

ThumbSense: タッチパッド用対話技法の提案

ThumbSense: Automatic mode sensing for touchpad-based manipulations

暦本純一*

Summary. When you manipulate a touchpad, your hands should be away from the keyboard's home position. This feature hinders smooth switching between text input and pointer manipulation modes, and is considered one of the primary drawbacks of the touchpad. The proposed ThumbSense input technique aims to solve this problem by sensing users' finger presence on a touchpad. Keys on the keyboard, such as "F" or "J" keys, act both as normal keys as well as mouse buttons according to the perceived user's input status. This technique is implemented by utilizing the sensor feature of the touchpad, and is possible to apply most of currently available notebook computers without requiring new hardware.

1 はじめに

ノートパソコン用のポインティングデバイスとしては、タッチパッド(トラックパッド)とトラックポイントが主に用いられている。タッチパッドは、キーボードの下側に配置されたパッド上の指の動きを検出する。初心者でも機能が理解しやすい、機構部品がないので耐久性が高い、などの特徴をもち、広く利用されている。また、最近では認証用の手描きサインの入力装置としての利用など、新しい入力手段のためのデバイスとしての利用が期待されている。

一方、トラックポイントは、キーボード上に設置されたスティックを指で押すことでポインタを操作し、キーボードのホームポジションに手を置いたまま操作できる、という長所がある(トラックポイントの元々の開発意図は、携帯機器用の入力装置というよりも、キーボードとマウス間で手が往復する時間を短縮したい、ということだった)。タッチパッドではホームポジションに手を置いたまま操作することは困難で、無理に操作しようとしても、パッド上とボタン上に左右の親指を置く図1(左)のような不自然な指の配置や、あるいはキーボードから完全に手を離れた図1(右)のような指の配置を強いられる。このため、テキスト入力とマウス操作が頻繁に交代する場合には不向きであり、タッチタイピングに習熟したユーザはタッチパッドよりもトラックポイントを好むという傾向があった。

本研究では、キーボードのホームポジションに手を置いたままでもタッチパッドを操作することを可能にする入力技術、“ThumbSense”を提案する。提案技法はマウス(左右ボタン+ホイール)の上位互換になっており、マウス用の入力技法はそのまま利用可能である。タッチパッド操作の上位互換でもあるので、タッチパッド用に開発された対話技法(た

例えばエッジスクロールやホットスポットなどと併用することも可能である。提案する新技法によって、タッチパッドのトラックポイントに対する短所を克服でき、より使いやすい携帯型のポインティングデバイスが構成できる。



図1. タッチパッド操作時の指の位置：ホームポジションから手を離さないで操作できない

* ソニーコンピュータサイエンス研究所 インタラクションラボラトリー

2 ThumbSense

2.1 基本原理

本研究で提案する対話技法、ThumbSense の動作原理について説明する。タッチパッドは静電容量の変化を検出して指の位置を計測している。通常は、マウス互換の入力装置としてデバイスドライバが設計されているので、単に指をパッドに置いただけでは入力イベントが発生しない。しかし、静電容量センサーとしてはパッド上に指が置かれているかどうかを検知することができる。つまり、一種のタッチセンサーとみなすことができる。

本研究の入力技術では、この特性を利用して、ユーザがテキスト入力をしようとしているのか、ポインティング（タッチパッド）操作をしようとしているのかを、パッド上の指の存在検知で判定する。指がパッドに置かれている間は（ユーザがテキスト入力をしないという想定で）、キーボードのモードが自動的に切り替わり、マウスボタンとして機能するようになる。たとえば、指がパッドに触れている間は、**[F]** や **[J]** キーがマウスの左ボタンに、**[D]** や **[K]** キーがマウスの右ボタンとして機能するようになる。この機構によって、図2のように、手をホームポジションに置いたままでパッド操作が可能になる。

