

VelvetPath : スケッチとペイント操作による レイアウトデザインシステム

飛田 博章[†] 磯 大輔^{††} 暦 本 純 一[†]

VelvetPath は、スケッチとペイントの手法を用いたレイアウトデザインシステムであり、ユーザの描画したストロークやペイント領域に対して、情報が配置される点に特徴がある。操作はストロークの描画により行われるが、ペン属性の組合せにより、3次元レイアウトや階層的なレイアウトなど、様々なレイアウトをデザインする機能を提供している。また、ペンの太さが情報の大きさや情報間の関連に対応していることに加え、ペイント領域も情報の大きさに対応しているので、情報表示の変数を直接的に操作することができる。こうしたペン属性と太さの調整により、ユーザの注目や用途に柔軟に対応できるレイアウト手法であると考えられる。また、本手法は、情報の閲覧や検索などの用途に加え、プレゼンテーション、コミュニケーションや3次元CGのコンテンツ生成支援など様々な分野での応用が可能である。本論文では、情報空間に自由にレイアウトをデザインできるシステム VelvetPath と、その応用について述べる。

VelvetPath: Layout Design System with Sketch and Paint Manipulations

HIROAKI TOBITA,[†] DAISUKE ISO^{††} and JUN REKIMOTO[†]

We describe the VelvetPath system, a system that allows users to design layouts for visualized information by using sketch and paint manipulations. Many systems enable users to visually search and browse through information by treating the data as visualized nodes. While these layouts are pre-defined and useful for the applications considered by the designers, users cannot freely change or redesign the layouts. In contrast, users of the VelvetPath system can freely design and change layouts by simply drawing strokes. Since the information layouts are defined automatically according to the user's strokes, the user can create layouts based on his/her own needs and retrieve information using these layouts. Moreover, because all the manipulations in the system are based on simple sketching and painting interactions, the system is easy to use. These simple manipulations are useful not only for information retrieval, but also for a wide variety of applications such as creation, presentation, and communication.

1. はじめに

コンピュータのソフトとハードの両者の発達にとともに、様々な形式の情報（画像、動画やドキュメントなど）を高速に表示することが可能になった。加えて、情報空間と自然に対話するためのコンピュータ（実世界指向インタフェースや携帯型コンピュータなど）が次々に登場している。今後、こうした情報やコンピュータの多様化はさらに進むと考えられ、柔軟で手軽な情

報表示手法は、情報空間とのインタラクションを支援するために必要な要素であると考えられる。

しかし、こうしたコンピュータ環境の向上に対して、情報群（情報のまとまり）を表示するためのレイアウト手法にはあまり変化が見られない。情報空間内での情報検索、プレゼンテーションのスライド群、コミュニケーションにおける情報交換や、3次元CGのコンテンツ生成において利用するデータ群など、様々な場面で情報群を扱う機会はあるが、こうした場合、ユーザはあらかじめデザインされたレイアウト手法を使わなければならない、柔軟で手軽な情報提示手法とはいえない。

情報群を効果的に表示する手法は、情報視覚化の分野において様々な研究がなされてきた。ズームングやスケールリングなどの手法を用いて情報を視覚化するこ

[†] 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
トリ

Sony Computer Science Laboratories Interaction Laboratory

^{††} 慶應義塾大学大学院理工学研究所開放環境科学専攻
Keio University

とにより、大量のデータを効果的に表示することが可能であり、大量情報からの目的情報の検索などを支援することに有効な手段である。しかし、こうした手法の多くが、レイアウトがあらかじめデザインされているため、あらかじめ想定された情報群や操作に対しては効果的であるが、情報を何気なく眺めることや、手軽に提示する用途には向いていない。また、情報の配置を3次元空間に拡張したシステムでは、情報量が増えるに従ってナビゲーション手法も必要になる。異なる形式のレイアウトを複数用意することで対応を考えたシステムも存在するが、同様に用途が限られているといえる。

一方で、OSがサポートするフォルダに見られる簡易なレイアウトが、情報群を扱う手法として一般的に使われている。こうしたレイアウトは様々なタイプの情報の閲覧に適用できるが、レイアウトが簡単すぎて情報ノード(画像や簡単な図形などを用いて視覚化された情報)の大きさや位置などをユーザの用途に応じて変更することや、複数の情報間に関連を付けることができない。結果として、手軽に使えて、かつユーザの意図を柔軟に反映させることができるようなレイアウトを提供することは困難であるといえる。

そこで、効果的なレイアウトを提供するのではなく、ユーザが手軽にレイアウトを作ることができる環境を提供することを考えた。今回提案する VelvetPath は、スケッチとペイント手法を用いたレイアウトデザインシステムであり、ユーザの描画したストロークやペイント領域に情報ノードが配置される点に特徴がある。本論文では、ストロークは描画操作により描かれた線を意味し、ペイント領域は描画操作により塗り潰された領域を意味する(図1)。操作がスケッチの手法でペンで紙に絵を描くような感覚で手軽にでき、矩形などの単純なレイアウトから、3次元レイアウトや階層的なレイアウトなど様々なレイアウトをデザインすることが可能である。また、ペン属性や太さにより、情報ノードの大きさやノード間の関連を直接調整することができるので、ユーザの注目や用途、コンピュータの形態に柔軟に対応できる。

