

# Bubble Clusters: アイコンの空間的なまとまりを利用した情報管理機構

## Bubble Clusters: Information Management Mechanism Using Spatial Clusters of Icons

渡辺 奈夕子 五十嵐 健夫\*

**Summary.** We propose “Bubble Clusters” as a mechanism for representing lighter group structures than folders by exploiting the use of spatial aggregation of icons. In actual workspaces, people “pile” related documents together to organize them. Bubble Clusters allows the users to manage files on their PCs by piling up them and access the files in piles easily. In particular, it provides the interface which has three features: automatic grouping of aggregated icons, automatic spreading of overlapping icons, and coexistence of operations of icons and groups. Our goal is not to substitute existing folders but complement them.

We implemented a prototype system, and show this mechanism is effective to the management of files when group structure is ambiguous and transient.

### 1 はじめに

グラフィカルユーザインタフェース (GUI) におけるファイルの管理は、通常、階層的なフォルダ構造とフォルダ内に並べられたアイコンによって行われる。フォルダ構造は、確定的・永続的なグループ構造を表現するには適しているが、明示的なフォルダの作成と各ファイルのフォルダ内への明示的な移動が必要であり、分類が曖昧なファイルを扱う場合や頻繁に再分類を行う場合には効果的でない。一方、関連したアイコンをまとめて配置することで、空間的にフォルダのグループ構造を表現する方法も多くのユーザによって利用されている。この方法は、明示的なグループ手続きが必要でないため非常に柔軟に扱うことができるが、計算の内部ではグループとして認識されていないため、グループとしてまとめて作業する場合には不便が生じる。また、数多くのアイコンが重なり合うと個々のアイコンへのアクセスが困難になるという問題もある。

このような問題を解決するため、本研究では、アイコンの空間的なまとまりを自動的にグループとして扱うことで、フォルダ構造よりも軽いグループ構造を実現する手法を提案する。本システムでは、空間的に近くにまとめられたアイコンは自動的にグループ(クラスター)として認識され、その結果がアイコンを取り囲む連結領域として提示される。ユーザはその連結領域上でマウス操作を行うことで、グループ全体に対して処理を行うことができる。また、重なり合っただけのアイコンのそれぞれに対する

アクセスを容易にするため、必要に応じてグループ内のアイコンを自動的に重なりなく広げて表示するという機能も提供している。

以降では関連研究に触れた上で、この手法の詳細およびこの手法を実装したプロトタイプシステム(図1)の詳細について述べる。また何人かのユーザに実際にタスクを与え、その結果得られた意見について議論する。

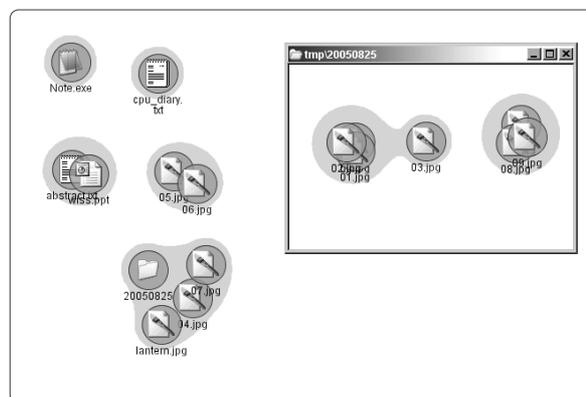


図 1. Bubble Clusters

### 2 関連研究

デスクトップ上でのアイコンの操作に関しては多くの研究がなされてきた。

Manderらはファイルを積み重ねてフォルダの代わりに pile(書類の山)を作成する ‘Pile’ Metaphorを提案した [11]。しかし ‘Pile’ Metaphor では結局 pile の中身进行操作するために特定の動作をせねばならず、また完全に重なってしまうために分類作業がしにくい。つまりこの ‘pile’ は、独自のフォルダの

ようになってしまっている。

また、ドラッグアンドドロップを支援するには drag-and-pop [1] や Drag-and-Guess [15] などの手法がある。Drag-and-Guess は分類における予測結果を利用した支援であるが、ドラッグアンドドロップ支援の多くがドラッグ方向などを考慮することで、ドロップ先へ簡単にドロップするためのものである。drag-and-pop は、アイコンがカーソルの方へ近づいてくるといったものであり、アイコンがカーソルの動きに応じて移動してもユーザに特に問題を与えていないことが示されている。

