試みを行なっている。[3]はノート作成作業による受講者の負担を軽減するために講義のビデオ映像にペンインタフェースで書き込みをするという手法を提案している。[4]は電子黒板とペン入力PCをネットワークで結んで講義内容の送受信や再利用を可能にしている。環境構築型のノートの利点は紙のノートに記録できない講義中の音声情報や画像情報等を取り込むことができることである。逆にハードウェアのコストが増加してしまう問題点がある。

端末型とは、既存のPCや携帯情報端末を利用してノートを作成しようとする研究である。[5]は携帯情報端末で作成したノートを計算機に集積して複数の学生で共有することを提案している。[6]はペンインタフェースで入力した情報の検索性を向上させる手法を提案している。[7]は蓄積した情報を整理し易いノートの構造について議論している。端末型の研究には作成するノートの構造や利用法に着目したものが多くあるが、学生が普段書いているようなノートを簡単に書けるアプリケーションが実現されているものは少ない。

環境構築型、端末型を問わず、ノート作成を目的とした研究の多くは入力手段としてペンインタフェースを採用している。これはペンを使うことで紙と鉛筆の場合と同様に図形を容易に描画できるためであり、前述したノートへの要求に対する1つの解決策である。しかし、ペンインタフェースのみでノートを作成すると編集し易さや検索し易さ、読み易さといった計算機を用いるメリットが低下し、これを補うことが難しい。

そこで本研究では、テキスト入力にキーボード、図形描画にペンを用いるという役割分担によってノート作成速度と多様な図形の描画という2つの要求を解決するソフトウェア“ノートエディタ”の開発を行った。ノートエディタは一般的なドローソフトを出発点として、ペンストロークで自由曲線を描画できる機能、及びノート作成速度を向上させる3つの機能を実装した。

本稿ではノートエディタの概要とノート作成速度を向上させる各機能について説明し、最後にそれらと統合したソフトウェアの総合評価実験について報告する。

2 ノートエディタの設計

2.1 基本仕様

ノートエディタをウィンドウプログラミング用APIが豊富に用意されているJava言語によるアプリケーションとして開発した(図1参照)。

ハードウェアは講義の行なわれる教室へ自由に持ち運べるタブレットディスプレイ付ノートPC:NEC VersaPro VA50H (CPU celeron500MHz, メモリ 192MB, 表示方式: SVGA800 × 600 ドット, 10.4 型 TFT カラー液晶)を使用した。

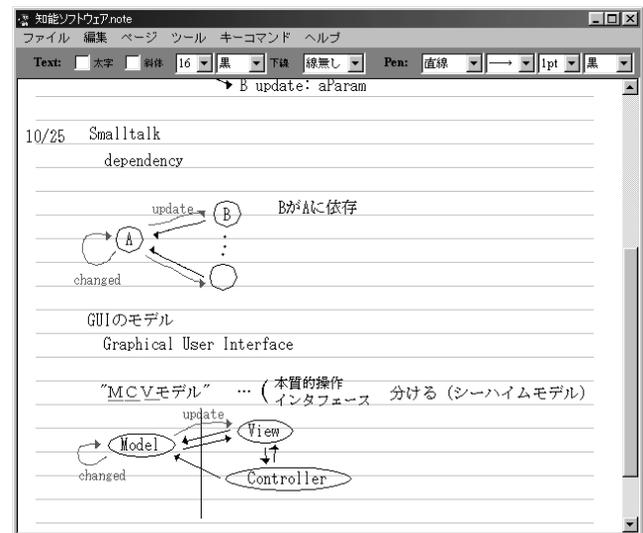


図1: ノートエディタのメイン画面

ノートエディタの出発点は一般的なドローソフトである。これに加えて、ペンストローク(またはマウスドラッグ)で自由曲線を描画できる機能を追加した。図1に書かれている矢印とテキストを囲む楕円は画面に直接ペン入力されたものである。さらにノート作成速度を向上させるテキストボックス自動生成機能、ポップアップキーコマンド機能、パターン描画機能をノートエディタに実装した。これら3つの機能について次節より説明する。

2.2 テキストボックス自動生成機能

テキストはノートに最も多く、あらゆる位置に書かれるオブジェクトである。そのため、自由な位置へのテキスト入力保証されていることはノートエディタにとって必要条件である。

