

NAIST-IS-MT9451017

修士論文

Spiral Tree : 順序付き階層構造の三次元視覚化手法

大隈 隆史

1996年2月16日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

大隈 隆史

指導教官： 横矢 直和 教授
千原 國宏 教授
竹村 治雄 助教授

Spiral Tree : 順序付き階層構造の三次元視覚化手法*

大隈 隆史

内容梗概

計算機のユーザインタフェースに三次元のコンピュータグラフィックスを利用する三次元ユーザインタフェースは奥行きを表現することによって作業空間を拡大することができるという利点がある。階層構造情報の三次元視覚化手法 Cone Tree は三次元ユーザインタフェースを利用し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。本論文では, Cone Tree の問題点として階層内の順序を効果的に利用できないという点を指摘し, この問題点を解決するために, 各階層を円周上にはなく螺旋上に配置する順序付き階層構造情報の三次元視覚化手法 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ない, 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。さらに, より奥行き感を与えることのできる奥行き提示手法のうち, 運動立体視と両眼立体視に注目し, Cone Tree や Spiral Tree を用いて階層構造情報をブラウジングする場合にこれらの立体視を利用できるようにすることでユーザの作業にどのような影響を与えるかを調べる実験を行なった。その結果から, 階層構造のブラウジング時の運動立体視, 両眼立体視の得失について論じる。また, Spiral Tree の実用性を評価するために図書情報のブラウジング支援に Spiral Tree を適用したので報告する。

キーワード

三次元視覚化, ユーザインタフェース, 階層構造, 情報検索

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9451017, 1996年2月16日.

Spiral Tree : A 3D Visualization of Hierarchical Ordered Information*

Takashi Okuma

Abstract

Recently 3D user interface has attracted much attention for visualizing structured information. However, existing techniques cannot visualize hierarchical ordered information effectively. In this paper, the Spiral Trees are proposed for 3D visualization of hierarchical ordered information. Spiral Trees place their nodes along a spiral and show the head and tail of ordered nodes explicitly. In order to evaluate the effectiveness of the method, an experiment with target node searching tasks was conducted. The experiment has consequently shown that the Spiral Trees reduce the time required for target searching tasks. In order to investigate characteristics of a head coupled display system and binocular stereopsis that enhance depth cues, some experiments were conducted. The results imply some characteristics of the head coupled display system and binocular stereopsis. To examine the usefulness of Spiral Trees in practical applications, we have attempted to apply Spiral Trees to user-interface for digital library

Keywords:

3D visualization, user interface, hierarchical structure, information searching

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9451017, February 16, 1996.

目次

1. はじめに	1
2. 階層構造の三次元視覚化手法 Cone Tree	4
2.1 Cone Tree の形状と操作	4
2.2 Cone Tree の利点	5
2.2.1 ノード探索タスク	7
2.2.2 二次元視覚化と Cone Tree との比較実験の結果	7
2.3 Cone Tree の問題点	7
3. 順序付き階層構造の三次元視覚化手法 Spiral Tree	9
3.1 Spiral Tree の提案	9
3.2 Cone Tree と Spiral Tree の比較実験	10
3.2.1 実験1 :Cone Tree と Spiral Tree の比較実験	10
3.2.2 実験2 :順序の有無, 階層の複雑さに関する比較実験	14
3.3 Spiral Tree に関するまとめ	20
4. 三次元視覚化における運動立体視と両眼立体視の利用	21
4.1 Fish Tank Virtual Reality	21
4.2 Cone Tree における奥行き提示手法の比較	22
4.2.1 実験3 :運動立体視と両眼立体視の比較実験	22
4.2.2 実験4 :運動立体視と両眼立体視に関する追加実験	25
4.2.3 実験5 :構造の複雑さの違いに関する実験	27
4.3 Spiral Tree における運動立体視と両眼立体視	34
4.3.1 実験6 : ノード探索タスクによる実験	34
4.3.2 実験7 : 相違点指摘タスクによる実験	36
4.4 運動立体視と両眼立体視に関するまとめ	40
5. 図書検索への応用	42
6. むすび	44

