

## Spiral Tree : 順序付き階層構造の3次元視覚化手法

大隈 隆史<sup>†</sup>      竹村 治雄<sup>†</sup>      岩佐 英彦<sup>†</sup>      片山 喜章<sup>††</sup>  
横矢 直和<sup>†</sup>

### Spiral Tree: 3-D Visualization of Hierarchical Ordered Information

Takashi OKUMA<sup>†</sup>, Haruo TAKEMURA<sup>†</sup>, Hidehiko IWASA<sup>†</sup>, Yoshiaki KATAYAMA<sup>††</sup>,  
and Naokazu YOKOYA<sup>†</sup>

あらまし 近年の情報の大規模化と多様化から、大規模な情報をユーザにわかりやすく見せるための視覚化手法が注目を集めている。階層構造情報の3次元視覚化手法 Cone Tree は、3次元コンピュータグラフィックスとアニメーションを用いて大規模な情報のブラウジングを支援するアプリケーションとして知られている。ユーザが階層構造をブラウジングするときの手掛りを増加させるために Cone Tree を拡張する手法がいくつか提案されているが、ブラウジングの際にユーザが頻繁に利用する階層内の順序を効果的に視覚化する手法は存在しない。そこで、各階層を円周上ではなくら旋上に配置する階層構造情報の3次元視覚化手法 Spiral Tree を提案する。提案手法の有効性を評価するためにノード探索タスクによる実験を行い、Spiral Tree は Cone Tree より短時間でタスクを達成することができるという結果を得た。また、Spiral Tree でのブラウジング時に、階層内の順序が実際にどの程度利用されているかを確認するための実験を行い、統計的な有意差は確認できなかったが、順序情報の利用を示唆する結果を得た。

キーワード 3次元ユーザインタフェース, 情報視覚化, 階層構造

#### 1. ま え が き

World Wide Web の急速な普及や電子図書館の実現などにより、図書館のような広い場所を用意しなくても、デスクトップ上で大規模な情報源からさまざまな情報を利用することが可能となってきた。しかし、利用可能な情報の大規模化と多様化は、ユーザが必要な情報を見つけ出すことを困難にしつつある。

このため、キーワード検索を用いてユーザが検討する情報の候補の数を減少させることがよく行われる。しかし、キーワード検索で得られるすべての情報がユーザの興味に深く関連するとは限らない。また、ユーザが自分の興味をキーワードとして具体的に表現できない場合には、キーワード検索を用いたの情報検索は困難となる。

そこで、ユーザが自分の興味に深く関連すると予想

される情報を直感的に見分けて検討するための手法が望まれる。このため、大量の文書間の関係や組織の階層構造のような大規模情報の複雑な構造をユーザにわかりやすい形で視覚化する手法が注目を集めている。

例えば、FilmFinder [1] ではスライドバーなどのグラフィカルユーザインタフェースを用いてパラメータを動的に設定し、このパラメータを用いた検索結果を、リアルタイムに視覚化することで、検索候補の素早い削減を可能にしている。また、Galaxy of News [2] では階層的に分類された情報を仮想空間に提示する。初期状態では大まかな分類名のみが表示され、ユーザが興味をもつ分類名に視点を近づけることで、一つ下の階層に属する分類名が表示される。これを繰り返すことで、視点の移動による直感的なブラウジングを行うことができる。Information Visualizer [3] は、3次元コンピュータグラフィックスとアニメーションを用いて複雑な情報の構造を視覚化し、ユーザにわかりやすく表示するシステムである。Information Visualizer のアプリケーションとして、線形情報を視覚化する Perspective Wall [4]、階層構造を視覚化する Cone Tree [5]、参考文献間のリファレンスの関係

<sup>†</sup> 奈良先端科学技術大学院大学情報科学研究科, 生駒市  
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0101 Japan

<sup>††</sup> 奈良先端科学技術大学院大学情報科学センター, 生駒市  
Information Technology Center, Nara Institute of Science and Technology, Ikoma-shi, 630-0101 Japan

を視覚化する Butterfly visualizer [6] などが知られている。

以上のように、さまざまな構造をもつ情報、若しくは情報の検索結果を視覚化することで、ユーザが必要な情報を捜し出すための手掛りを提供する手法が提案されている。本論文では、視覚化される情報の構造として階層構造に注目する。階層構造は計算機のファイルシステムや電子図書館における書籍の管理など、大規模情報の管理によく用いられる構造であるため、階層構造情報をユーザにわかりやすく視覚化する手法は重要であると考えられる。