一般的にドローソフトはテキストボックスと呼ばれる機能によって自由な位置へのテキスト入力を可能にしている。製品による差異はあるものの、従来のテキストボックスを用いたテキスト入力には次のような操作を要する:

1. テキストボックスアイコンをクリック
2. 書きたい位置にポインタを移動
3. 書きたい位置をクリック（または領域をドラッグ）
4. テキストをタイプ

これらの操作はノートに書かれるテキストボックスの数だけ繰り返される。そこで、この一連の操作を効率化させる“テキストボックス自動生成機能”を実装した。

この機能は、ノートキャンパス上でキーボードをタイプするとポインタのある位置にテキストボックスが自動的に生成され、そこに文字が追加されて行くというものである。ポインタ位置にテキストボックスが既にある場合には、その末尾に文字が追加される（図2参照）。テキストボックスの幅は改行または行末までの文字数、高さは行数で決まる。

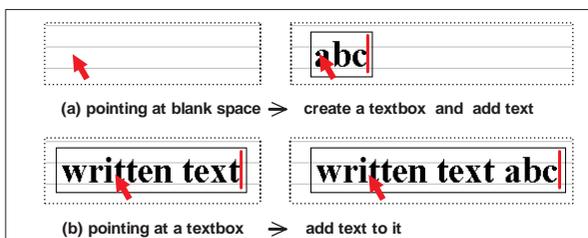


図 2: テキストボックス自動生成機能

これによりテキスト入力に必要な操作は次のようになる。

1. 書きたい位置にポインタを移動
2. キーボードからテキストを入力

この機能を実装したためノートエディタにはテキストボックスやペンストロークのためのアイコンが無い。任意の時点でのタイピングがテキスト入力、ペン入力が入力描画に対応している、キーボードとペンを持ち替えながらノートを書く作業が最適化されている。

2.3 ポップアップキーコマンド機能

フォントサイズやテキストの下線プロパティの変更といったコマンドはテキスト入力作業中に実行される。コマンド実行のためにメニューバーやポップアップメニューなどを用いると、実行の度にキーボードとマウス（またはペン）の間で手を移動させなければならず時間のロスが発生する。キーボードからコマンドを実行できるショートカットキーではキー単位のコマンドの割り当てを行なっているためにユーザが記憶できるコマンド数に限界があるという問題点がある。

この章で提案する“ポップアップキーコマンド機能”は、コマンドのリストを画面に表示させた上で目的の

コマンド名をキーボードでタイプして実行するというものである。コマンドはキー単位ではなく単語単位の割り当てになるためショートカットキーよりも多くのコマンドに対応でき、さらにコマンドの内容をそのまま表すコマンド名を用いることでより多くのコマンドをユーザが記憶できる。

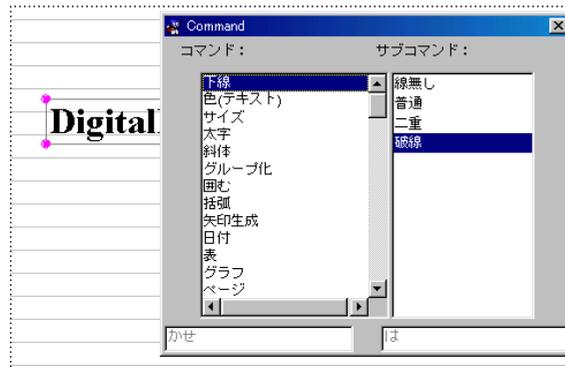


図 3: ポップアップキーコマンド機能

図3は実装したポップアップキーコマンドのウィンドウである。ここではコマンドのリストとそれを修飾するサブコマンドのリストを分けて表示し、コマンドの一覧性を確保している。

コマンドとサブコマンドの組み合わせは、フォントサイズ < 10, 12, 14... >, 下線 < 線無し, 普通, 二重, 破線 >, 色 < 黒, 赤, 青... > などである。

コマンド実行方法は次の通り：

1. *Shift + Enter* によりウィンドウを起動
2. 実行したいコマンドをタイプして選択
3. *Tab* でサブコマンドリストに移動し、2. と同様にサブコマンドを選択する
4. *Enter* で実行, *Esc* でキャンセル