謝辭	46
參考文獻	47

目 次

1	Cone Tree	4
2	ノード選択時の回転アニメーション	5
3	Spiral Tree の一階層	10
4	評価実験に使用した Spiral Tree	11
5	実験の様子	23
6	相違点指摘タスク中の画面	38
7	図書検索用 Spiral Tree	42
8	図 7 の部分拡大図	42

表 目 次

1	Spiral Tree と Cone Tree の比較	12
2	順序の有無による Spiral Tree でのタスク達成時間の違い	16
3	順序の有無による Spiral Tree でのドラッグ数の違い	16
4	順序の有無による Spiral Tree でのクリック数の違い	17
5	Spiral Tree と Cone Tree でのタスク達成時間の違い	18
6	Spiral Tree と Cone Tree でのドラッグ数の違い	18
7	Spiral Tree と Cone Tree でのクリック数の違い	19
8	ノード探索タスクにおけるドラッグとクリックの使用回数	20
9	Cone Tree における運動立体視と両眼立体視の比較	24
10	運動立体視と両眼立体視に関する追加実験の結果	26
11	複雑さの異なる階層構造におけるタスク達成時間の比較	28
12	複雑さの異なる階層構造におけるドラッグ数の比較	29
13	複雑さの異なる階層構造におけるクリック数の比較	30
14	エラー率 (単位 %)	32
15	Cone Tree での各奥行き提示方法の平均順位	32
16	Spiral Tree と Cone Tree における運動立体視使用の比較実験の結果	35
17	サブタスク 1 の平均終了時間 (単位 :秒)	39
18	サブタスク 2 の平均終了時間 (単位 :秒)	39

1. はじめに

個人や企業が扱う情報は増加する一方であり、大規模情報の管理や利用を支援するツールへの要求が高まってきている。ユーザが情報を利用する場合、情報の検索だけを目的とすることは少なく、通常は他の大きな作業、例えば論文の執筆や報告書の作成などを目的とし、これを達成する手助けとして必要な情報を大規模な情報群から取り出して利用する。このような場合、「出版社 A の書籍 B」というようにいつも具体的な目標がある訳ではなく、「統計処理の入門書」というように漠然としか目標が定まっていない場合や、あるいはただ情報群を眺めていて自分の目的作業に関連がありそうな情報があれば取り出そうと考えている場合など、ユーザ自身が必要な情報が何であるのかを明確に表現できない場合が多い。このような場合には、情報全体を通して眺めることで引き出す情報を明確にするきっかけを得ることができる。このように明確な目標を持たずに全体を通して眺める作業をブラウジングと呼ぶ。文献 [1] で水野らは、情報の検索時においては、従来からのデータベースなどによる情報のフィルタリングと、情報を様々な視点で鳥瞰するブラウジングが等しく重要であるとしている。大規模情報の利用支援の立場からユーザインタフェースに関する研究も、動的な問い合わせのフィルタをデータベースのインタフェースに適用している FilmFinder[2]、魚眼視野を与えることで大規模な表を効果的に視覚化する Table Lens[3] など、盛んに行われている。これらのことから、情報を利用するにあたってブラウジングを支援するユーザインタフェースは重要な役割を果たすと考えられる。

ブラウジングの対象となる情報群が階層構造として管理されていれば、ユーザは必要な情報が何であるのかを明確に意識していない場合でも、自分の興味に近そうなグループからブラウジングしていくことで、徐々に具体的な情報へと興味を絞り込んで行き、引き出す情報を具体化していくことができる。このように大規模な情報を属性や内容によって分類し階層的に管理することで、ユーザが情報を利用する際に必要な情報がどこにあるのかが分かりやすくなり、ブラウジングの負担を軽減することができると考えられる。このため、図書館の書籍の分類や UNIX のファイルシステムなど、大規模な情報は階層的に管理されることが多く、多くのアプリケーションで階層構造情報が利用されると考えられる。このことか