階層構造情報を3次元的に視覚化する手法としては Cone Tree が知られている。また、検索に用いるキーワードとの関連度を Cone Tree に反映する視覚化手法 [7] や、1000 ノード以上の大規模な階層構造情報を視覚化しても、画面が煩雑にならないようにするために、Cone Tree のノード配置方針の改良や色情報の利用 [8] など、Cone Tree をもとに、検索の手掛りを増大させる手法が提案されている。本研究では、各階層内の情報の順序に着目し、Cone Tree のノード配置方針に階層内の順序を反映した手法 Spiral Tree を提案する [9], [10]。各階層内の順序情報は検索の手掛りとして階層構造のブラウジング時に頻繁に用いられるため、順序情報の明示的な視覚化はユーザのブラウジングに有効であると考えられる。

以降、2. では順序付き階層構造の3次元視覚化手法 Spiral Tree について述べる。3. では Spiral Tree の有効性を確認するために行った Cone Tree と Spiral Tree の比較実験について述べ、実験結果を検討する。4. で考察を加え、5. でまとめと今後の課題について述べる。

## 2. Spiral Tree

本章では、本論文で提案する順序付き階層構造の3次元視覚化手法 Spiral Tree について述べる。まず、Spiral Tree の基になっている階層構造の3次元視覚化手法 Cone Tree について述べる。次に、Cone Tree に対する階層内順序情報の追加手法について検討し、採用した Spiral Tree についてその特徴と実現方法について説明する。

### 2.1 Cone Tree : 階層構造の3次元視覚化手法

Cone Tree は一つの情報をノードと呼ばれる長方形のプレートで視覚化する。そして、情報の親子関係を半透明の3次元の円すいで視覚化し、親を円すいの頂

点、子を円すいの底面の円周上に均等に配置する。この円すいを階層的に配置することで階層構造を視覚化する (図 1)。すなわち、あるノード  $n$  の位置を  $p_n$ 、 $n$  の子が属する階層を視覚化する円すいの半径を  $r_n$ 、円すいの高さを  $l$  とし、 $n$  が  $m$  個の子ノードをもつとき、 $n$  の  $i$  番目の子ノードの位置  $p_i$  は以下のように表される。

$$p_i = p_n + \begin{pmatrix} l \\ -r_n \sin\left(\frac{i \times 2\pi}{m} + \alpha_n\right) \\ r_n \cos\left(\frac{i \times 2\pi}{m} + \alpha_n\right) \end{pmatrix}$$

但し、円すいの軸上を頂点から底面に向かう方向を  $x$  軸の正の方向とする。 $r_n, l$  の各々は、階層構造全体が視覚化されたときにディスプレイの表示領域に収まるように計算して決定される。 $\alpha_n$  はノード  $n$  の子の階

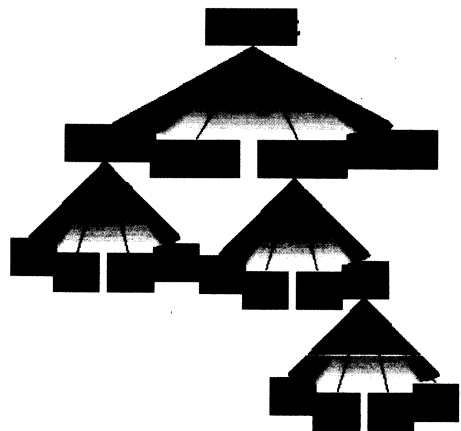


図 1 Cone Tree  
Fig.1 Cone Trees.

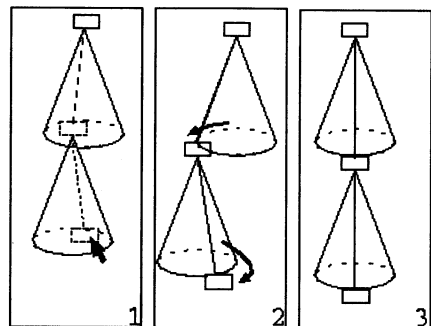


図 2 ノード選択時の回転アニメーション  
Fig.2 Animation of rotation in selecting a node.