コマンドとサブコマンドは補完機能があり、コマンド名が一意に決まる文字数までタイプすれば選択できる。また、ユーザがサブコマンドを記憶している場合は、ウィンドウ起動後に直接サブコマンドをタイプすることで即座にそれを選択・実行できる。

コマンド実行手段別の操作時間

メニューバー、ポップアップメニュー、ショートカットキーとポップアップキーコマンドの4種類のコマンド実行手段についてコマンド操作にかかる時間を測定・評価する実験システムを作成した。実験システムは39個ずつの全く同じメニューアイテム（またはコマンド）で構成されるメニューバーとポップアップメニュー、ポップアップキーコマンドを持つウィンドウアプリケーションである。39のコマンドの内22にショートカットキー

を割り当てている．これを用い、タイピング上級者 5 名を被験者として次のような実験を行なった．

1. 4 種類の実行手段（メニューバー、ポップアップメニュー、ショートカットキー、ポップアップキーコマンド）に関し 2～5 の手順を 30 回ずつ繰り返す
2. 画面に実行すべきランダムなコマンドと、タイピング用文字列が表示される
3. 表示された文字列をタイピングする
4. 表示されたコマンドを実行する
5. タイピング終了時点からコマンド実行までの時間を測定する

ポップアップキーコマンドはテキスト入力中のコマンド実行を目的とする機能のため、手順 2 で被験者の手をキーボードの上に戻す作業を設定した．

測定結果の 30 ずつのデータ列の標準偏差の被験者平均とコマンドを間違えて実行した延べ失敗数、データ列の中央値の被験者平均とその信頼係数 95 % の信頼区間を実行手段別に表 1 にまとめた．実行手段の略字はポップアップキーコマンド (KC)、ショートカットキー (SC)、ポップアップメニュー (PM)、メニューバー (MB) である．

実行手段	標準偏差 [s]	延べ失敗数	中央値 [s]
KC	0.89	1	2.12 ± 0.40
SC	1.00	9	1.36 ± 0.41
PM	0.48	2	3.08 ± 0.27
MB	0.43	1	2.63 ± 0.15

表 1: 実行手段別の集計値

ショートカットキーのばらつきが大きいのは被験者がショートカットキーを思い出すのに手間取った場面があったためである．ショートカットキーは失敗数も多く、種類が増えると記憶が難しいために失敗しやすい手段であることが現れている．

ポップアップキーコマンドもメニューバーやポップアップメニューと比較すると操作時間のばらつきは大きい．これはコマンドのタイプミスでバックスペースで消すなどの操作に時間を取られる場面があったことが原因である．タイプミスで実行に手間取る可能性のあることがポップアップキーコマンドの欠点の一つであることが分かる．ただ、ポップアップキーコマンドでは選択されるコマンドがウィンドウに明示されるためコマンドを間違えて実行してしまうことは少ない．

中央値の被験者平均によるとポップアップキーコマンドの操作時間はショートカットキーに次ぐ値を示している．コマンド 1 つ当たりの操作時間においてポップアッ

プメニューと比較して 0.96 秒、メニューバーと比較して 0.51 秒短縮できている．

この実験結果から、テキスト入力中に実行するコマンドのうち使用頻度の高いコマンドをショートカットキーで行い、それ以外のコマンドにポップアップキーコマンド機能に対応させることでノート作成作業を向上させることができると言える．

ノートエディタでは切り取りや貼り付け、テキストボックスの移動などにショートカットキーを割り当て、その他のコマンドの大部分はポップアップキーコマンドとメニューバーのどちらでも実行できるよう実装している．

2.4 パターン描画機能

ノートに描かれる多様なオブジェクトにも、繰り返し描かれるパターンが存在する．関係線や境界線などの単純図形は比較的一定の形状で描かれているし、特定の講義ノートでは同じような図が繰り返し使用されることが少なくない．

ここで提案するパターン描画機能はノートに繰り返し現れるパターンを 1 つのコマンドに集約して簡単に描画することを可能にする．この機能がオブジェクトの単純な挿入と異なる点は、パターンが呼び出されたときにシステムがパターンを描画する位置と大きさを自動的に決定できることである．さらに、パターンを統一的に扱えるアルゴリズムによってユーザがユーザ自身の必要性に応じてパターンを簡単に定義することも可能にしている．以下、ユーザあるいはプログラマによって定義されるひとまとまりのオブジェクトの集合をパターンと呼ぶ．