一方、分類作業に関する研究も多くなされている。分類するためのフィールドとして空間的配置を利用したのが Robertson らの Data Mountain である [13]。Data Mountain では Web のブックマークの管理をするのに、3次元フィールドに自由にページのサムネイルを置いておく。従来のツリーリストでの管理に比べ、ユーザが内容と保存位置の関係を覚えやすく、一度分類したものに関して再び目的のページを探し出すのが容易であると示された。また空間的配置はツリー構造よりも、曖昧な構造や不完全な構造を作成できるという点においてユーザの意図を反映しやすい。それを利用した研究では、Marshall らの VIKI がある [12]。

Grossman らの Bubble Cursor は、エリアカーソルの応用である [5]。アイコンにヒットしているかどうかを示すために、カーソルがアイコンにヒットするとアイコンの一部がカーソルを包み込むような bubble エフェクトが表示される。この複数のオブジェクトを包み込むエフェクトは、MacIntyre らの Kimura [10] でも見られる。Kimura では関連するプロセスのスクリーンショットをこのエフェクトでまとめて表示している。本研究ではこのようなエフェクトをアイコンがグループ化されていることを示すために利用する。

Bubble Clusters は、アイコンの空間的配置によるファイルの効果的な管理をより行いやすく、更にフォルダよりも柔らかいグルーピングによってファイルをより整理しやすくすることが目的である。

### 3 ユーザインタフェース

#### 問題と概要

以下では Bubble Clusters の詳しいユーザインタフェースデザインについて述べる。Bubble Clusters の大きな特徴は

- 自動グループ化
- 重なりを解消したアイコンの表示
- グループ操作とアイコン操作の共存

である。

#### 3.0.1 自動グループ化

従来の GUI においてフォルダを使用する場合、フォルダの作成には右クリックをしてメニューから作成を選ばなければならない。少数のファイルをまとめたいだけのときにも明示的にフォルダの作成とフォルダへの名前の付加が必要なため、非常に煩わしい。

そこで Bubble Clusters ではグループ化を自動で行うようになってきている。それぞれのアイコンは自らの連結領域 (バブル) を持っており、図 2 のようにアイコンの周りに薄い色で表示される。この領域はアイコンがグループ化されるかどうかを決める領域であり、またユーザに作成されたグループの範囲を明示するためのものである。新しいグループの作成及びアイコンのグループへの追加は、この領域を通して自動的に行われる。ユーザがアイコン同士を近づけることでこの連結領域が接すると、その領域に含まれるアイコン群は 1 つのグループとして認識される。アイコンが離れると、連結領域はそれぞれのアイコンに分かれてグループは自動的に解消される。またこの領域は、グループの操作においてはハンドルとしての役割も果たす。グループ操作については後述する。



図 2. アイコンの連結領域

図 3 では、アイコンが自動でグループ化される過程 (a→b→c→d)、また逆にグループが解除される過程 (d→c→b→a) を示されている。連結領域は、アイコンを核とした泡のような挙動を示す。(a) の上段ではアイコンは個別のアイコンであり、グループ化されていない。また下段では右上のアイコンは左下のグループに属していない。連結領域は離れて表示されている。ユーザが右上のアイコンを左下のアイコン (グループ) に近づけていくと、(b) のようにお互いの連結領域が延びていく。(c) で領域が 1 つになり、2 つのアイコン (アイコンとグループ) は 1 つのグループとして認識される。更に近づけると (d) のように重ねることができる。グループを移動する場合にも、領域が他のグループと接すれば 1 つ