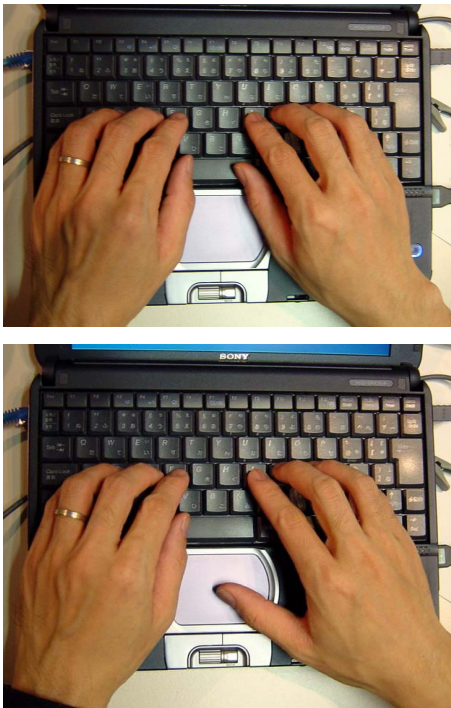


図 2. ThumbSense 利用時の指の位置（左：キーボード入力、右：パッド操作）

マウスボタンをキーボードで代用するためには、左右キーに対応して 2 個のボタン、中ボタン（あるいはホイール）を代用するためにさらに 1 個のボタ

ンが必要である。本提案では、ホームポジションに指をおいたままで操作を可能にするため、**[F]** および **[J]** キーをマウスの左ボタンに、**[D]** および **[K]** キーを右ボタンに割り当てている。つまり左右対称に機能を割り付けているので、右手・左手でも同じ感覚で操作することができる。ホイールキーとしては **[S]** および **[L]** キーを割り付けている（この場合、画面上のマウスは移動せず、スクロール処理などが起きる）。これらの割り当ては利用者の好みによって再定義可能である。

テキスト入力時には、手形状が図 2 左のようになり、パッドから指が離れる。システムはこれを検知して、キーボードを通常モードに戻す。マウスボタン兼用のキー群も普段のとおりに利用可能になる。英語入力でも日本語入力でも、テキスト入力時には頻繁にスペースキーを打鍵するので、（右手の）親指は自然にスペースキー上に置かれることになり、利用者が特に意識しなくてもタッチパッドモードとテキスト入力モードを切り替えることができる。

本技法では、タッチパッド用のボタン（通常パッドの下側に配置されている）の機能を変更しているわけではないので、従来通りのパッド操作ももちろん可能である。したがって従来からの上位互換性を満たすことができる。逆に、タッチパッド用ボタンに、パッドに指が触れていない間だけ別の機能を割り当てることも可能意である。頻繁に利用するアプリケーションを起動するためのショートカットボタンや、インターネット電話やボイスチャットの応答ボタンなどに割り当てることができる。さらに、本方式に特化したキーボードを設計した場合は、パッド下側のボタンが不要になるので、ノート PC のサイズを小型化することができる。

2.2 状態遷移

図 3 に、ThumbSense の状態遷移を示す。基本的にはパッド上の指の有無でモードを識別しているが、ドラッグ中に指が離れることを考慮して、マウスボタン（実際にはキーボードキー）が押されると、それが解除されるまではマウスモードに固定している。図では、状態 S0 から出発して、パッドに指が触れていない場合は S0 → S5 → S0 という経路で通常のキーボード打鍵を処理する。パッドに指が触れている場合は、マウスに相当するキー（たとえば **[F]**、**[J]** キー）の押下時には S1 → S2 と遷移する。キーを押した後はパッドから指がはなれてもドラッグモードのままである（S2, S3）。これらの状態ではシステムが発生するオートリピートイベントも無視される。一方、パッドに指が触れている場合でも、マウスボタンに割り当てられていないキーは通常通り処理する（S1 → S4）。また、状態 S3 と状態 S5 はともにキーが押され、パッドから指が離れているという状態だが、初期状態からそこに至る経路が異なるので異な