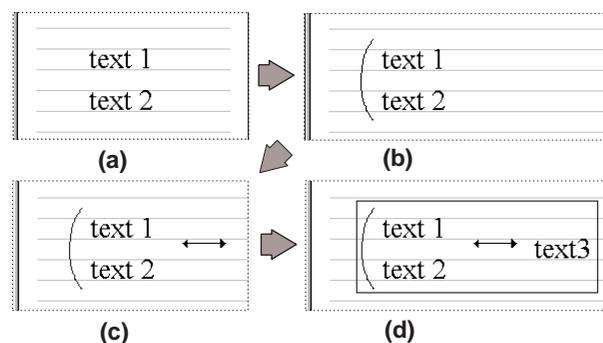


図 4: パターン描画実行の様子

図 4 はパターン描画を実行した例である．ノートに書いた 2 つのテキスト (a) に対して“左括弧”というパターンを描画するとそれらまとめる左括弧が生成される (b)．さらに“左右向き矢印”というパターンを描画すると右側に左右を向いた矢印が生成される (c)．そして矢印の隣に 3 つめのテキストを入力し、それまで生成したオブ

ジェクトに対して“ 長方形 ”というパターンを描画するとそれらを囲む長方形が生成される。

これらのプロセスにおいてユーザは描画されるパターンを移動したり変形したりしなくてよい。パターン描画機能はコマンド実行時に選択されていたオブジェクト(以下ターゲットオブジェクト)に合わせてパターンの生成位置とサイズを決定する。

位置特性と大きさ特性

パターン描画では、全てのパターンに位置特性と大きさ特性という2つのパラメータを与える。位置特性とはターゲットオブジェクトに対してどの位置にパターンを生成するかを決めるパラメータであり、大きさ特性とはターゲットオブジェクトの X-Y 方向のどちらのサイズに合わせてパターンのサイズを決定するかを決めるパラメータである。それぞれのパラメータは次のような値を取る。

位置特性： 上, 下, 左, 右, 右上, 左上, 右下, 左下, 重ねる, 自由

大きさ特性： 両方向依存, 水平方向依存, 垂直方向依存, 独立

図4で例として挙げた“ 左括弧 ”の位置特性は「左」、大きさ特性は「垂直方向依存」、 “ 左右向き矢印 ”の位置特性は「右」、大きさ特性は「独立」、 “ 長方形 ”の位置特性は「重ねる」、大きさ特性は「両方向依存」である。

大きさ特性が「水平方向依存」または「垂直方向依存」、 「独立」のパターンは生成時にパターンを構成するオブジェクトの形が必ず保存される。大きさ特性が「両方向依存」のパターンはオブジェクトの形が歪められる可能性がある。

パターン登録

図5は、ノートエディタに実装しているパターン登録画面である。この画面でユーザはノートエディタに独自のパターンを登録することができる。

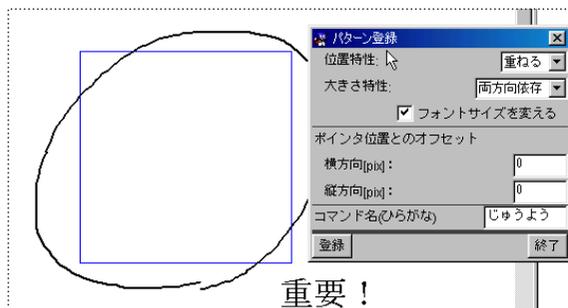


図 5: パターン登録画面

パターンを登録する手順は以下の通り：

- ・仮想的なターゲットである長方形(図中左)に合わせて登録したいパターンを描画する
- ・位置特性と大きさ特性をリストから選択する
- ・パターンを呼び出すコマンド名を定義する

図5では、ユーザはターゲットを囲む丸と“ 重要!”と書かれたテキストボックスを一つのパターンとし、「じゅうよう」というコマンド名を付けて登録している。登録したコマンドはポップアップキーコマンド機能を用いて実行できる「じゅうよう」を実行する様子を図6に示した。