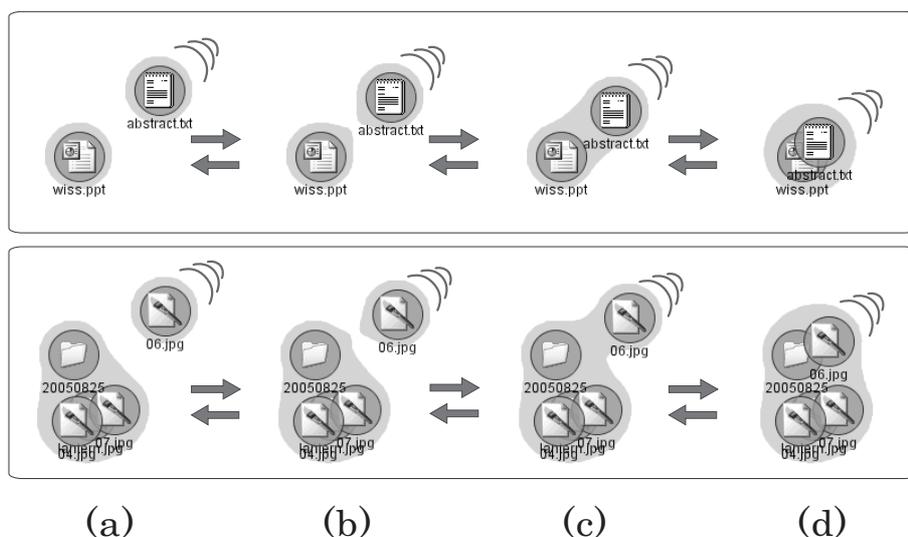


図 3. アイコンのグループ化

のグループに統合される。

### 3.0.2 重なりを解消したアイコンの表示

フォルダでは中のファイルを見たり操作したりするためには、フォルダのアイコンをダブルクリックしてファイルを立ち上げなければならない。また現在の GUI では、図 4 の (a) のようにアイコンが重なっている場合、下にあるアイコンを見ることや操作することができない。操作するためには、上に乗っているアイコンをマウスでドラッグして退けなければならない。

Bubble Clusters ではそのような問題を解消するために、ユーザが操作したい場所でアイコンが重なっていた場合にはその集合を互いに重ならないように広げて表示する。またアイコンが広がった状態でも、ユーザは個々のアイコンに対しての操作を行うことが出来る。まず仮定として、連結領域上にカーソルがしばらく留まっているということは、ユーザはそこに注目しているということを示しているとする。あるグループの連結領域上にカーソルが留まっていると、図 4 の (b) のようにそのグループの中の重なっているアイコンが広がって下のアイコンが見えるようになる。広がる時はアニメーションし、広がる前と後のアイコンの対応がとれるようになっている。また広がっている状態では連結領域の色が変わり、他のグループや他のアイコンの連結領域とくっつくことはない。グループからカーソルが出ると、広がっていたグループは元に戻る。

### 3.0.3 グループ操作とアイコン操作

グループに属していても属していなくても、それぞれのアイコンは普通のアイコンとして操作できる。

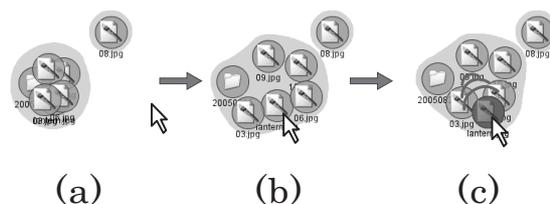


図 4. 重なりを解消

アイコンのドラッグをした際にはドロップした位置によって操作の意味が変わる。ドロップ時にカーソルが他のアイコン上に乗っている場合にはドラッグアンドドロップ、アイコン上に乗っていない場合は移動（場合に応じて自動グループ化）として認識される。このときユーザに対して現在の操作がドラッグアンドドロップと認識されるか移動と認識されるかを明確に示すために、ドラッグアンドドロップ時にはアイコンの連結領域の表示を一時枠表示にするというフィードバックを用意している。図 5 の (a) がドラッグアンドドロップ、(b) が移動となる。(a) ではカーソル位置が左下側のアイコンの上に乗っているが、(b) では乗っていないことがわかる。

これまでの GUI ではたとえアイコンが重なっていても、アイコン 1 つずつのみでしか操作ができなかった。範囲選択をすることでアイコンを複数選択すればアイコン群全体の移動やドラッグアンドドロップもできるが、アイコンの並びが複雑になると矩形選択の範囲選択では選択が困難になってくる。Bubble Clusters のグループの操作では、図 6 の (b) のように操作したいグループの連結領域を掴むと、そのグループ全体を操作することができる。またグ