層の回転角であり、ユーザのブラウジング時に動的に変化する。

このようにして視覚化された Cone Tree の任意のノードをマウスで選択すると、選択されたノードは画面手前側にアニメーションを伴って移動する。このとき、各階層の円すいが独立に回転して、選択されたノードからルートノードまでのパス上のすべてのノードが画面手前側に移動する (図 2)。すなわち、ノード  $n$  の  $i$  番目の子が選択された場合、

$$\frac{i \times 2\pi}{m} + \alpha_n = 0$$

となるように  $\alpha_n$  が変化する。

また、Cone Tree では必要に応じてマウスカーソルによるポインティングにより指示された階層以下についての表示/非表示を切り換えることができる。これは、ユーザにとって興味がない階層が他の階層をブラウジングするときの邪魔にならないようにするためである。

Cone Tree の特徴はアニメーションと 3 次元透視投影表示を積極的に利用している点にある。アニメーションと 3 次元透視投影表示の利用については、それぞれ以下の利点が挙げられる。

- ノードが選択されたことによって起こるノードの移動をアニメーションにより補間することで、ユーザが移動後の構造を再認識して心内で再構築する必要がなくなる。
- 透視投影による 3 次元的な表示は、視点に近いほど画面に大きく表示されることになり、自然に魚眼視野効果を実現する。

このように Cone Tree は階層構造のブラウジングに 3 次元透視投影表示とアニメーションをうまく適用しているが、ユーザが実際に階層構造のブラウジングを Cone Tree を用いて行う場合に、階層内の順序を利用しづらいという問題がある。階層内の順序とは、例えばファイル構造では「ファイル」という情報に対する「ファイル名」「ファイルサイズ」などの属性をもとにして付けた順序のことである。各階層内の情報に順序を付けた階層構造を順序付き階層構造と呼ぶ。ユーザは、ある情報を検索するとき、この順序を手掛りに探索を行うことが多いと考えられる。しかし、Cone Tree では同一階層のノードを円周上に並べるので、各階層内の情報のある順序に従って整列しても、先頭と末尾の位置がわかりづらい。このため、ユーザは同一階層内のノードの順序を探索の手掛りとして十分に利

用することができない。これは Cone Tree のノード配置方針が階層内の順序を十分に視覚化できていないことが原因である。よって、ユーザは順序付き階層構造情報の探索に不便を感じる可能性がある。

## 2.2 Spiral Tree

### 2.2.1 順序の利用

Cone Tree において情報探索過程で階層内の順序が利用しづらいという問題点は、各階層のノードを円周上に配置するという配置方針のために階層内の先頭ノードと末尾ノードの位置が明確でないことが原因である。そこで、階層内の先頭と末尾の位置を明示する手法について検討する。これには、

- (1) 末尾と先頭の間に切れ目を入れる、
- (2) 先頭ノードの色を変えて表示する、
- (3) 各階層のノードをら旋上に並べる、

などの方法が考えられる。これらの手法のうち、(1) の手法は Cone Tree の外見をほとんど変えることなく実現できるが、階層内のノード数が増えたときに切れ目が目立たなくなるため、ユーザに対して「切れ目を探す」という負荷を与えることになるという問題がある。(2) の手法については、順序関係を色空間で視覚化する最も有効な指針が明確でないため、実現が困難である。そこで、本研究では (3) の方法を採用することとした。

各階層のノードをら旋上に並べる方法で階層の先頭と末尾を明示する Cone Tree 型の階層構造視覚化手法を Spiral Tree と呼ぶ (図 3) [9],[10]。

Spiral Tree は Cone Tree に比べて、以下の利点をもつと考えられる。

- ら旋配置により順序の利用が容易になる、
- 円すいの軸方向へのずれにより同一階層内のノード

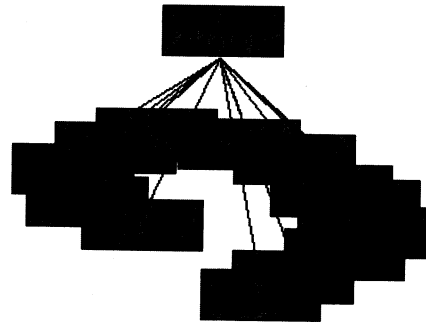


図 3 Spiral Tree の一階層  
Fig.3 A layer of Spiral Tree.

ド同士の重なりが減少するため、注目している階層についての一覧性が増す。

### 2.2.2 ノードの配置

Spiral Treeのノード座標はCone Treeのノード座標を、円すいの頂点から底面に向かう軸方向に平行移動して求められる。すなわち、ノード  $n$  の子ノードの先頭と末尾のノードの  $x$  座標の差を  $d$  とし、ノード  $n$  の  $i$  番目の子ノードの位置  $\mathbf{p}_i$  は以下のようにして計算される。

$$\mathbf{p}_i = \mathbf{p}_n + \begin{pmatrix} l + \frac{i \times d}{m} \\ -r_n \sin\left(\frac{i \times 2\pi}{m} + \alpha_n\right) \\ r_n \cos\left(\frac{i \times 2\pi}{m} + \alpha_n\right) \end{pmatrix}$$