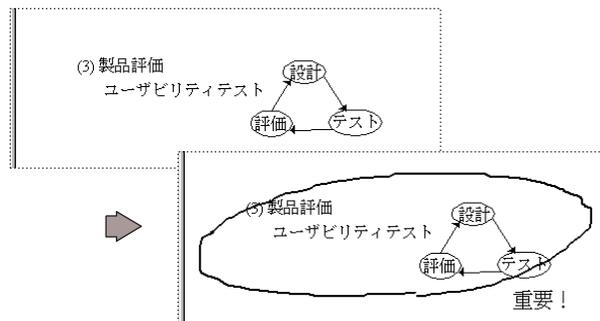


図 6: 登録したパターンの実行

数式への応用

数式は理工学分野のノートによく書かれるオブジェクトである。しかし、ノートエディタに使用するノートPCのタブレットディスプレイは分解能の点で紙と鉛筆に劣っていて細かい数式がきれいに書きづらいという問題点がある。

数式描画のために開発されたツールは存在する。Microsoft Word の Equation Editor[8] は数式をテキストと図形の組合せとして扱い、対話的に数式を描画できるようになっている。しかし、このソフトウェアでは数学記号の選択をマウスで、数字や文字の入力をキーボードで行うため、入力手段の切替えに時間を要する。ノートエディタではポップアップキーコマンド機能によってパターンをキーボードから呼び出すことが可能なため、パターン描画機能を応用してより速くきれいに数式を描画できることが期待できる。

ノートエディタに実装した数式パターンの例を図7に3つ示した。

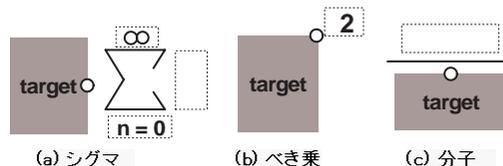


図 7: 数式パターンの例

「シグマ」(図7(a))の位置特性は右、大きさ特性は垂直方向依存である。オブジェクト構成は Σ 記号(ストローク)と上下及び右のテキストボックスである。右のテキストボックスは初期値では空になっている。これらのテキストボックスはパターン生成後、テキストボックス自動生成機能を使って編集できる。

「べき乗」の位置特性は右上、大きさ特性は垂直方向依存。初期値で“2”の入ったテキストボックス1つで構成される。「分子」の位置特性は上、大きさ特性は水平方向依存。パターンは分母と分子の区切り線と空のテキストボックス1つで構成される。

「シグマ」と「べき乗」の2つのパターンを用いて式(1)を描画する手順を図8に示した。

$$f(x) = \sum_{m=1}^{\infty} (m+x)^3 \quad (1)$$

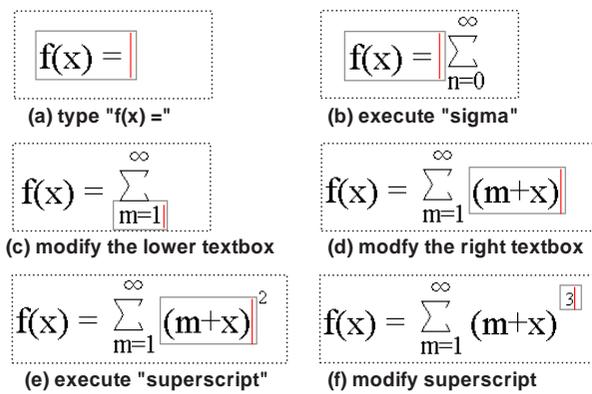


図8: パターン描画を利用して数式を描画する手順

図9は後述する長期使用実験において被験者が実際に作成したノートの一部である。パターン描画を利用することでフォントサイズと同程度の大きさの読み易い数式を描画できているのが分かる。

$$\text{入力 } x(t) : \text{連続 } X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt$$

$$\text{入力 } x(nT) : \text{離散 } X(f) = \sum_{n=-\infty}^{\infty} x(nT) e^{-i2\pi fnT}$$
 ここで $Z = e^{i2\pi fT}$ ($= e^{i\omega T}$) とおいて、又 $x(nT) = X(Z) = \sum_{n} x(n) Z^{-n}$ $X(n)$ の Z 変換