デザインが自由にできる反面、デザイン作業自体が負担になる可能性がある。しかし、単純なストロークでさえレイアウトとなることに加え、既存のレイアウトを手本にして部分的に変更を加えることなどから、デザインの難しさはある程度回避されると考える。また、ユーザはすでにあるレイアウトにストロークを加えることでデザインし直すことができ、こうした作業を繰り返すことで自分にあったレイアウトを実現で

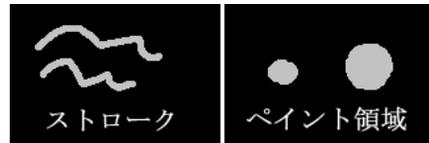


図1 ストロークとペイント領域
Fig. 1 Stroke and paint area.



図2 VelvetPath システム
Fig. 2 VelvetPath system overview.

きる。

こうしたレイアウト手法は情報検索だけではなく、3次元CGのコンテンツ生成支援、プレゼンテーションやコミュニケーションでも有効な手法であると考えられる。

本論文では、ユーザがストロークを描画することによりレイアウトをデザインするシステム VelvetPath と、その応用について述べる。

2. ユーザインタフェース

図2は VelvetPath システムの概要である。システムは描画を行う描画領域と、描画をサポートする GUI 領域に大別される。

GUI 領域には、ペン属性(基本(緑色)、影(黄色)や関連(白色)ストローク)を変更するボタンが含まれている。描画において、ストロークの色は GUI が示すボタンの色に対応しているため、ユーザは色によりストロークの種類を認識できる。また、ペンの太さを GUI 領域をクリックすることで直接変更でき、ペン GUI を左ボタンをクリックするとペンが細くなり、右クリックすると太くなる。ストロークの描画後に Set ボタンを押すことで情報ノードが配置され、Clear ボタンにより描画情報および、情報ノードのパラメータが初期化される。

2.1 基本ストローク

基本ストローク(緑色表示)はレイアウトを作る基



図3 基本ストローク
Fig. 3 Normal stroke.



図4 影ストローク
Fig. 4 Shadow stroke.

本要素として使われる。情報ノードはユーザが描画したストロークに沿って配置されるので、ユーザは描画領域の好きな場所や、すでに存在するオブジェクトの上に基本ストロークを描画することでレイアウトを作る。また、ペンの太さは情報の大きさに直接的に対応しているため、ユーザが太いペンで描画した場合には、大きな情報ノードが現れ、細いペンで描画した場合には、小さな情報ノードが現れる(図3(上))。

同様に、ペイント領域はシステムにより高さを持つ領域と見なされるので、その高さや幅が反映された情報ノードが表示される(図3(下))。また、ペイント領域に対しては、情報ノードの大きさを表す矩形が表示されるので、イメージする大きさと同じ領域をペイントしなくても大きさの設定が行える。

2.2 影ストローク

影ストロークは単体で使う場合と組み合わせて使う場合で役割が異なる。影ストロークのみを描画すると、情報ノードがペンの太さに対応した大きさに地面に配置される(図4(上))。したがって、Data Mountain²⁾のようなレイアウトもストロークの描画により実現可能である。また、基本ストロークを描画した後で影ストローク(黄色表示)を加えることにより、システム

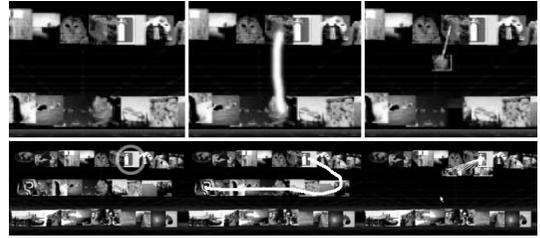


図5 関連ストローク
Fig. 5 Relation stroke.

は影ストロークを奥行き成分として見なすので、基本ストロークにより描画されたレイアウトは3次元レイアウトに自動変換される。図4(下)は、影ストロークの描画により3次元レイアウトに変換した例である。

2.3 関連ストローク

関連ストローク(白色表示)は、複数の情報間に関連を持たせるために使われる。このストロークではペンの太さは関連度に対応しているため、太さを考えながら情報ノード間を結ぶことで、関連度を調整することが可能である(図5(上))。また、注目ノードに対して一括して関連付ける機能を提供している。ある情報ノードをクリックし、そのノードと他の情報ノードとを関連ストロークでつなげ続けることで実現される(図5(下))。こうした機能は、ユーザ独自のデータ(デジタルカメラで撮影した画像や、デザインシステムで作った2次元や3次元データなど)を関連付け、整理する場合に利用できる。

このような関連付け手法により、情報視覚化に見られる階層的なレイアウトやグラフィックレイアウトをデザインすることが可能である。こうしたレイアウトでは、あらかじめ関連を設定するためのデータベースが必要となるが、本システムではストロークで結ぶだけでこうしたレイアウトが実現される。