ここで、階層間距離  $l$  に対してずれの距離  $d$  が小さいと先頭と末尾の位置が明確になりにくい ( $d = 0$  のときCone Treeとなる)。また、大きすぎるとSpiral Tree全体が画面内に収まらない上、見づらくなる。よってこれらの視覚化パラメータは適切な大きさに設定する必要がある。しかし、Spiral Tree全体が最もわかりやすくなる視覚化パラメータは視覚化の対象となる階層構造の一つの階層内のノード数や階層構造全体の深さなどにも依存し、すべての階層構造をわかりやすく視覚化する最適値を決定することは困難である。そこで、Spiral Treeを用いるアプリケーションプログラムなどにおいてはユーザによるこれらの視覚化パラメータの設定を考慮する必要がある。

次章では、Spiral Treeの特徴を調べるために行った、二つの評価実験について述べる。

## 3. 評価実験

本章では、階層構造情報の探索に関するSpiral TreeとCone Treeの比較実験と、Spiral Treeにおいて順序情報が有効に利用されているかを確認する実験について述べる。

### 3.1 実験1: Cone TreeとSpiral Treeの比較

Spiral Treeのノード配置アルゴリズムがCone Treeのノード配置アルゴリズムの改良になっていることを確認するために、同じ階層構造情報を視覚化したSpiral TreeとCone Treeを用いて、特定の情報を選択する作業について、作業時間と操作数を比較した。

[実験タスク] 実験タスクにはノード探索タスクを用いた。ノード探索タスクとは以下の手順を1試行とする実験タスクである。

(1) 被験者に目標ノードをフルパス名で指定する。(フルパス名とはルートノードから目標ノードまでのパス上に存在する各階層のノード名を“/”記号を挟んでつなげた文字列とする)

(2) 被験者は視覚化された階層構造を操作して目標ノードを選択(マウスでクリック)する。試行中も画面上部に目標ノード名を表示する。

ノード探索タスクはブラウジングによる情報探索作業を想定している。ノード探索タスクのタスク達成時間やタスク達成に必要な作業数は視覚化された階層構造情報の見やすさや操作性の影響を受けると考えられるため、これらを分析することで、視覚化された階層構造情報の見やすさや操作性について調べることができる。

[実験手順] 実験用Cone TreeおよびSpiral Treeを使用して階層構造情報を被験者に提示した。実験で視覚化された階層構造情報は被験者の記憶力や背景知識の影響を受けないようにするために、階層構造としての意味をもたないものとした。ノードには順序を付けるための属性として名前を与え、同一階層内のノードは名前によって辞書式順序で並べた。ノードに付した名前は英文から無作為に抽出した単語である。被験者はこの階層構造に対してノード探索タスクを行う。

被験者は両方の視覚化手法を用いて作業を行った。ノード探索タスクは、実験前に各々の視覚化手法を用いて300試行ずつ練習し、本実験では各々100試行ずつのデータを測定した。ノード探索タスクのターゲットノードはランダムに指定された。実験の際には、ノード探索タスク達成時間(目標が指定されてから選択するまでに要した作業時間)とノード探索タスクの達成に必要なマウスボタンのクリック数を測定した。また、実験後に被験者に対してCone TreeとSpiral Treeのどちらを用いた方が検索が容易に感じたかを尋ねた。

[被験者] 被験者は本学の学生6名で、マウス操作に慣れており、階層構造の概念は理解していた。

[実験環境] 実験に用いたCone Tree, Spiral Treeは横置き型とした。これは、通常のCone Treeで全ノードのノード名を明記しようとするとノードの形が横長になるためノードの重なりが多くなり、ノード名が読みづらくなるからである。横置き型のCone TreeはCam Treeという別名で呼ばれる[5]。しかし、本論文においてはこれもCone Treeの名称で統一して呼ぶことにする。実験において、操作対象として視覚化

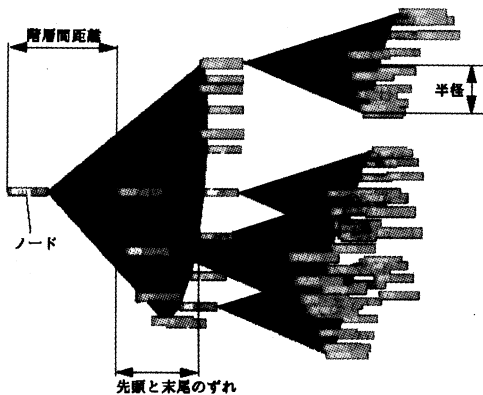


図4 評価実験に使用した Spiral Tree

Fig. 4 The Spiral Tree used in this experiment.