(注) $\sum_{n=0}^{\infty}$ 片側 Z 変換 $\sum_{n=-\infty}^{\infty}$ 両側 Z 変換

図9: 数式パターンの実用例

3 総合評価実験

被験者 10 名の短時間使用実験と、被験者 4 名の長期使用実験を行なった。

3.1 短時間使用実験

被験者は PC 使用歴 4 年以上のタイピング上級者 10 名。ノートエディタ使用経験は全員 1 時間未満。実験前にノートエディタの機能説明と使い慣れるための時間を 30 分設定した。講義ノートと研究ノートに関する以下の 2 つのタスクを行ってもらった：

講義ノート： 4 種類の講義 (文系 1, 情報系 1, 工学系 2) を撮影・編集した 48 分のサンプルビデオを用意した。これを観ながらノートエディタで講義ノートを作成する

研究ノート： 被験者それぞれの研究内容を論文化するための章立てをノートエディタを使って推敲する。時間制限は無し

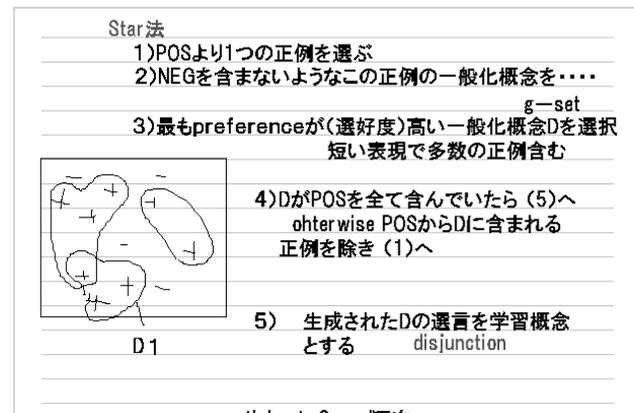


図10: 短時間使用実験で作成されたノート

図10は講義ノートのタスクにおいて、人工知能の学習アルゴリズムを図を使って説明している場面で書かれたノートの一例である。中央左の図はペンで、テキストはキーボードで書いている。

タスク終了後、ノートエディタを講義ノート(研究ノート)として使いたいか7段階評価で評価してもらった(1: 全く使いたくない.. 4: どちらでもない.. 7: 非常に使いたい)。全員の回答は図11の通り。

講義ノートとしてノートエディタを使いたい(5以上)と答えた被験者は6名、使いたくない(3以下)と答えたのは2名であった。回答の平均値は4.9、有意水準5%の片側 t 検定によりノートエディタは講義ノートとして有効という結果になった。

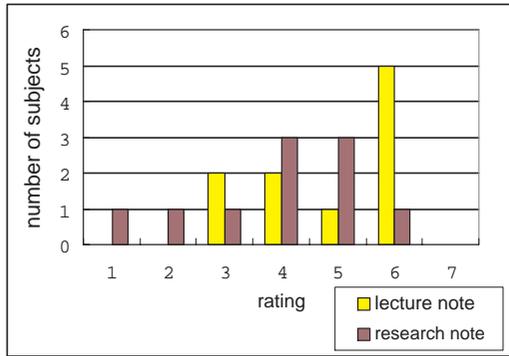


図 11: 短時間使用実験結果

肯定的な評価の理由にはオブジェクトの移動、消去、色の変更などの編集し易さとテキストや数式の読み易さを挙げた被験者が多かった。否定的な評価の主な理由は、図の描きにくさと、操作に気を取られてしまうことの2点であった。

研究ノートとしてノートエディタを使いたいと答えた被験者は4名、使いたくないと答えたのは3名であった。全く使いたくないと答えた被験者が1名含まれる。回答の平均値は3.9、ノートエディタは研究ノートとして有効とは言えないという結果になった。

他のアプリケーションと比較して使い慣れるのが簡単だと思うか(1:非常に難しい.. 4:普通.. 7:非常に簡単)という質問には10名中8名が簡単(5以上)、2名が普通(4)と答え、キーボードとペンを併用するノートエディタは他のアプリケーションと比較して使い慣れるのが難しいという印象は与えていないことが分かった。

3.2 長期使用実験

短時間使用実験の問題点は、実験時間が短いために被験者がノートエディタの利点や欠点を十分把握した上で評価しているとは考えにくいことである。そこで、ノートエディタを長期間継続的に使用してもらった実験を行った。