2.4 レイアウトに対する操作

本システムでは、配置した情報ノードに対して、マウス操作とキー入力により情報ノードを移動させる機能を提供している。

ペイント領域に対して情報ノードを移動させることで、大きさを変化させ詳細を見ることが可能である(図6)。図6(上)では、レイアウトを行った後で空いた領域にペイント領域を描画し、そこに情報ノードを移動させることで情報ノードを拡大表示した例である。図6(下)では、情報ノードの位置を入れ替えている例である。これらの操作は簡単なデータベースとしてファイルに格納される。初期状態では、システムは指定されたフォルダから直接データを読み込み、レイ

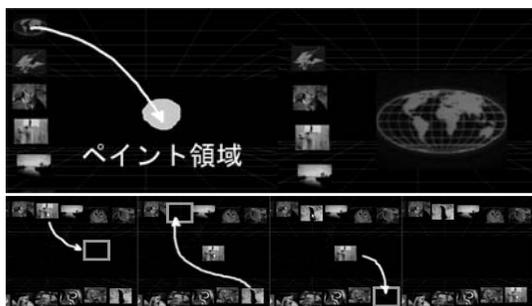


図 6 マウスによる操作
Fig. 6 Mouse interactions.



図 7 情報の移動
Fig. 7 Shift animation.

アウト上にファイル更新時間の新しい順に配置する。レイアウトに配置した情報ノードは、キー入力(矢印キー)によりシフト(レイアウト上の上下左右への移動)させることができ、ストローク上に表示できなかった情報ノードを閲覧することができる。図 7 はシフトアニメーションの例を示したものである。キー(矢印キー右)入力によって、情報ノードがレイアウト上を右にノード 1 つ分移動している様子が示されている(図 7(上))。また、キー(矢印キー上)入力によって、ストローク単位で情報ノード群が上に移動している様子が示されている(図 7(下))。こうした操作により、多量の情報をレイアウト対象として扱える。

3. 実装

システムは 3 次元空間内にスクリーンと地面の 2 つの描画面により構成され、ストロークの属性によりこの 2 つの描画面を使い分けている(図 8(左))。

基本ストロークと関連ストロークはスクリーンに対して描画され、影ストロークは地面に対して描画される。影ストロークは、スクリーン座標に対する 2 次元ストロークの描画を 3 次元シーンにプロジェクションしたものである。スクリーンに対する描画の地面への変換は見かけ上ずれがなく計算されるので、スクリー

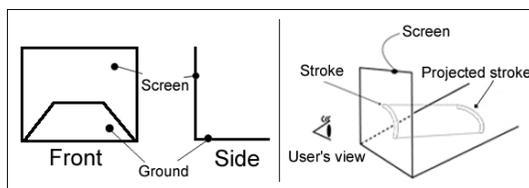


図 8 スケッチ手法
Fig. 8 Calculation of sketch manipulation.

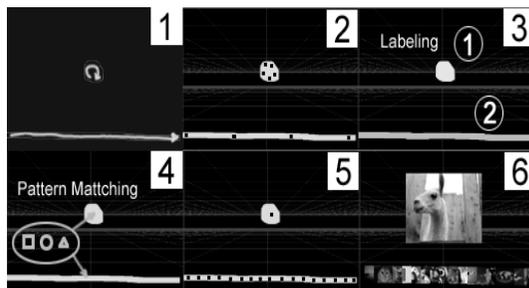


図 9 描画に対する計算
Fig. 9 Calculation to set data.

ン面に対する描画と同じストロークとして扱える。したがって、ユーザは 2 次元操作の感覚のまま地面への描画を行える(図 8(右))。

3.1 基本ストローク

ユーザが基本ストロークを描画すると、スクリーンの描画に対する画像データと、マウスの軌跡データの 2 つのデータが保持される(図 9(1), (2))。画像データに対してラベリング処理を行い、処理結果に対してパターンマッチングを行う。ペイント領域かストロークかを区別する簡単なパターンマッチングの結果から、描画エリアの形がペイント領域かどうか判断される。ペイント領域に対しては高さを持つ領域として認識されるので(図 9(3), (4)), ペイント領域の中心位置に高さや幅情報が反映された情報ノードが配置される。線と認識された場合には、ペンの太さに依存した間隔が設定され、その間隔を保つようマウスの軌跡データが再計算される(図 9(5))。情報ノードが再計算されたマウスの軌跡データ上に配置される(図 9(6))。

ストロークが交差した場合、描画データのラベリング結果がパターン(円形、または四角)に合わなければ線として処理され、レイアウト上に情報が配置される。図 10 では、パターンに照合する部分がないため、単なる線と見なされペンの太さに対応した大きさの情報ノードが配置されている。

ストロークの長さに対して配置する情報ノード数が多い場合には、ストロークの長さ分だけ情報が表示

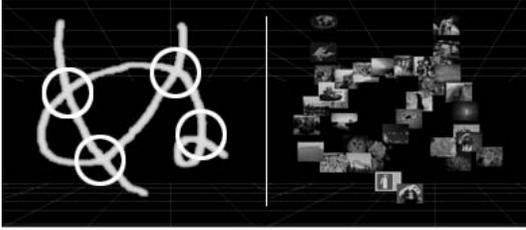


図 10 交差したレイアウト
Fig. 10 Cross layout.