した階層構造情報は以下のような構成であった(図4参照)。

- 1層目(ルートノードの子): ノード数 16
- 2層目(ルートノードの孫): ノード数 16の階層が四つ

この階層構造は以下のパラメータで視覚化された。

- ノード: 縦 0.6 cm × 横 3.6 cm
- 階層間距離  $l$ : 9.0 cm
- 先頭ノードと末尾ノードのずれ  $d$  (Spiral Tree のみ): 4.5 cm
- 半径 1階層目: 8.0 cm, 2階層目: 3.1 cm
- 回転時のノードの移動速度: 1 cm/秒

2.2.2で述べたように、すべての階層構造をわかりやすく視覚化する視覚化パラメータの最適値を決定するのは困難である。そこで上記のパラメータは次の手順で手で決定した。まず、実験の準備段階で、ノード内の文字が読める程度にノードの大きさを決定した。次に、先頭と末尾が十分に明確になるように、階層内の先頭ノードの右端の  $x$  座標よりも末尾ノードの左端の  $x$  座標が大きくなるようにずれの距離  $d$  を決定した。最後に、階層構造全体が画面内に収まるように半径と階層間距離  $l$  を決定した。これらのパラメータでノード探索タスクが十分行えることを予備実験により確認した上で本実験を行った。

実験にはグラフィック WS (SGI社 Onyx RE2), WS に標準装備のマウス, 21 インチディスプレイを使用した。

### 3.1.1 実験1の結果

被験者 6 名のノード探索タスク達成時間を Cone

表1 Spiral Tree と Cone Tree の比較

Table 1 Comparison between Spiral Trees and Cone Trees.

	Spiral Tree	Cone Tree
平均タスク達成時間(秒)	5.68	6.57
分散	1.49	2.92
分散分析 ( $\alpha = 0.05$ )	有意差あり ( $P \ll 0.01$ )	
平均クリック数	1.34	1.38
分散	0.043	0.064
分散分析 ( $\alpha = 0.05$ )	有意差なし ( $P = 0.24$ )	

Tree, Spiral Tree を使用した場合のそれぞれについて平均値を求めて比較した。結果を表1にまとめる。Spiral Tree を使用した場合の平均タスク達成時間は Cone Tree を使用した場合に比べて1秒近く短縮されている。これらのノード探索タスク達成時間の差に統計的な有意差が認められた。有意差の検定には分散分析 ( $\alpha = 0.05$ ) を用いた。平均クリック数についても、ノード探索タスク達成時間と同様の分析を試みたが有意な差は見られなかった。

表1より、この実験において Spiral Tree を利用した場合、Cone Tree を利用した場合よりも、ノード探索タスク達成時間が減少することが統計的に認められることがわかる。この結果は、Spiral Tree による視覚化は Cone Tree による視覚化よりもノード探索時間の短縮(作業効率の向上)に有効であることを示している。

また、被験者の主観として「Spiral Tree と Cone Tree のどちらを用いた方が検索が容易に感じたか」の問に対しては全被験者が Spiral tree の方が容易であると答えている。

以上をまとめると、Spiral Tree は Cone Tree よりも、

- (1) ノード探索作業時間の短縮に有効、
- (2) ノード探索作業の操作感の向上に有効、

であると考えられる。

### 3.1.2 実験1の考察

被験者の主観から、Cone Tree よりも Spiral Tree の方がノードの探索における操作感の向上に有効であったと考えられる。しかしながら Spiral Tree は Cone Tree に比べて階層としてのまとまりが認識しにくいと報告した被験者もいた。この意見は Spiral Tree の欠点を指摘している可能性がある。

また、この実験において使用された実験用の Spiral Tree と Cone Tree の実装上の不満として、以下の3点が挙げられた。

- 階層を自由な角度から眺めることができない,
- 回転させたくない階層 (親の階層) も回転する,
- 文字が読みづらい,

これらの不満点は, 実験用 Spiral Tree, Cone Tree に共通であるため, 実験結果に及ぼす影響は少ないと考えられる。

上で述べた Spiral Tree の特徴はノードの配置方針の違いから生じるものであるが, 順序の利用が可能になったことによるものかはわからない。これは Spiral Tree の配置方針では順序の利用を容易にする効果と, 注目階層の一覧性を向上させる効果の両方が期待されているからである。このため順序の利用に関する, より詳細な検討を行う必要がある。

以上の考察に基づき, 実装上の不満点を解消した実験用プログラムを用いて, 順序の利用に関する評価実験を試みた。実験用プログラムに施した改善は以下の3点である。

- 階層の回転角度を自由に操作できるようにするために, ノードをマウスにより回転軸に垂直な方向にドラッグすることで, ドラッグされた量に応じて階層を回転させる。
- 操作の対象としている階層のみを回転させることにし, 親の階層以上は回転させない。
- 文字はフォントを読みやすいものに変更する。