まず4名の学生(理系:A, B, 文系:C, D)に彼らが実際に受けている講義で3回ノートエディタを使って講義ノートを作成してもらい、継続してノートエディタを使用したいか希望調査を行った。A, B, Cはノートエディタが便利であるとして継続使用を希望した。DはノートPCを持ち運んで使用するほどのメリットを感じないためノートエディタを継続して使用したくないと答えた。A, B, CにはノートPCを貸し与えてその後も自由にノートエディタで講義ノートを作成してもらった。

図1, 図9, 図12, 図13はこの実験で作成されたノートの一部である。被験者A, B, Cのノートエディタ使用回数とアンケート結果を表2にまとめた。質問1, 2

の内容は以下の通り(7段階評価):

質問1: ノートエディタを使用して満足だったか

質問2: ノートエディタに使い慣れるのは大変だったか

被験者	回数	質問1	質問2
A	30	6: かなり満足	5: やや簡単
B	15	5: やや満足	6: かなり簡単
C	10	4: どちらでもない	2: かなり大変

表 2: 長期使用実験結果

被験者Aは30回の講義に出席してA4用紙60ページ相当の講義ノートを作成し、被験者Bは15回の講義に出席してA4用紙30ページ相当の講義ノートを作成した。共にノートエディタに肯定的な評価を与えている。評価の理由として、テキストと簡単な図形が素早く書けること、編集が容易なこと、ポップアップキーコマンド機能が便利であったことを挙げている。パターン描画機能は使われているが、パターン登録機能は使用されていない。ノート作成速度については、数式が連続する場面を除いて講義から遅れる場面は無かったとしている。評価の理由からも、テキストボックス自動生成機能及びポップアップキーコマンド機能がノート作成速度の向上に寄与していたことが分かる。

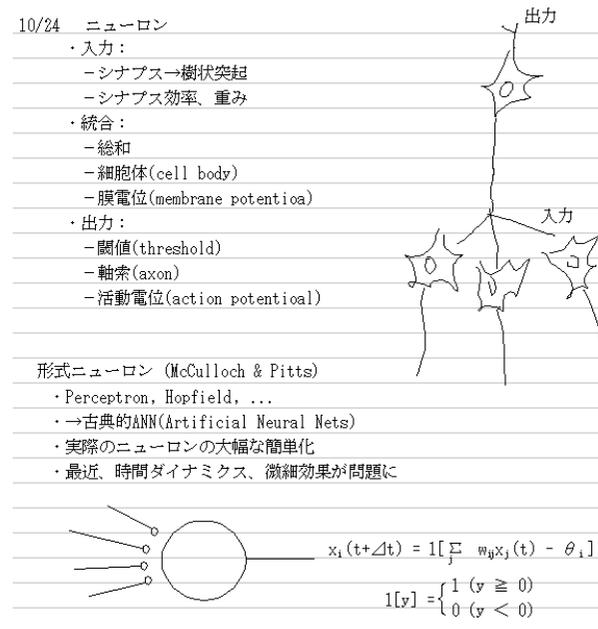


図 12: 長期使用実験で書かれたノート(1)

ノートエディタに不満だった点として被験者Aは別形式へのファイル変換ができないこと、被験者Bは自由曲線や直線などを組み合わせて書く図形が書きづかったことを挙げている。

図形の書きづらは短時間使用実験でも指摘されていた点である。これはペンの分解能などハードウェアの問題と、図形選択のアイコンが無かったなどソフトウェアの不備の両方が原因である。

被験者 C は 10 回の講義でノートエディタを使用した結果、肯定的ではない評価となった。使い慣れるのが難しくマニュアルを必要としたとコメントし、特に板書が少なく講師の話が中心の講義ではノート作成作業が遅れてしまうことがあったとしている。

ノートエディタがユーザに十分なパソコンスキルを求めることは想定していた。被験者 C, D は被験者 A, B と比較してパソコン使用経験が少なく、それが評価の違いに与えた影響は小さくなかったと考えられる。短時間使用実験の被験者は全員十分なパソコン使用経験を有していた。パソコン使用経験に関する条件付きで、長期使用実験でもノートエディタが講義ノートの作成に関して有効であったと言える。