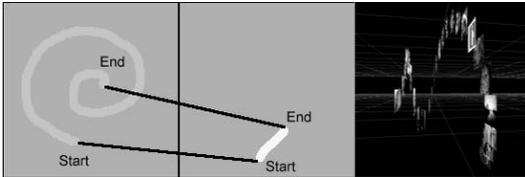


図 11 3次元レイアウト
Fig. 11 Implementation of shadow stroke.

され、残りの情報はキー入力により、移動させることで表示させることができる。これらの一連の計算は、GUI 領域に存在する Set ボタンが押されるたびに行われる。また、Clear ボタンにより画像データとストロークデータが初期化される。

レイアウトのためのストロークが描画されると、それぞれの情報ノードの変数 (ID, 位置, 大きさ, 貼り付け画像や形) が設定される。画像を扱っている場合には、対応する画像データが張り付けられる。これらの変数は情報ノード間の位置を交換する場合にも使われる。ユーザがクリックとドラッグの操作で情報ノードの位置を交換すると、システムにより位置に関する変数も書き換えられる。

3.2 影ストローク

影ストロークのみが描画された場合には、基本ストロークと同様の計算が行われ、地面への描画に対する描画データとストロークデータの計算により、ペンの太さをやポイント領域の大きさが反映した情報が配置される。

Set ボタンを押す段階で基本ストロークと影ストロークが存在する場合には、3次元レイアウトへ変換するモードに変わる。この場合、システムは基本ストロークの始点と終点を、影ストロークの始点と終点に対応付けることで、基本ストロークで計算したマウスの軌跡データを元に、奥行き計算を行う。結果として、基本ストロークが奥行き情報を持つ3次元ストロークに変換される (図 11)。

3.3 関連ストローク

関連ストロークによる関連付けでは、システムはス



図 12 階層レイアウト
Fig. 12 Implementation of relation stroke.

トロークの始点を親ノード、ストロークがつながった情報ノードを子ノードとしてシーングラフに保持する。システムはストロークの太さにより、親ノードに対する子ノードの大きさと位置を再計算する。図 12 は、関連ストロークによる親子関係の構築例である。ペンの太さが 2 であるため、子ノードの大きさが親ノードの半分になっている。このように、ユーザはシーングラフをストローク描画により直接的に構築することが可能である。同様のスケッチベースの関連付けテクニックはすでに存在しているが¹¹⁾、ペンの太さが関連度に直接対応している点で異なる。

こうしたストローク描画による情報の操作は、ノード情報 (名前, 順序, 位置, 関連やテキストなど) とレイアウト情報 (ストロークの番号, 位置や太さなど) の 2 つに分類され、簡単なテキストファイルに保存される。たとえば、関連ストロークの場合、システムは親と子ノードの名前と、それぞれの関連度をこのファイルに保存する。このファイルは、テキストファイルなので、ユーザがテキストエディタで編集可能である。システムはこのデータファイルがある場合には、最初に読み込み、情報を表示するための変数を設定する。したがって、ファイルの内容から以前の操作を反映させることが可能であり、関連や位置の入替えなどのユーザの操作がレイアウトに反映される。

4. システム適用例

システムを 3次元 CG のコンテンツ生成支援、プレゼンテーションやコミュニケーションに応用した例を以下に示す。

4.1 レイアウトデザイン

図 13 は既存の情報視覚化システムに見られるレイアウトを、VelvetPath システムにより作ったものである。図 13 (下) は図 13 (上) に示したレイアウトが、どのように作られたかを示している。

図 13 (1) では、影ストロークを直接地面に描画することで、地面に対して情報ノードが配置されるので、DataMountain²⁾ のようなレイアウトが構築されている。図 13 (2) はペンの太さに依存したテキストレイ

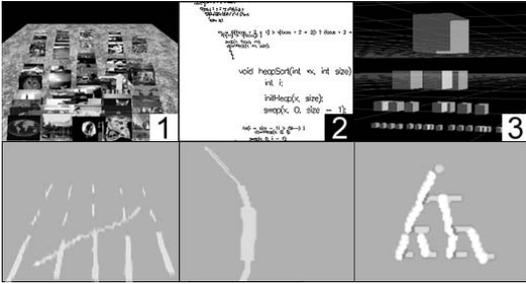


図 13 レイアウト例

Fig. 13 Examples of layout design.

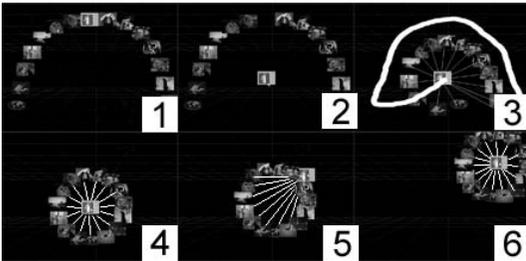


図 14 グラフレイアウト

Fig. 14 Graph layout.