### 3.2 実験2: 階層内順序の利用に関する実験

この実験の目的は Spiral Tree で階層内の順序が利用されていることを確認することである。これは, 実験1から考えられる Spiral Tree の利点は

- ノードの重なるの減少による一覧性の向上
- 順序の利用

の両方の特徴に影響を受けていることを考慮し, ブラウジングの手掛りとしての順序が Spiral Tree で利用できるかを確認する必要があると考えたからである。

[実験環境] 以下に示す階層構造 A, 階層構造 B を, 順序なし Spiral Tree (階層内の順序をランダムにした Spiral Tree) と通常の Spiral Tree で視覚化した。2種類の階層構造データを用いた理由は, 視覚化の対象となる階層構造に実験結果が大きく依存しないことを確認するためである。

- 階層構造 A
  - 1層目 (ルートノードの子): ノード数 30
  - 2層目 (ルートノードの孫): ノード数 10 の階層が三つ
- 階層構造 B

- 1層目 (ルートノードの子): ノード数 30
- 2層目 (ルートノードの孫): ノード数 10 の階層が九つ

[被験者] 被験者は本学の学生 5 名であった。なお, 本実験の被験者は実験 1 には参加していない。

[実験手順] 階層構造 A, 階層構造 B を順序なし Spiral Tree と通常の Spiral Tree で視覚化した場合の 4 通りそれぞれについて, ノード探索タスク 10 試行を被験者に行わせた。この 10 試行を 1 セッションとし, 1 回の実験で 10 セッションを行った。実験は練習 2 回 (20 セッション), 本試行 3 回 (30 セッション) 行った。各回の間隔は半日から 1 日であった。記録したデータはタスク達成時間 (ノード探索タスク達成時間), ドラッグ数 (タスクの達成までにドラッグを使用した回数), クリック数 (タスクの達成までにクリックによる自動アニメーションを使用した回数) である。

#### 3.2.1 実験2の結果

階層内に与えられている順序を有効に利用できているかを調べるために, 順序が使用できる場合とできない場合についてのタスク達成時間, ドラッグ数, クリック数について結果を比較する。

まず, それぞれの階層が順序どおりに並べられている Spiral Tree を用いた場合と, 順序どおりに並べられていない Spiral Tree を用いた場合での, ノード探索タスクのタスク達成時間, ドラッグ数, クリック数のそれぞれについて, 統計的な有意差を検定した。統計的な有意差の検定には分散分析 ( $\alpha = 0.05$ ) を用いた。この際, タスクの難易度が目標ノードの存在する階層の深さに影響を受けることを考慮して, 分析に使用するデータは目標ノードが 2 階層目に存在する場合に限った。この結果, 階層構造 B を操作対象とするときのクリック数を除いて, 統計的な有意差は認められなかった (表 2~表 4)。

そこで, それぞれの場合の平均値に注目する。表 2~

表 2 順序の有無による Spiral Tree でのタスク達成時間の違い

Table 2 Result of experiment 2: node searching time.

階層構造 A に対する順序の有無	順序なし	順序あり
平均タスク達成時間 (秒)	7.30	6.44
(分散)	(10.98)	(10.79)
有意差	なし ( $P = 0.08$ )	
階層構造 B に対する順序の有無	順序なし	順序あり
平均タスク達成時間 (秒)	13.25	12.63
(分散)	(44.52)	(43.14)
有意差	なし ( $P = 0.49$ )	

表3 順序の有無による Spiral Tree でのドラッグ数の違い  
Table 3 Result of experiment 2: number of drags.

階層構造 A に対する順序の有無	順序なし	順序あり
平均ドラッグ数 (回)	0.49	0.32
(分散)	(0.49)	(0.32)
有意差	なし ( $P = 0.09$ )	
階層構造 B に対する順序の有無	順序なし	順序あり
平均ドラッグ数 (回)	1.14	1.04
(分散)	(1.53)	(1.36)
有意差	なし ( $P = 0.52$ )	

表4 順序の有無による Spiral Tree でのクリック数の違い  
Table 4 Result of experiment 2: number of clicks.