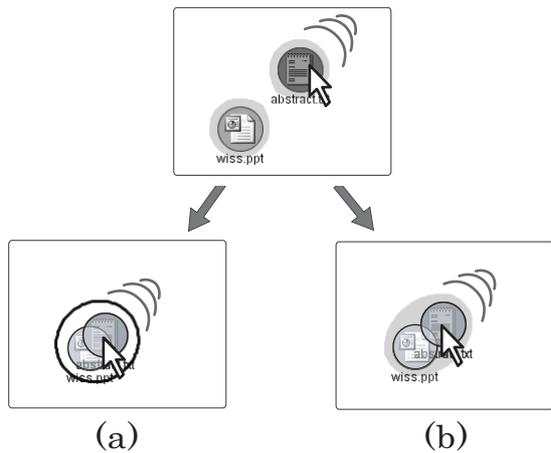


図 5. ドラッグアンドドロップと移動

グループの中でも (a) のように連結領域でなくアイコン内を掴むと、そのアイコン 1 つに対する操作と認識される。アイコンが広がって表示されている状態でも同じようにアイコン、グループ全体双方に対する操作が可能である。

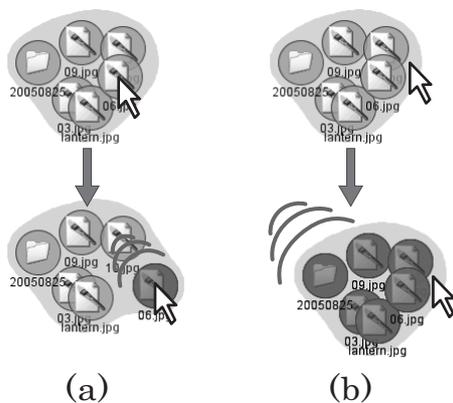


図 6. アイコンの操作とグループの操作

## 4 プロトタイプシステム

プロトタイプシステムは Java™1.4 で作成した。図 1 にプロトタイプシステムの画面を示す。以下では、自動グループ化とアイコンの重なり解消のアルゴリズムについて述べる。

### 4.1 自動グループ化

アイコンの周りの連結領域は、アイコンをメタボール (濃度球) のように中心からの距離によって濃度に変化する円とみなし、その等値面の内側の領域として求められる [3]。すなわちそれぞれのアイコ

ンの中心からの距離を変数とする濃度関数をアイコンの存在する場 (デスクトップやファイラ) ごとに保持し、そのアイコンごとの和があるスカラー値  $t$  となる境界線を作り、その値が大きくなる側をクラスタ範囲とする。そしてそのクラスタの中にあるアイコン群を 1 つのグループとして扱う。

濃度関数からの等値面の算出を行うために、マーチングキューブ [8] の手法を 2 次元に変えて利用している。画面に対してグリッドを敷き、そのグリッドの頂点のスカラー値と  $t$  の大小の比較を行う。ある格子点と隣接する格子点の間で比較結果が入れ替わったら、その間には境界線があることになる。この比較結果の入れ替わりと境界線の形をパターン化してルックアップテーブルにすることで (図 7)、高速に全体の境界線を算出することができる。

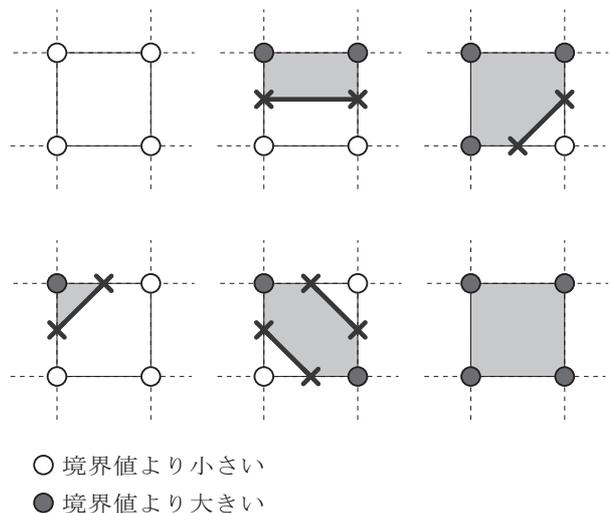


図 7. 等値面境界のパターン

### 4.2 重なりを解消したアイコンの表示

ここでは、互いに重なり合うようにして配置されたアイコンの集合を、互いに重ならないように広げるためのアルゴリズムについて説明する。より具体的には、重なりのある配置を入力として受け取って、なるべく元の相対的な位置関係を維持しつつ、重なりが無くなるような配置を出力として返す処理を行う。もっとも簡単な実装としては、近くのアイコン同士の間で反発力を発生させてアイコンを少しずつ動かしていくばねモデルによるものが考えられる [14] が、この方法では安定した結果を得るのに時間がかかる、また正しく動作させるためのパラメータ調整が困難である、といった問題がある。そこで本システムでは、理想的な位置関係を表すエネルギー関数を設計し、それを最小二乗法によって最小化することで、パラメータ調整や繰り返し計算を行うことなく適切な結果を計算する手法を採用している [7]。