ウトの例である．中心付近に太いストロークを描画し，上下に遠ざかるに従ってストロークの太さを細くすることで注目部分をはっきりさせ，多くの情報を見せることを目的としている．また，ConeTree⁶⁾ のような階層的なレイアウトも，情報ノード間を関連ストロークで結ぶことで親子関係を作り，位置と大きさを反映させることで実現されている（図 13 (3)）．

関連ストロークをユーザが使う場合，その太さは情報ノードの大きさに対応しているが（図 12），この機能を拡張しグラフレイアウトに適用した応用例を図 14 に示す．この場合，関連ストロークの太さは，情報の大きさではなく情報間に働く力の大きさに対応している．ユーザがノード間を関連ストロークでつなぐと，システムにより親子関係が形成され，ノード間につながっていることを示す白い線が表示される（図 14 (1)～(4)）．これらの情報ノードは描画フレームごとに距離に比例した引力斥力の計算が行われるので，ユーザがあるノードを移動させると，関連するノードも注目ノードから力を受けて動く（図 14 (5)，(6)）．このように，ペン属性の組合せと，ペンの太さの調整によりレイアウトを比較的簡単に作ることが可能である．

4.2 3次元 CG のコンテンツ生成支援

3次元 CG のコンテンツ生成においてストローク描画やペイント操作は一般的なペンの機能で行えるので，

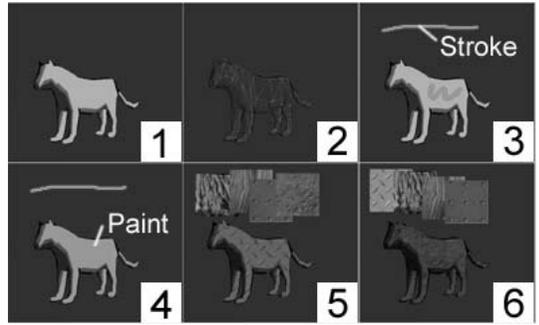


図 15 テクスチャマッピング

Fig. 15 Texture mapping.



図 16 シーンレイアウト

Fig. 16 Scene layout.

そうした操作と本システムを結び付けることで，データやパーツを試す作業を支援できると考える．

4.2.1 テクスチャマッピング

図 15 は本システムを用いて，3次元モデルに対してテクスチャマッピングを行っている例である．モデル上でテクスチャを張りたい部分を塗りつぶし，さらに，モデルの上部にストロークを描画する（図 15 (3)，(4)）．マップ候補のテクスチャがストローク上および，モデル上に現れるので，ユーザはデータをシフトさせることで，候補を手軽に試すことができる（図 15 (5)，(6)）．一般にテクスチャマッピングを行う場合は，1回ごとにデータの張り替え操作を行わなければならないが，本手法ではキー入力のみでデータを切り替えることが可能である．また，テクスチャだけではなくシェーダやモデル構築におけるパーツを試すことなどにも適用可能であると考えられる．

4.2.2 シーンレイアウト

同様の方法で，3次元シーンにオブジェクトをレイアウトしている例を図 16 に示す．この場合も，ストローク描画と，実際のモデルの大きさに対応する領域をペイント操作で作ることにより，モデルの候補をシーンに直接配置することができる．テクスチャマッピング同様，データをシフトさせる操作だけで候補を確認できる．こうした手法は，複雑なパラメータ設定や GUI の操作を介さないため，CG 初心者の製作作業を強力に支援しようと考えられる．

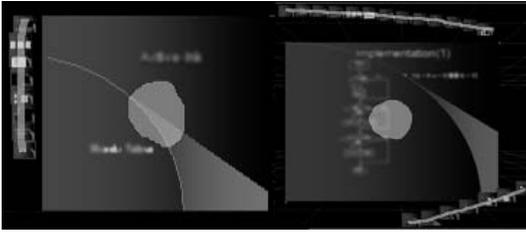


図 17 プレゼンテーション

Fig. 17 Presentation with VelvetPath.

4.3 プレゼンテーション

レイアウトを作りスライドを配置することにより、プレゼンテーションを支援することも可能である。図 17 のレイアウトでは、現在のスライドを大きく、その他のスライドを小さく表示することで、すべてのスライドを表示している例である。プレゼンテーションにおいて、現在のスライドと、前後のスライドの関連を知ることは、視聴者にとって内容の理解を助けるうえで重要である。したがって、図に示したレイアウトは、視聴者がスライド間の関係を理解する手助けになると考える。こうした例は、本システムのデモンストレーションを通じて得られたコメントをシステムに反映させたものである。加えて、近年デザイナーを含めた多くのユーザが、Director や Flash¹⁷⁾ などのソフトウェアにより、プレゼンテーションシステムを自分で作る傾向にある。しかし、こうしたシステムを使えるようになるためには多くの時間を必要とし、誰でも使えるとはいえない。

また、本システムは、ちょっとしたミーティングにおいても、データを表示する手法として有効であると考えられる。ミーティングの状況や、データの種類にあわせて、その場でレイアウトを変えることで意見交換などの手助けになると考える。

4.4 コミュニケーション

実世界指向インタフェース^{9),10),16)}などのコンピュータを介したコミュニケーションにおいても本手法は有効であると考えられる。こうしたシステムにおいても、他のユーザに情報を見せることや、コンピュータ間で情報を交換する場合などに、情報群のレイアウト手法は重要な要素となる。しかし、多くの実世界指向インタフェースにおいて、インタラクション手法は工夫されているが、情報の表示手法にはあまり焦点が当てられていない。

図 18 は Pick and Drop システム⁹⁾と本システムとの組合せ例を示したものである。ユーザは最初に情報を表示するためのストロークを描画し、次に表示された情報を Pick and Drop の操作により交換する。加え



図 18 コミュニケーション

Fig. 18 Communication with VelvetPath.