階層構造 A に対する順序の有無	順序なし	順序あり
平均クリック数 (回)	0.15	0.18
(分散)	(0.34)	(0.31)
有意差	なし ( $P = 0.69$ )	
階層構造 B に対する順序の有無	順序なし	順序あり
平均クリック数 (回)	0.25	0.04
(分散)	(0.63)	(0.04)
有意差	あり ( $P = 0.01$ )	

表4のそれぞれについて、順序が利用できる場合と利用できない場合の平均値を比較すると、階層構造 A を対象とするときのクリック数を除いて、タスク達成時間、ドラッグ数、クリック数のすべてについて階層内順序を利用できる場合の方が平均値が減少している。

この結果は、Spiral Tree を用いてのノード探索タスクでは階層内順序を利用できる場合の方が、タスク達成時間やタスク達成に必要な作業数を減少する傾向を示している。このことは、Spiral Tree では階層内の順序が利用されている可能性が高いことを示す。

### 3.2.2 実験2の考察

実験2から新たに提供したドラッグによる階層の回転が多く用いられており、クリックはほとんど用いられていない。このことから、ドラッグによる階層の回転は被験者に好んで使用されたことがわかる。しかしながら、ドラッグによる回転は被験者が行う操作の自由度を上げるため、結果として、タスク達成時間の分散を大きくしたと考えられる。このことが、順序が利用できる場合と利用できない場合でのタスク達成時間に関する統計的な有意差が認められなかった原因の一つであると考えられる。

実験2では2種類の階層構造を用いた。これは、結果が視覚化の対象となる階層構造に依存しないことを確認するためであった。この2種類の階層構造に関して、タスク達成時間やドラッグ数については構造に依存せず、順序がある方がタスク達成時間などが減少す

る傾向にあることを観察することができた。しかし、この2種類の検討だけでは不十分であり、より複雑な階層構造を用いた場合についても検討を重ねる必要がある。しかしながら、すべての種類の階層構造に関して検討することは不可能であり、視覚化対象となる階層構造の複雑さがノード探索タスクに与える影響を厳密に調べるのは困難である。

## 4. 考 察

本章では、Spiral Tree を利用するとき決定が難しいと考えられる視覚化パラメータの設定について考察する。

前章での二つの実験では、Spiral Tree におけるずれの距離や階層間距離、各階層の半径といった視覚化パラメータをあらかじめ調整した上で実験を行った。しかし、これらの視覚化パラメータが不適切に設定されればかえって見づらくなる可能性もある。例えば、階層間距離に対してずれの距離  $d$  が極端に小さいと先頭と末尾の位置が明確になりにくく、大きすぎると Spiral Tree 全体が画面内に収まらない上に階層としてのまとまりが見づらくなることが考えられる。また、半径の大きさも階層としてのまとまりやノードの重なりなどに影響を与えるパラメータであると考えられる。実験1において被験者が「階層としてのまとまりが認識しづらい」と報告したことや、実験2において階層内順序の有無に関する統計的な有意差が認められなかったことなどは、これらのパラメータが最適な値に設定されていなかった可能性を示している。

Spiral Tree では Cone Tree とは違い、階層の深さの違うノード同士の重なりが起こる可能性がある。例えば、階層間距離に対してずれの距離が大きく設定された場合など、視覚化パラメータが適切に設定されていない場合には、ある階層内の先頭付近のノードの子孫と末尾付近のノードが重なって見えることが考えられる。また、視覚化パラメータが適切に設定されていても、深い階層をもつ階層構造を視覚化した場合、祖先として各階層の先頭ノードばかりをもつ階層と末尾ノードばかりをもつ階層では、ずれの距離の蓄積によって、違う深さの階層が重なってしまう。このことは、実験で用いたような小規模な階層構造ではあまり問題にならないが、階層が大規模な場合には画面を繁雑にする可能性がある。しかし、大規模な階層構造の視覚化に関しては、Cone Tree においても階層数においては10階層程度、ノード数においては1000ノード

程度が限界であることが指摘されている。これより大規模な階層構造への対応は今後の課題として残されるが、Cone Tree で扱える規模の階層構造においては階層の深さが違うノード同士の重なりはそれほど問題にはならないと考えられる。

## 5. むすび

本論文では大規模な情報を利用するための順序付き階層構造情報の表示手法として、従来の Cone Tree の改良型である Spiral Tree を提案した。

提案手法 Spiral Tree が順序を利用する検索に有効であることを確認するために、ノード探索タスクを用いた比較実験を行った。実験の結果、Spiral Tree を用いた場合の方が Cone Tree を用いる場合に比べてタスク達成時間や、タスク達成に必要な作業数を減少することがわかった。これは Spiral Tree が Cone Tree に比べて注目階層の一覧性が高く、階層内の順序を利用することができるという特徴によると思われる。

また、Spiral Tree で実際に階層内順序が有効に利用されているかを確認するための実験も行った。階層内の順序を利用できる Spiral Tree と利用できない Spiral Tree を用いてのノード探索タスクを用いた比較実験の結果、順序が利用できる場合の方がタスク達成時間などが減少する傾向が観察され、Spiral Tree では順序が利用されている可能性が高いことを示した。