ら、階層構造情報のブラウジング支援は重要で応用範囲も広いと考えられる。

階層構造情報のブラウジング時には多くの情報が同時に見やすい形で提示されている方が望ましい。情報が持つ階層構造全体を視覚化することで、個々の情報を見るだけでは分からない情報、例えば情報同士の関連やユーザの作業に関連する情報の分布の状況などが分かるようになる。ところが二次元的に情報を配置しても、限られた画面内ではすぐに限界に達する。このため、奥行き表現を利用することで作業空間を拡大できる 3DUI(三次元ユーザインタフェース) に注目が集まってきている。ここで、3DUI とは実時間三次元コンピュータグラフィックスを計算機のユーザインタフェースに適用したものをいう。3DUI を用いたシステムの例として、変電設備の保守作業を視覚化したシステム [4] や抽象的な構造を持つ情報を視覚化するシステムである Information Visualizer [5, 6] が知られている。Information Visualizer 中のアプリケーションとして特に情報のブラウジングを支援するものでは、図書目録の引用関係をブラウジングする Butterfly application [7] や線形構造をもつ情報の視覚化手法 Perspective Wall [8], 階層構造情報の三次元視覚化手法 Cone Tree[9] などが知られている。

以上の考察から本論文では、階層構造として管理された情報群のブラウジングを 3DUI で支援するツールが重要であると考え、これを実現するためには、階層構造を三次元的に視覚化する必要がある。そこで本研究では、階層構造情報の三次元視覚化手法 Cone Tree に注目する。Cone Tree は 3DUI を用いることで多くの利点を持つ。しかし、三次元的に視覚化する際に各階層の情報を示すノードを円周上に並べるため、階層ごとに情報に与えられた順序が視覚化できないう問題点を持つ。この問題点を解決するために各階層を螺旋上に配置する三次元視覚化手法 Spiral Tree を提案する。提案手法の有効性を評価するために比較実験を行い、Spiral Tree が Cone Tree より短時間かつ少ない作業数で情報を探索できるという結果を得た。さらに、Cone Tree や Spiral Tree を用いて三次元的に視覚化された階層構造情報をユーザに見やすく提示するための奥行き提示手法、運動立体視と両眼立体視特について調べるために評価実験を行った。その結果、運動立体視についてはノードの発見を容易にするがマウスによる選択に影響を与えることが分かり、両眼立体視についてはノードの発見を容易にすることが

分かった。これらの実験の結果は 3DUI 一般に適用可能な基本的特性を示唆し得るものとして重要であると考えられる。

以下 2 章では, Cone Tree の形状上の特徴, 考えられる利点, 問題点について述べる。3 章では, Cone Tree の問題点を解決する三次元視覚化手法として Spiral Tree を提案し, Cone Tree と Spiral Tree との比較実験について述べ, Spiral Tree の利点を検証する。4 章では, 3DUI における奥行き提示手法として運動立体視 (透視投影表示の視点と利用者の視点を一致させることで与えられる運動視差による立体視) と両眼立体視に注目し, これらの奥行き提示手法がユーザの作業に与える影響を調べる実験について述べ, これらの特性を明らかにする。5 章では Spiral Tree の実用性を評価するために, 図書情報のブラウジング支援に Spiral Tree を適用した例について述べる。最後に本論文を総括し, 今後の課題について述べる。

2. 階層構造の三次元視覚化手法 Cone Tree

この章では、従来の研究として階層構造の三次元視覚化手法 Cone Tree について述べる。まず Robertson が提案している Cone Tree の形状と操作について説明し、次に、土本らによる評価実験の結果を踏まえて Cone Tree の特徴を明らかにする。

2.1 Cone Tree の形状と操作

この節では、まず Cone Tree の形状の定義について述べ、次にユーザが最も頻繁に Cone Tree に対して行う操作である、ノードの選択による回転について説明する。

Robertson らは、階層構造の三次元視覚化を行なうための一手法として Cone Tree を提案している [9]。Cone Tree が視覚化する階層構造は一般的な木構造である。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表し、情報の親子関係を半透明の円錐で視覚化し、親ノードを円錐の頂点、子ノードを円錐の底の円周に均等に配置する。この円錐を階層的に配置することで階層構造を三次元視覚化する (図 1)。