て、ストロークを直接交換対象と扱うことで、情報群をまとめて移動させることも可能である。また、ジェスチャーインタラクションとの組合せでは、指の動きでストロークを描くことにより、レイアウトが実現される。

こうしたシステムでは、情報の識別がインタラクションをサポートするうえで重要であり、情報に ID を振ることでコンピュータ間での情報交換を実現している。本システムの情報ノードはレイアウトが終わった際に ID などの情報が設定されるため、こうしたシステムと組み合わせるうえで有効利用できると思われる。

5. 考 察

本システムのインタラクション手法と、アプリケーションについて、ユーザテストとデモンストレーションでの知見を基に考察を行う。また、本研究の関連研究についても述べる。

5.1 ユーザテスト

VelvetPath を情報検索とクリエーション作業に適用したユーザテストについて述べる。本システムと他のシステムとの比較実験を行い、その結果を示す。

5.1.1 実験内容

本システムを用いて、ランダムな画像群での検索、似た画像群での検索および、テクスチャマッピングの 3 種類のテストを行った。また、それぞれのテストの後で、ユーザにシステムの操作性について 5 段階の評価をしてもらった(5: 良い~1: 悪い)。すべてのテストにおいて、ユーザはまずレイアウトをデザインした後で、テストを行う。その際、レイアウトストローク上にテストで用いる画像は表示しない。また、すべての実験においてウインドウの大きさはすべて同じ(横: 800 pixel 縦: 600 pixel)である。

テスト 1, 2 では、Windows でサポートされているフォルダと VelvetPath による情報検索の比較を行い、50 枚の画像から指定された画像を検索する時間を計った。テスト 1 では、素材集から抜粋した画像 50 枚からの検索を行う。この画像群は描画されている絵の構



図 19 フォルダ
Fig. 19 Folder.

成がすべて異なっている画像群である(図 19(左)). テスト 2 では, 似通った画像 50 枚からの検索を行う. この画像群は背景が同じで, 描画されている文字のみが異なる画像群である(図 19(右)). また, テスト 3 では, 指定された位置に対してテキストをマッピングする作業を行った. Photoshop¹⁸⁾, Maya¹²⁾ と VelvetPath により, 10 種類のテキストをマッピングする作業を行い, 合計の時間を計った. マッピング対象は, 3 次元ソフトが比較対象として含まれているため, 本システムにおいても図 15 に示す 3 次元モデルへのマッピングを行った.

5.1.2 被験者

VelvetPath システムを初めて使うユーザ 6 人に各テストを 5 回ずつ行ってもらった. ユーザはコンピュータに関する簡単な操作はできるが, デザインに対する特別な技術は持っていない. ユーザテストの前に, 本システムと比較対象のシステムの使い方や特徴について, それぞれ 5 分間の学習時間を与えた.

5.1.3 結果

テストの結果を図 20 に示す. 作業時間の平均をとったものを図 20(a) に, 作業時間の標準偏差を図 20(b) に, ユーザの操作性に対する評価の平均を図 20(c) に示す.

作業時間の平均(図 20(a))から, テスト 1 において, ユーザは本手法を用いて描画空間により多くの情報を表示することを考えレイアウトをデザインしたことで(図 21(1), (2)), 検索時間がフォルダに比べて早かったと考えられる. また, テスト 2 においては, 本システムでは, ペイント領域やペンの太さにより情報の大きさを変更することが可能であり, ペイント領域をレイアウト内に設定することで(図 21(3)), データの詳細を見ることができ, 似通ったデータから目的のデータを早く選び出せたと考える. テスト 3 においても, すべてのユーザが本システムにより作業を早く行えた. ストロークの描画やペイント領域により, データを直接作業空間に配置しシフトさせるだけで試すことができるので, 他の手法より効果的な結果が得

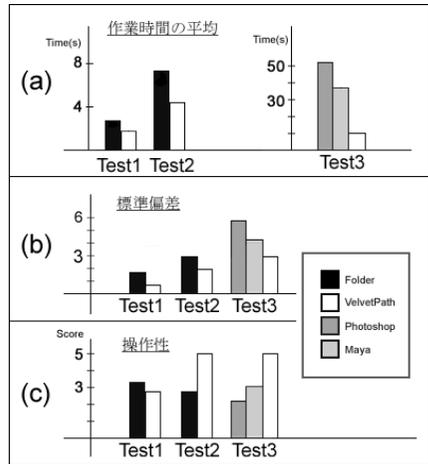


図 20 テスト結果
Fig. 20 Results of user experiments.