今回検討した Cone Tree 型の視覚化ではノード数が1000程度より大規模な階層構造に適用した場合には画面にノードが密集してマウスで操作しづらくなるという問題が指摘されている。今後の課題としては、この問題について、提示ノード数を増加させるための従来研究[8]との統合、仮想物体提示領域の拡大による密集の回避や、ユーザの使用状況に応じて変化するノードの配置方針など[7]について検討することが挙げられる。

## 文 献

- [1] C. Ahlberg and B. Shneiderman, "Visual information seeking: Tight coupling of dynamic query filters with starfield displays," Proc. CHI '94: Human Factors in Computing Systems ACM, pp.313-317, 1994.
- [2] E. Rennison, "Galaxy of news an approach to visualizing and understanding expansive news," Proc. of the ACM Symposium on User Interface Software and Technology (UIST '94) ACM, pp.3-12, 1994.
- [3] S.K. Card, G.G. Robertson, and J.D. Mackinlay, "The information visualizer: An information workspace," Proc. CHI '91: Human Factors in Computing Systems,

pp.181-188, 1991.

- [4] J.D. Mackinlay, G.G. Robertson, and S.K. Card, "The perspective wall: Detail and context smoothly integrated," Proc. CHI '91: Human Factors in Computing Systems, pp.173-179, 1991.
- [5] G.G. Robertson, J.D. Mackinlay, and S.K. Card, "Cone Trees: Animated Visualizations of Hierarchical Information," Proc. CHI '91: Human Factors in Computing Systems, pp.189-194, 1991.
- [6] J.D. Mackinlay, R. Rao, and S.K. Card, "An organic user interface for searching citation links," Proc. CHI '95: Human Factors in Computing Systems, pp.67-73, 1995.
- [7] 寺岡照彦, 丸山 稔, "ユーザの視点に基づく適応型3次元インタフェース," 信学技報, MVE96-52, 1996.
- [8] J. Carriere and R. Kazman, "Interacting with huge hierarchies: Beyond cone trees," Proc. IEEE Sympo. on Information Visualization, pp.74-81, 1995.
- [9] T. Okuma, H. Takemura, Y. Katayama, H. Iwasa, and N. Yokoya, "3d visualization of hierarchical ordered information," Advances in Applied Ergonomics (Proc. ICAE '96), pp.668-671, 1996.
- [10] 大隈隆史, 竹村治雄, 片山喜章, 岩佐英彦, 横矢直和, "順序付き階層構造の三次元視覚化の一手法," 情処研報, pp.HI63-3, 1995.

(平成9年4月24日受付, 9月8日再受付)



大隈 隆史 (学生員)

平6阪大・基礎工・情報中退。平8奈良先端科学技術大学院大学博士前期課程了。現在、同大学院大学博士後期課程在学中。3次元ユーザインタフェース、仮想現実、情報視覚化の研究に従事。情報処理学会、日本VR学会各会員。



竹村 治雄 (正員)

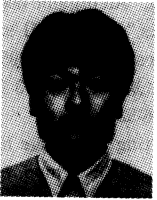
昭57阪大・基礎工・情報卒。昭62同大大学院博士後期課程単位取得退学。同年(株)ATR入社。3次元ユーザインタフェース、CSCW、仮想現実の研究に従事。平6奈良先端科学技術大学院大学・情報科学研究科助教授。工博。情報処理学会、IEEE、ACM、日本VR学会各会員。





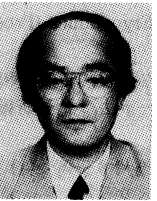
岩佐 英彦 (正員)

平2 阪大・工・通信卒。平6 同大大学院博士後期課程中退。同年奈良先端科学技術大学院大学・情報科学センター助手。現在 同情報科学研究科助手。人工知能、機械学習の研究に従事。情報処理学会、人工知能学会各会員。



片山 喜章 (正員)

平2 阪大・基礎工・情報卒。平6 同大大学院博士課程中退。同年奈良先端科学技術大学院大学・情報科学研究科助手。平7 同大情報科学センター助手。分散アルゴリズム、HCIなどに関する研究に従事。情報処理学会会員。



横矢 直和 (正員)

昭49 阪大・基礎工・情報卒。昭54 同大大学院博士課程了。同年電子技術総合研究所入所。以来、画像処理ソフトウェア、画像データベース、コンピュータビジョンの研究に従事。昭61~62 マツギル大・知能機械研究センター客員教授。平4 奈良先端科学技術大学院大学・情報科学センター教授。現在、同大情報科学研究科教授。平1 情報処理学会論文賞受賞。工博。情報処理学会、人工知能学会、日本認知科学会、映像情報メディア学会、IEEE 各会員。