まず、アイコンの近接関係を抽出するために、ア

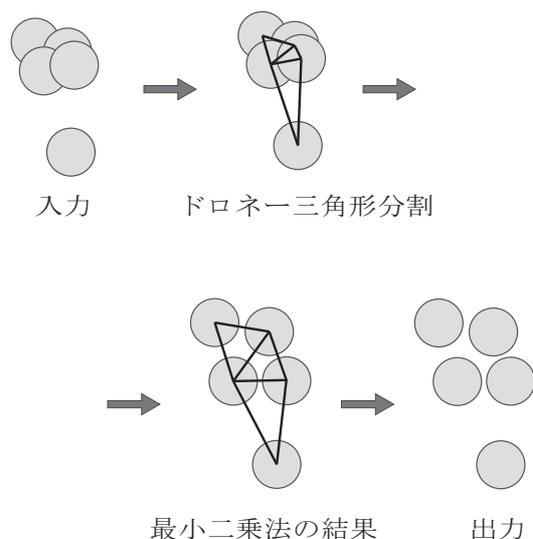


図 8. 重なり解消の過程

アイコンの中心を入力点群としたドロネー三角形分割を行う。次にドロネー辺で結ばれたアイコン同士の間で、「重なりのない理想的な位置関係」を表す相対位置を求める。具体的には、各アイコンの対について、重なりがない場合にはそのままの相対位置を、重なりがある場合には重ならなくなるように両者を結ぶ直線に平行にずらしたものを理想的な相対位置として設定する。最後に、計算の結果得られるアイコンの座標を自由変数として、それらの求めるアイコンの間の相対位置と理想的な相対位置との差の2乗の総和を表すエネルギー関数を作る。このエネルギー関数を最小2乗法で最小化する（自由変数でエネルギー関数を偏微分し、得られる連立1次方程式を解く）ことで、アイコン間の相対位置ができるかぎり理想的な相対位置に近くなるようなアイコンの配置を得る。なお、相対位置だけでは絶対的な位置が一意に定まらないので、アイコン全体の重心は動かないという制約を加えている。また、一度の計算では重なりを完全に回避できない場合もあるので、現在の実装ではドロネー三角形分割を含む上記の一連の処理を10回繰り返している。

## 5 まとめと今後の展望

本研究ではフォルダのよりも柔軟なグループ構造でファイルの管理をより効率的に行うために、アイコンの空間的な位置を利用して自動グルーピングを行うファイル管理機構、Bubble Clustersを提案した。またこの機構の詳細、実装におけるアルゴリズムを述べ、簡単なユーザテストを行った。今後このBubble Clustersの導入がフォルダのみに比べて有効かどうか、詳細な評価実験を行う予定である。

このBubble Clustersの機構は他の応用も考えられる。例えば2次元上に発言していくコミュニケー

ションウェア [4] などでは古い発言を重ねていって管理したい場合や、議論する上で意見ごとに発言をまとめたい場合がある。まとめたい発言を一度に動かしたいとき、現状では1つの発言ずつ動かさなければならない。また、1度重ねてしまった発言をもう一度見るのは手間がかかる。Bubble Clustersの機構を使えば、意見ごとに近くにまとめたい発言は自動グループ化され、その発言グループは一度に操作することができる。カーソルを置くことで自動で重なりを解消することで、グループの中の過去の発言も簡単に見ることができる。

さらにGUIのアイコンだけでなく写真やメモなどの分類作業にも応用できると思われる。写真の分類においては、フォルダを新たに作成しグループ化し直すという作業が頻繁に起こり、また2つのグループ両方に分類可能なことも多い。既存の手法では、そのような頻繁な再グループ化という面倒な作業が結局解消されず、また分類に困るという点も解決されていない [2][9]。Bubble Clustersのインタフェースを使うことで、再グループ化が簡単に行え、また空間的配置を使って曖昧な分類を行えるようにすることができる。そのように空間的配置をうまく使って分類を楽にするためには、現在はアイコンの選択とクラスタの選択の2段階のみであるところを、クリック回数などで選択範囲を段階的に変えるという工夫が考えられる [6]。空間的な配置を利用する場合、段階的な範囲の変更は有効と思われる。例えば1クリックで1アイコン、2クリックでクラスタ内の重なっているアイコン群、3クリックで1クラスタ...を選択するといった応用が考えられる。そうすることでグループに入っているか入っていないかという2値に囚われることなく曖昧な分類が更にしやすくなる。