図 21 テストでのレイアウトデザイン
Fig. 21 Layout design in user experiments.

られた.

作業時間に加え, 標準偏差(図 20(b))も VelvetPath を使った場合が最も小さくなっている. これはユーザによる作業時間のばらつきが小さく, 平均的に早く作業を行うことができたことを示している. また, 各実験間の相関について有意水準 5% の下で統計的検定を行ったところ, すべての実験において, VelvetPath を使った場合に統計的な有意差が見られた. こうした結果は, 実験において VelvetPath システムが比較対象のシステムよりも効率であったこと示唆している.

それぞれの実験での操作性に対する評価(図 20(c))も, 作業の効果に比例して良くなった. 特に, テスト 3 において, 作業効率に加え, すべてのユーザが良い(5)の評価をしたことから, テキスチャマッピングなどのデータを試す作業で有効であると考えられる.

レイアウトデザインにおいて, ほとんどのユーザが, 情報検索の前にオリジナルのレイアウトをデザインする姿が観察された. 複雑なレイアウトを作るユーザはいなかったが, 簡単なレイアウトを即座に作っていた. こうした様子から, システムのコンセプトやインタラクション手法は比較的簡単に理解されたと考える. また, 学習の段階で, 文字や絵を描いてそこに情報を配

置するユーザも観察された。このように、ユーザは簡単で(怪しい)レイアウトを特別な技術なしで作ることができるので、デザインに関する難しさはある程度回避されると考えられる。本システムでは、レイアウトにさらにストロークを加え、デザインし直すことが可能であり、こうした操作の繰返しにより、ユーザがより効果的なレイアウトを獲得できることを目標としている。実際に、実験中に最初のレイアウトにストロークを加えて、デザインし直し実験に臨むユーザが多かった。

5.2 デモンストレーションからの知見

これまで学会などで、スケッチとペイントの手法により情報ノードを直接操作する手法を見せてきた。レイアウト手法に加え、自分のオリジナルデータに対して、関連度を付加できる関連ストロークに好感を持つ人が多かった。一般に、自分のオリジナルデータはフォルダに関連する名前を付けて一括管理されるが、本手法によりユーザの興味に依存した関連を持たせることができる。一方で、本手法が細かい操作をサポートしていない点に対する質問もあった。本システムでは、手軽な利用を目的としているため、操作を難しくする複雑で詳細な設定をサポートしていない。

デモンストレーションを通じて、訪問者は多くの応用例をイメージし、ユニークなコメントを提供してくれた。特に、実世界指向インタフェースシステムとの組合せに対するコメントは好意的なものが多かった。手軽にレイアウトを作ることができる本手法と、そうしたシステムとの組合せは有効であると考えられる。加えて、本手法は OS に依存せず、特別な入出力デバイスを必要としないので、その適用範囲は広いといえる。

また、本手法が動画の編集に使えるとのコメントも多かった。ストロークを直接時間軸として使うことにより、ストローク上に各フレームデータを表示し、目的に応じて入れ替えることが可能である。この場合も、ペイント領域によるデータの拡大手法はデータの詳細を見るために役立つと考えられる。

5.3 関連研究

システムは、スケッチとペイントの操作で情報を表示するためのレイアウトを作る点に特徴がある。したがって、情報視覚化とスケッチ手法の関連研究について述べる。

5.3.1 情報視覚化

情報検索を支援する情報視覚化システムが数多く存在する¹⁾。2次元や3次元空間に大量の情報を表示することや、注目ノードや関連ノードを目立たせるために、スケーリングやズームングなどの手法が使われて

いる^{3)~5)}。こうしたシステムでは情報のレイアウトはあらかじめデザインされていて、想定された状況では有効であるが、それ以外の用途では制約が多い。3次元空間を対象とすることで、位置情報と時間情報といった異なる情報を、同時に表示することができるシステムも存在する^{6),7)}。しかし、3次元空間に情報を配置する場合、効果的なナビゲーション手法も必要になる。パネモデルを用いて対話的にレイアウトを変更できるシステムも存在し⁸⁾、情報間の関連を知るために有効手法であるが、あらかじめ何らかのデータベースに関連を記述しておく必要がある。

5.3.2 スケッチインタフェース

2次元の操作で3次元クリエーションを可能にする、スケッチインタフェースも数多く研究されている。紙にペンで字を書くような手軽な操作で3次元空間と対話できる点に特徴がある。Sketch¹³⁾のユーザは2次元の操作で3次元の曲線を描画するシステムであり、2次元ストロークに影をつけることで、3次元曲線を生成する。Harold¹⁴⁾はスケッチの手法で、3次元シーンに2.5次元的なシーンを構築する。Teddy¹⁵⁾は3次元モデリングシステムで、閉曲線に対してポリウムを自動計算することで、3次元モデルを生成している。これらのシステムではマウスの軌跡情報のみが使われていて、ペイント領域やペンの太さ情報は使われていない。