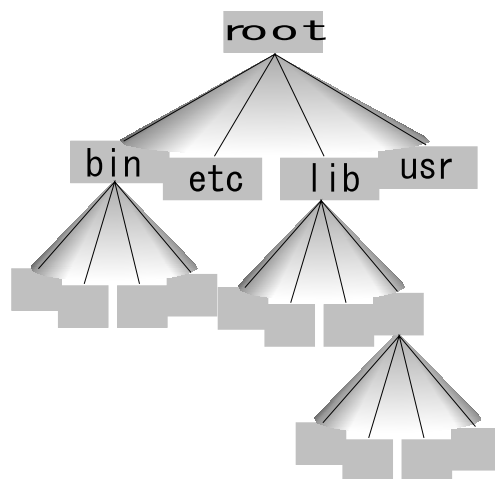


図 1 Cone Tree

Cone Tree においてノードがマウスカーソルでポインティングされると画面の全面にアニメーションしながら移動する。このとき、各階層の円錐が独立に回転し、選択されたノードを含むパス上の全てのノードは画面の手前側へ移動する(図 2)。

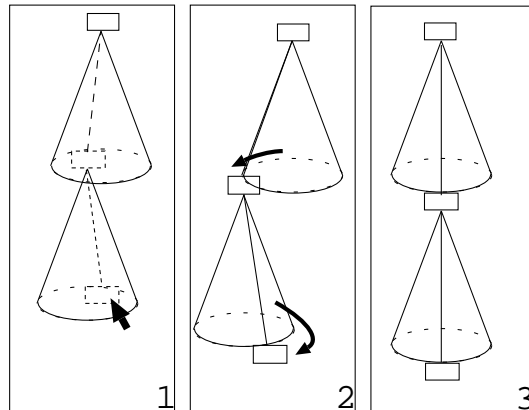


図 2 ノード選択時の回転アニメーション

2.2 Cone Tree の利点

この節では Cone Tree の利点について説明した後、Cone Tree の利点を検証するために土本らによって行われた比較実験について述べる。

Robertson らは、三次元グラフィックスとインタラクティブアニメーションの利用により、二次元の視覚化と比較して Cone Tree の方がより大規模な情報の取扱いが可能になると主張している。以下、三次元グラフィックスの利用の利点とインタラクティブアニメーションの利用の利点について順番に説明する。

Cone Tree における三次元グラフィックスの利用の利点として、以下の 3 点を挙げている。

1. 画面の効果的利用による全体構造の視覚化. 階層構造を二次元で表現する場合、階層構造全体の縦横比が問題になる。この場合にノード数が増えると、画面には部分しか表示できず、全体構造を把握するためにスクロールや

全体の縮小像が必要となる。Cone Tree は三次元の奥行きを利用してノードを円錐上に配置するので、全体像を保ちながら、より多くの情報を提示できる。

2. **魚眼視野 (fisheye view) 効果の提供.** Cone Tree では正面のノードは、他のノードに比べ近くて大きくみえる。これはユーザの注目しているもの程大きく見えるという魚眼視野効果を自然に実現する。
3. **奥行き感覚による知覚の手がかりの増加.** 三次元グラフィックスがもたらす透視投影表示や陰影はユーザの情報構造の知覚の手がかりとなる。

また、アニメーションの利用の利点については以下の 3 点を挙げている。

1. **構造の再認識に要する認知的負荷の軽減.** 画面のスクロールや切り替わりの際にユーザは注目している対象を見失うことがある。見失った対象を探して再認識することはユーザにとって負担となる。画面のスクロールや切り替わりの間をアニメーションで補間してユーザに提示することにより、ユーザは対象を見失うことが少なくなり、再認識による認知的負荷は大きく軽減される。
2. **アニメーションの観察による情報構造の理解.** ユーザはツリーの回転により、全体構造を様々な角度から観察できる。アニメーションの観察はユーザの情報構造の知覚を促進する。
3. **運動視差による奥行き知覚の促進.** アニメーションによる運動視差はユーザの三次元空間の知覚を促進する。

上記の Cone Tree の利点は三次元ユーザインタフェースの基本的特質の側面をとらえるものとして重要だと考えられるが、実験などによる具体的な検証は行なわれていなかった。土本らは実使用における単純な作業を想定したノード探索タスクによって、一般的な木構造の二次元視覚化と Cone Tree の比較実験を行なった [10]。以下、ノード探索タスクとこのタスクによる実験の結果について述べる。