フォルダをフォルダとして扱う以外に、フォルダも同じように中身を広げて表示するという考えられる。もしくは、クラスタでは無く階層をサポートするという方向性もある。しかし本研究では階層に収まりきれないグループ構造をサポートすることが目的であったため、クラスタという形を取った。

グループを自動作成するため、フォルダのように名前を付けることが出来ない。グループ内のファイルの名前、ファイルの種類などから自動でグループの属性を付加するという拡張が可能である。グループの名前を自動作成して表示したり、連結領域の色などでアンビエントに表示したりすることで、ユーザに個々のファイルに注目しなくても内容がわかるようにできる。

また課題として、ドラッグアンドドロップをするときと近づけるときの区別を明確にする必要がある。現在はドラッグアンドドロップとして認識されるときには連結領域を枠表示にするフィードバックだが、更に明確なフィードバックを返すことを考慮

しなければならない。

## 参考文献

- [1] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Proceedings of Interact 2003 Conference*, pp. 57–64, 2003.
- [2] B. Bederson. PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubbles. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST2001)*, pp. 71–80, 2001.
- [3] J. F. Blinn. A generalization of algebraic surface drawing. In *ACM Transactions on Graphics, Volume 1, Issue 3 (July 1982)*, pp. 235–256, 1982.
- [4] A. M. Fass, J. Forlizzi, and R. Pausch. MessyDesk and MessyBoard: Two Designs Inspired By the Goal of Improving Human Memory. In *DIS 2002 Designing Interactive Systems*, pp. 303–311, 2002.
- [5] T. Grossman and R. Balakrishnan. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–290, 2005.
- [6] T. Igarashi, S. Matsuoka, and T. Masui. Adaptive Recognition of Implicit Structures in Human Organized Layouts. In *Proceedings of Visual Languages '95*, pp. 258–266, 1995.
- [7] T. Igarashi, T. Moscovich, and J. F. Hughes. As-Rigid-As-Possible Shape Manipulation. In *ACM Transactions on Computer Graphics, Volume 24, Issue 3, ACM SIGGRAPH 2005*, pp. 1134 – 1141, 2005.
- [8] A. Kaufman. *Volume Visualization*. IEEE Computer Society Press, 1991.
- [9] J. Kim, S. M. Seitz, and Maneesh Agrawala. Video-Based Document Tracking: Unifying Your Physical and Electronic Desktops. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST2004)*, pp. 99–107, 2004.
- [10] B. MacIntyre, E. D. Mynatt, S. Vaida, K. M. Hansen, J. Tullio, and G. M. Corso. Support For Multitasking and Background Awareness Using Interactive Peripheral Displays. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2001)*, pp. 41–50, 2001.
- [11] R. Mander, G. Salomon, and Y. Y. Wong. A 'Pile' Metaphor for Supporting Casual Organization of Information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 627–634, 1992.
- [12] C. C. Marshall and F. M. Shipman. VIKI: Spatial Hypertext Supporting Emergent Structure. In *Proceedings of 1994 European Conference on Hypermedia Technology (ECHT '94)*, pp. 13–23, 1994.
- [13] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data Mountain: Using Spatial Memory for Document Management. In *Proceedings of UIST '98, 11th Annual Symposium on User Interface Software and Technology*, pp. 153–162, 1998.
- [14] 伊藤 貴之, 井上 恵介, 土井 淳, 梶永 泰正, 池端 裕子. 力学モデルを用いたグラフデータの画面配置手法の改良. 情報処理学会グラフィクス& CAD研究会, 2001-CG-103, 2001.
- [15] 西田健志, 五十嵐健夫. Drag-and-Guess: 予測付きドラッグアンドドロップ. WISS 第12回インタラクティブシステムとソフトウェアに関するワークショップ, pp. 31–34, 2004.