6. まとめ

ストローク描画により、ユーザがレイアウトを作ることができる VelvetPath システムについて述べた。ペンの太さや属性を組み合わせることで、様々なレイアウトを手軽にデザインできる例を示した。また、システムの柔軟さを示す様々な応用例(3次元CGのコンテンツ生成支援、プレゼンテーションやコミュニケーション)を示し、最後にユーザテストの結果やデモンストレーションでの知見から有効性について述べた。

今後は、レイアウトデザインを支援するための新たなペン属性を加えることや、情報を閲覧するための効果的なアニメーション手法の実現を考えている。また、本手法を映像編集に適用し、有効性を探っていくつもりである。

参考文献

- 1) Card, S.K., MacKinlay, J.D. and Shneiderman, B.: *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann (1999).
- 2) Robertson, G., Czerwinski, M., Larson, K.,

- Robbins, D., Thiel, D. and van Dantzich, M.: Data Mountain: Using spatial memory for document management, *Proc. UIST '98*, pp.153-162 (1998).
- 3) Koike, H.: Fractal views: a fractal-based method for controlling information display, *ACM Trans. Inf. Syst. (TOIS)*, Vol.13, No.3, pp.305-323 (1995).
- 4) Furnas, G.W.: Generalized fisheye views, *Proc. ACM Trans. Computer-Human Interaction*, Vol.1, No.2, pp.126-160 (1994).
- 5) Bederson, B.B., Hollan, J.D., Perlin, K., Meyer, J., Bacon, D. and Furnas, G.: Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics, *Journal of Visual Languages and Computing*, Vol.7, No.1, pp.3-31 (1996).
- 6) Robertson, G.G., Mackinlay, J.D. and Card, S.K.: Cone Trees: Animated 3D Visualization of hierarchical information, *Proc. ACM Conference on Human Factors in Computing Systems (CHI '91)*, pp.189-194 (1991).
- 7) Koike, H. and Takada, T.: VisuaLinda: A Framework for Visualizing Parallel Linda Programs, *Proc. 1997 IEEE Symposium on Visual Languages (VL'97)*, pp.174-180 (1997).
- 8) Davidson, R. and Harel, D.: Drawing Graphics Nicely Using Simulated Annealing, *ACM Trans. Graphics*, Vol.15, No.4, pp.301-331 (1996).
- 9) Rekimoto, J.: Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments, *Proc. UIST'97*, pp.31-39 (Oct. 1997).
- 10) Koike, H., Sato, Y., Kobayashi, Y., Tobita, H. and Kobayashi, M.: Interactive textbook and interactive Venn diagram: natural and intuitive interfaces on augmented desk system, *Proc. CHI2000*, pp.121-128 (2000).
- 11) Lin, J., Thomsen, M. and Landay, J.A.: A Visual Language for Sketching Large and Complex Interactive Designs, *Proc. CHI2002*, pp.307-314 (2002).
- 12) Maya and PaintEffect.
<http://www.aliaswavefront.com/>
- 13) Zeleznik, R.C., Herndon, K.P. and Hughes, J.F.: An Interface for Sketching 3D Curves, *Proc. SIGGRAPH '96*, pp.163-170 (1996).
- 14) Cohen, J.M., Hughes, J.F. and Zeleznik, R.C.: Harold: A World Made of Drawings, *NPAR2000 (Symposium on Non-Photorealistic Animation and Rendering)*, pp.83-90 (2000).
- 15) Igarashi, T., Matsuoka, S. and Tanaka, H.: Teddy: A sketching interface for 3D freeform design, *Proc. SIGGRAPH '99*, pp.409-416 (1999).
- 16) Rekimoto, J. and Saitoh, M.: Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments, *Proc. CHI'99*, pp.378-385 (1999).
- 17) Director and Flash.
<http://www.macromedia.com/>
- 18) Photoshop.
<http://www.adobe.com/>

(平成 15 年 4 月 10 日受付)

(平成 15 年 9 月 5 日採録)



飛田 博章

1999 年電気通信大学大学院情報システム学専攻博士前期課程修了。同年ソニー株式会社入社。クリエイションとコミュニケーションを支援する、インタラクティブ 3 次元コンピュータグラフィックスに関する研究に従事。また、そうした研究の情報視覚化や実世界指向インタフェースへの応用に興味を持つ。



磯 大輔

2001 年慶應義塾大学理工学部情報工学科卒業。2003 年同大学大学院理工学研究科開放環境科学専攻修士課程修了。同年ソニー株式会社入社。在学中は画像処理、コンピュータビジョンの研究に従事。2002 年電子情報通信学会学術奨励賞受賞。



暦本 純一 (正会員)

1986 年東京工業大学大学院情報科学専攻修了。日本電気、カナダアルバータ大学を経て、1994 年より(株)ソニーコンピュータサイエンス研究所に勤務。現在、同研究所インタラクショナルラボラトリー室長。理学博士。ヒューマンコンピュータインタラクション全般、特に実世界指向インタフェース、拡張現実感、情報視覚化等に興味を持つ。ACM、日本ソフトウェア科学会会員。1990 年情報処理学会 30 周年記念論文賞、1998 年 MMCA マルチメディアグランプリ技術賞、1999 年情報処理学会山下記念研究賞受賞。