2.2.1 ノード探索タスク

ノード探索タスクは以下に示される手順を1試行とするタスクである。

1. 被験者に各階層のノード名をつなげたフルパス名で目標ノードをランダムに指定する。
2. 被験者は各 Tree 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。

実験はこの試行を繰り返すことで進められる。ノード探索タスクは階層構造情報に対してユーザが頻繁に行なうと考えられる作業を想定している。ノード探索タスクは視覚化された階層構造情報の見やすさや操作性に影響をうけると考えられる。このため、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と考えられる。

2.2.2 二次元視覚化と Cone Tree との比較実験の結果

土本らによる、上述のノード探索タスクを用いての実験の結果、平均タスク達成時間 (目標が提示されてから選択されるまでの時間) は二次元視覚化の方が短かった。しかし、探索対象の情報が多い場合においては、大部分の被験者が Cone Tree の方が操作しやすいと感じたとを土本らは報告しており、Cone Tree が大規模情報の操作感の向上に有効であるとしている。この結果は被験者の主観と操作時間に食い違いを生じている。次節ではこの食い違いが生じた原因についての考察から Cone Tree の問題点を明らかにする。

2.3 Cone Tree の問題点

一般に、情報には順序づけ可能な属性を持つものが多い。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」などがこれにあたる。各階層内のノードの属性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時、この順序を手がかりに探索を行な

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ため, 各階層内の情報のある順序にしたがって整列しても先頭と末尾の位置が分かりづらく, ユーザは同一階層内のノードの順序を探索の手がかりとして十分に利用することができない。このためユーザは順序付き階層構造情報からの探索に不便を感じる。土本らの実験で Cone Tree でのノード探索時間が二次元の視覚化よりも遅かった理由の一つとして, 実験に使用したデータが順序付きの階層構造であり, 二次元視覚化では順序を探索の手がかりとして利用できたにも関わらず, Cone Tree ではうまく利用できなかったことが考えられる。

次章では順序付き階層構造のブラウジング時に 3DUI でも順序を利用できるようにするために提案した三次元視覚化手法 Spiral Tree について述べる。

3. 順序付き階層構造の三次元視覚化手法 Spiral Tree

この章では前章で述べた Cone Tree の問題点を改善した三次元視覚化手法 Spiral Tree について述べる。まず問題点の解決方法について説明し、この方法を適用した視覚化手法 Spiral Tree を提案する。次に Spiral Tree が Cone Tree よりも効率良くノードの探索が行なえるかを調べるために比較実験を行ない、この実験の結果から Spiral Tree の特徴について考察する。

3.1 Spiral Tree の提案

Cone Tree において情報探索過程で順序情報が利用しづらいという問題点は、各階層を円周上に配置するという配置方針に起因する。この問題を解決するためには、順序に従って整列された階層の先頭と末尾を明示する必要がある。これには、

1. 末尾と先頭の間切れ目を入れる、
2. 先頭ノードの色を変えて表示する、
3. 各階層のノードを螺旋上に並べる、

などの方法が考えられる。本手法では 3 の方法を採用した。3 を採用する理由は、1, 2 に比べて、

- 円錐の軸方向へのずれにより同一階層内のノー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、
- ノードの色を別の属性の視覚化に使用できる、

といった利点があるからである。この方法で階層の先頭と末尾を明示するように Cone Tree を改良した視覚化手法を Spiral Tree と呼ぶ [11, 12, 13](図 3)。Spiral Tree においても、三次元グラフィックスとインタラクティブアニメーションが Cone Tree と同様に用いられる。

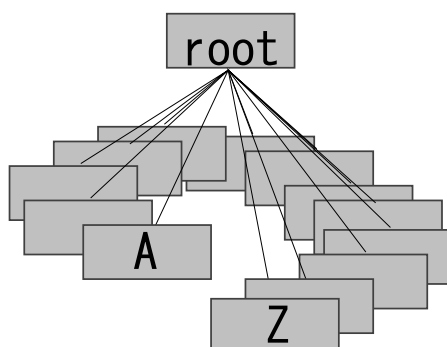


図 3 Spiral Tree の一階層

3.2 Cone Tree と Spiral Tree の比較実験

この節では提案手法 Spiral Tree の有効性を評価するために行った比較実験について述べる。まず Cone Tree と Spiral Tree との比較実験について述べ、結果から Spiral Tree の利点について考察する。次に, Spiral Tree の特徴をより詳細に調べるために, 順序の有無や階層の複雑さなどを変えて行った比較実験について述べる。最後に Spiral Tree の特徴をまとめる。