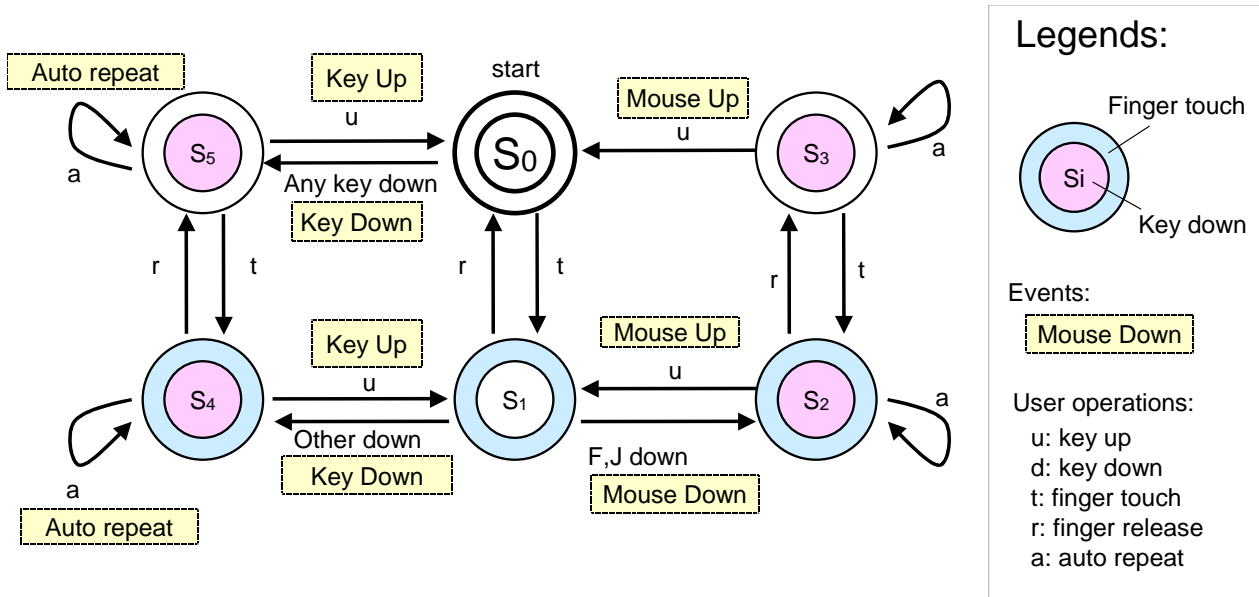


図 3. 状態遷移図

る状態として扱われる。即ち、前者はドラッグ中にパッドから指が離れている状態、後者はキー入力時の状態として識別される。このような状態遷移を用いることで、利用者にとってはモードを意識することなくマウス、キーボードの操作が可能になる。

2.3 超多鍵マウスとしての操作

以上が ThumbSense の基本機能だが、単に現状のマウス互換の操作を提供するだけでなく、新しい対話技法を構築することが可能である。とくに、キーボード上の多くのキーをマウスボタンとみなすことができるので、多くの操作をワンタッチで呼び出すことができる。

操作手段には「パッドに指を置いた状態で特定のキーを押す」、つまりパッドをシフトキーやコントロールキーなどと同様の新たな修飾キーとして用いる場合がまず考えられる。通常の修飾キーはキーボードの左右に配置されており、ホームポジションから操作する場合に指が無理な動きをすることが多いが、タッチパッドを利用した方法では「親指シフト」と同様、ホームポジションの手の形を崩さずに操作できる。従来の修飾キーは、既にアプリケーションや OS によって多くの機能が割り付けられているので、個人用のマクロキーの割り当てが必ずしも自由ではないが、ThumbSense によってあらたな修飾キーの空間が提供されることになる。

さらに、パッド上での指の動きを組み合わせると豊富な対話操作を実現することが可能である。たとえば図形エディタなどでは、図形作成や移動などを特定のキーに割り当てることができる。Emacs などのテキストエディタでは、多くの機能をメニュー選