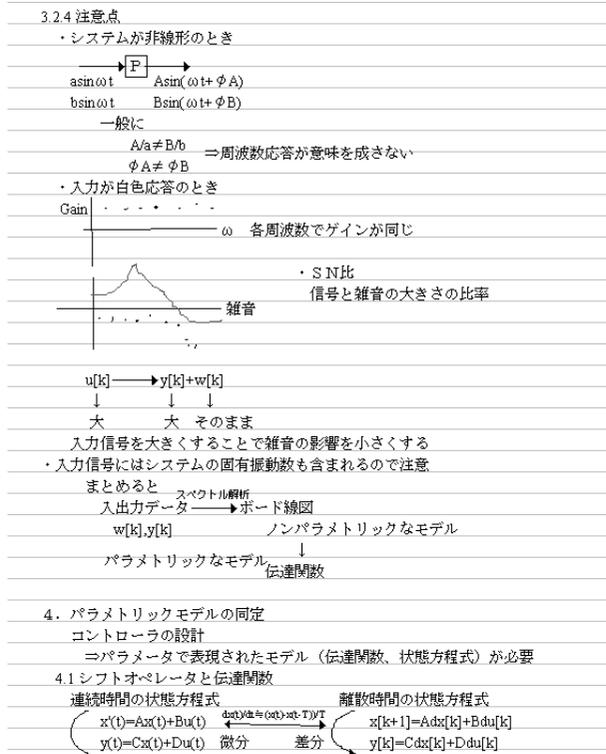


図 13: 長期使用実験で書かれたノート (2)

4 総括

我々はノートが未だ手書きで書かれているという問題に対して、ノートに文書作成速度と多様な図形の描画という要求があることに着目した。そして、それらの要求を解決するアプローチとしてキーボードとペンを併用するノートエディタを提案し、ノート作成速度を向上させ

るためのテキストボックス自動生成機能、ポップアップキーコマンド機能、パターン描画機能を実装した。ノートエディタの総合評価を行った短時間使用実験と長期使用実験では、パソコン使用経験が十分あるユーザという条件付きながら、開発したノートエディタが講義ノートの作成に関して有効であることが示された。

ノートを電子化して教官と学生で共有したり、あるいは音声情報や画像情報を取り入れるという研究はこれまで数多く行なわれてきた。しかしそこで扱われるノートは紙のノートに書かれている内容を書けるものではなく、システムが求めるノートと学生が普段書いているノートにギャップがあった。

本研究の成果は、キーボードとペンの併用によって紙のノートと同じようにノート PC でノートを書くことを可能にしたことである。これを言い換えれば、上述のようなこれまで存在していたギャップを埋める可能性を提示したものだと言える。今後はこれまで行なわれてきた研究成果への応用が求められる。

ダウンロード

ノートエディタのダウンロードはこちらから：
<http://www.sanpo.t.u-tokyo.ac.jp/~tatukawa/java.html>

参考文献

- [1] Peggy Van Meter, Linda Yokoi, Michael Pressley. College Students' Theory of Note-Taking Derived from Their Perceptions of Note-Taking. *Journal of Educational Psychology*. 1994. pp.323-338.
- [2] K. Truong, G. Abowd, J. Brotherton. Personalizing the Capture of Public Experiences. *Proceedings of UIST*. 1999. pp.121-130.
- [3] 駒方信子, 大平雅雄, 蔵川圭, 中小路久美代. リアルタイム講義における受講者の思考活動に着目した支援に関する研究. 情報処理学会研究報告 ヒューマンインタフェース 2001.7. 2001. pp.35-40.
- [4] 石田準, 坂東宏和, 加藤直樹, 中川正樹. 情報交換を可能とした電子黒板・ノートシステムの試作. 情報処理学会研究報告 コンピュータと教育 2001.12. 2001. pp.33-40.
- [5] James A Landay. Using Note-Taking Application for Student to Student Collaboration. *IEEE Frontiers in Education Conference*. 1999. 12c4-16.
- [6] J. Gwizdka, M. Fox, M. Chignell. Electronic Engineering Notebooks: A Study in Structuring Design Notes. *Proceedings of CHI'98, ACM Press*. 1998. pp.355-356.
- [7] Thomas Erickson. The Design and Long-term Use of a Personal Electronic Notebook: a reflective analysis. *Conference proceedings on Human factors in computing systems*. 1996. pp.11-18
- [8] Carol Vogt. Information Systems & Technology : Creating Equations With Microsoft Word. <http://ist.uwaterloo.ca/ec/equations/equation.html>