3.2.1 実験 1 :Cone Tree と Spiral Tree の比較実験

提案手法 Spiral Tree の有効性を確認するために, 実験用の Cone Tree, Spiral Tree を作成し, 比較実験を行なった。以下にその詳細を述べる。

実験 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 [9]。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

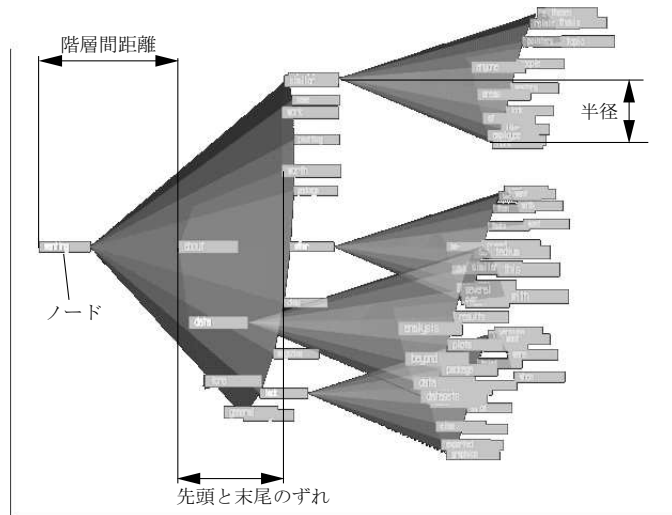


図 4 評価実験に使用した Spiral Tree

- 2 層目 (ルートノードの孫) : ノード数 16 の階層が 4 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 0.6cm × 横 3.6cm
- 階層間距離 : 9.0cm
- 先頭ノードと末尾ノードのずれ (Spiral Tree のみ) : 4.5cm
(注 : Spiral Tree のノード座標は Cone Tree のノード座標を, 円錐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.)
- 半径 一階層目 : 8.0cm , 二階層目 : 3.1cm
- 回転時のノードの移動速度 : 1cm / 秒

以上の実験用 Spiral Tree および Cone Tree を使用して階層構造情報を被験者に提示した. 実験用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持たないものとした. ノードには順序をつ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なう。

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とした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 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を尋ねた。

試行数は Spiral Tree, Cone Tree とともに一人 100 試行, 被験者は 6 名であった。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。

この実験から得られたデータについて、以降で検討する。

結果 被験者 6 名のノード探索タスク達成時間を Spiral Tree, Cone 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する。Spiral Tree を使用した場合の平均ノードタスク達成時間は Cone Tree を使用した場合の時間に比べて一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 ($\alpha = 0.01$) 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差は見られなかった。

以上の結果を表 1 にまとめる。

表 1 Spiral Tree と Cone Tree の比較

	Spiral Tree	Cone Tree
平均時間 (秒)	5.68	6.57
分散	1.49	2.92
分散分析 ($\alpha = 0.01$)	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 ($\alpha = 0.01$)	有意差なし ($P = 0.24$)	

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」の間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。

考察 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分かる。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に有効であることを示している。

また表1における平均タスク達成時間の分散に注目すると Spiral Tree 利用時の平均タスク達成時間の分散は Cone Tree 利用時に比べて半分近くの値を示している。これは Spiral Tree 利用時は Cone Tree 利用時に比べて安定した探索時間で目標を選択できていることを示す。

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。

以上の考察をまとめる。

- 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. 安定した探索時間で目標を選択可能,
 3. ノード探索作業の操作感の向上に有効,

と考えられる。

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、以下の三点が挙げられた。

- 階層を自由な角度から眺めることができない,
- 選択されたノードの属する階層だけを回転させて欲しい,
- 文字が読みづらい,

そこで、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、階層を回転を利用できるようにした。階層の回転は直接操作している回転だけとした。文字はフォントを読みやすいものに変更した。このようにして実装上の問題を解決し、Spiral Tree の特徴をより詳細に観察するための追加実験を行なった。