択ではなく修飾キーとキー入力の組み合わせによって指示するが、ThumbSense による入力操作はそれのポインティング・デバイス版であると言える。

また、両者を組み合わせると、キー入力のみでは単純な機能、ポインタの動きを組み合わせると操作の調整を指示することができる(図 4)。たとえば「アプリケーション切り替え」ボタンをキーに割り当てるとする。単にこのキーを打鍵したときは直前のアプリケーションに画面が切り替わるが(図 4 左)パッド上で指を移動させるとメニューがポップアップし、ポインタ操作によって任意のアプリケーションを選択することができる(図 4 右)。このように基本機能をワンタッチで呼び出すことで操作の迅速性を確保し、拡張機能にポインタ操作を用いた柔軟な対話技法を組み合わせることができる。同様な技法は、Web でのナビゲーション(何段階前へ戻るかも指定できる「戻るボタン」)などにも適用可能である。

2.4 実現と評価

ThumbSense の機能を実現するプログラムを、Windows XP 上に実現した。タスクバーアプリケーションとして動作する。Windows のイベントフック機能によって本来のマウス入力とキーボード入力をトラップし、タッチパッドのデバイスドライバの機能を用いて指の近接を認識し、前節で説明した状態遷移に従って適切なマウス・キーボードイベントを生成している。したがって、アプリケーション側には特に変更を加える必要がなく、Windows 上のすべてのソフトウェアで ThumbSense の機能を利用することができる。

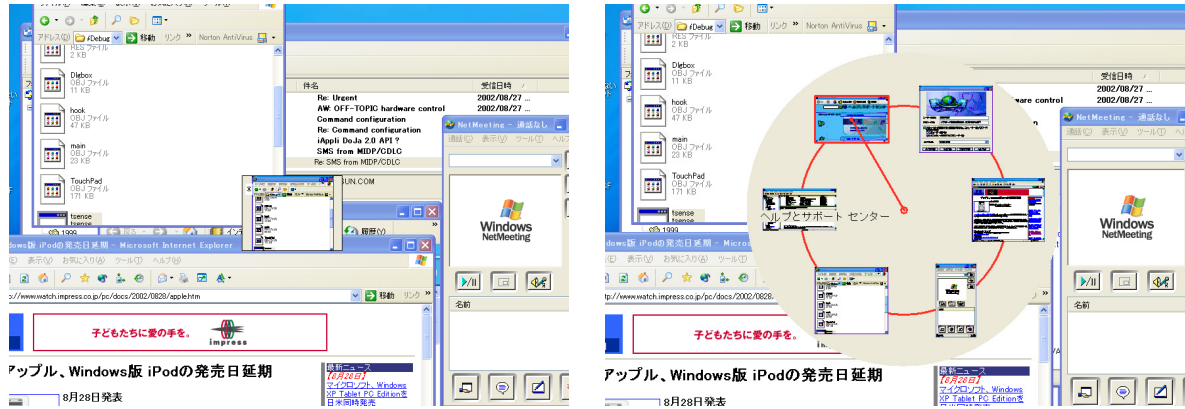


図 4. キーコマンドとポインティングの組み合わせ操作の例 (左:パッドに指を置き、キーを押した直後。右:キーを押したままパッド上で指を動かした場合。)

筆者の日常業務で約一ヶ月間継続的に使用し、非公式ではあるが良好な結果を得ている。試用の結果、タッチパッドの下側のボタンを使う機会が激減した。左手でマウスボタン(キーボード上)、右手親指でタッチパッドという組み合わせが基本だが、片手による操作(右手親指がパッド、人差し指・中指で[J], [K]キーを操作)も可能であった。また、細かな作業の場合、右手人差し指でパッドを操作し、マウスボタンは左手([F], [J]キー)というスタイルをとることができる。この場合、右手はホームポジションからはずれるが、左手がホームポジションに置かれているので、従来の操作スタイルと ThumbSense の操作スタイルの中間的な使い方といえる。この場合、片方の手がホームポジションに残っているため、両手がホームポジションからはずれている場合と比較して、体感的には手の復帰がより容易であるように感じられた。

多ボタンマウスとしては、「アプリケーション切り替え」、「Webの前後移動」、「最大・最小化」、「アプリケーション終了」、「ブラウザ起動」、「メール起動」などの機能をキーに割り当てて試用を行なったが、(Emacs などと同様に)キーアサインの記憶・習熟を前提とはするものの、非常に快適に操作することができた。

3 関連研究

Hinkely らは、マウスにタッチセンサーを装着したタッチセンシティブマウス [3] を提案している。ユーザがマウスに触れているときだけツールパネルが画面に表示されるなどの利用方法が示されている。Embodied UI [1] や Hinkley らの PDA [2] では、PDA 筐体に圧力センサーや近接センサーを配し、利用者が機器を手にとっているかどうか、などの操作状態を機器への入力としている。このように、比較的簡単なセンシング技術によって利用者の操作意図を認

識することで、余分なモード遷移コマンドが不要になり、ユーザインタフェースが単純化される。

本研究もタッチパッドのセンシング機能を応用して、利用者の操作状況を認識している。テキスト入力時とパッド操作時で指の位置が自然に変化することを利用しているため、ユーザに特別な「モード切り替えコマンド」の必要を感じさせない。また、キーボードのキーを動的にマウスボタンに変化させるなどの新しい技法を提案している。従来の研究では新規にセンサーハードウェアを機器に追加する必要があったが、本インタフェースは、タッチパッドの特性を利用しているため、現在普及している大半のノート PC で即座に利用可能である。MacKenzy らは、タッチパッドの圧力検知能力を利用してマウスプレスを表現する方法を提案している [4]。パッドの下に組み込まれたアクチュエータで触感フィードバックを生成し、ボタンのクリック感を提供している。この方法では単一ボタンのマウスは表現できるが、2 ボタン以上やホイールなどの機能は実現できない。またタッチパッドの圧力検知は指の触れ方や個人差によって変化するので調整が必要である。

タッチパッドのドライバでは「タップ&ドラッグ」のような特定のジェスチャーによってドラッグなどの操作を代用する機能が提供されているが、誤操作の問題を完全に回避しているとは言いがたい。また、マウスの左右ボタンのちがいを表現できないので、ThumbSense と異なり、マウス機能を完全に代替しているわけではない。

タッチパッドのその他の特徴的な機能として、パッドの特定の場所をタップすると設定した機能が立ち上がる「ホットスポット」機能や、パッドの端を指でなぞってスクロールする「エッジスクロール」機能などがある。これらの機能は ThumbSense と共存可能である。

4 まとめ

本論文では、タッチパッド用の新しい入力技法 ThumbSense を提案した。本研究によって、従来のタッチパッドの操作上の大きな問題点である、「ホームポジションから手を離さないと操作できない」という課題が解決できた。また、特別なハードウェアを必要としないので、現状のノートパソコン上で直ちに利用可能である。

尚、本システムの評価版は

<http://www.csl.sony.co.jp/person/rekimoto/tsense/>
よりダウンロード可能である。

謝辞

タッチパッドのデバイスドライバに関する情報を提供して頂いたアルプス株式会社、ソニー株式会社モバイルネットワークカンパニーに感謝する。

参考文献

- [1] Kenneth P. Fishkin, Thomas P. Moran, and Beverly L. Harrison. Embodied User Interfaces: Towards invisible user interfaces. In *Engineering for Human-Computer Interaction: Seventh Working Conference on Engineering for Human-Computer Interaction*, pages 1–18, 1998.
- [2] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. Sensing techniques for mobile interaction. In *UIST 2000*, pages 91–100. ACM, 2000.
- [3] Ken Hinckley and Mike Sinclair. Touch-sensing input devices. In *CHI'99 Proceedings*, pages 223–230, 1999.
- [4] I. Scott MacKenzie and Aleks Oniszczak. A comparison of three selection techniques for touchpads. In *CHI'98 Proceedings*, pages 336–343, 1998.