3.2.2 実験 2 : 順序の有無, 階層の複雑さに関する比較実験

目的 この実験の目的は以下の 3 つの疑問点を確認することであった。

- 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。実験 1 での Spiral Tree の利点はノードの重なりによる一覧性の向上のみに起因するものでないことを確認する。
- 構造が複雑になっても Spiral Tree はノード探索タスクに有効か。実験 1 に用いた階層構造は比較的単純なものであったにも関わらず階層としてのまとまりが認識しにくいという意見が挙げられているので、階層構造が複雑になればさらに見にくくなる可能性がある。
- ドラッグによる回転は使用されるか。

実験 以下に示す階層構造 A, 階層構造 B を、順序無し Spiral Tree(階層内の順序をランダムにした Spiral Tree) と通常の Spiral Tree, Cone Tree で視覚化し、それぞれについてノード探索タスク 10 試行を被験者に行なわせ、これを 1 セッションとした。1 回の実験で 10 セッションを行ない、これを 5 回 (計 50 セッション) 行なった。各回の間隔は半日から一日であった。被験者は 5 名であった。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使用した回数) である。実験の 5 回のうち前半 2 回は練習とし、後半 3 回分の実験データを分析に用いた。

- 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 3 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 9 つ

結果と考察 以上の実験で記録したデータでは, 分散値が大きくデータが不安定であった. これはタスクの難易度にばらつきがあるためだと考え, タスクの難易度を統一して分析するために, 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. 以降では実験結果を参照しながら前述した 3 つの疑問点について検討する.

順序の利用に関する結果と考察 ここでは, 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し, 最後に順序の利用に関して考察する.

まず, 順序無し *Spiral Tree* と通常の *Spiral Tree* のそれぞれを用いた場合の平均タスク達成時間に注目する. ここでは視覚化される階層構造の A,B に関わらず, この二つの場合のタスク達成時間に統計的な差は認められない. 統計的有意差の検定には分散分析 ($\alpha = 0.05$) を用いた. この結果を表 2 にまとめる. この結果からは *Spiral Tree* において順序の有無がタスク達成時間に影響を与えているとはいえない. しかしながら, 順序がある場合の方がタスク達成時間が短いという傾向が見られる.

表 2 順序の有無による Spiral Tree でのタスク達成時間の違い

階層構造 A	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	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 ($P = 0.49$)	

次に、ドラッグ数について検討する。ドラッグ数についても統計的な有意差は認められないが、全体的に、順序がある場合の方がタスクの達成に要した平均ドラッグ数が減少している傾向が見られる。この結果を表 3 にまとめる。

表 3 順序の有無による Spiral Tree でのドラッグ数の違い

階層構造 A	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	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 ($P = 0.52$)	

さらに、クリック数については、階層構造 B では順序がある場合と無い場合について統計的に有意差が認められ、順序がある場合の方が減少している。階層構造 A については統計的な有意差は認められない。この結果は表 4 にまとめる。

表 4 順序の有無による Spiral Tree でのクリック数の違い

階層構造 A	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	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 ($P = 0.01$)	

以上タスク達成時間, ドラッグ数, クリック数に関する結果をまとめると

- 順序がある方が平均タスク達成時間が減少している.
- 順序がある方が平均ドラッグ数が減少している.
- 順序がある方が平均クリック数が減少している. 特に, 階層構造 B においては, 統計的な有意差も認められる.

となる. これらの結果から, Spiral Tree では, 順序がある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ことができるといえる. このことは Spiral Tree を用いてのノード探索タスクにおいてある程度順序が利用できていることを示している.

階層構造の複雑さに関する結果と考察 次に, 階層構造が複雑になっても Spiral Tree はノード探索タスクに有効かを, 順序の有無を比較した場合と同様に, タスク達成時間, ドラッグ数, クリック数を比較することで検討する.

まず, Spiral Tree と Cone Tree の平均タスク達成時間について表 5 にまとめる. この結果から次の 2 点が言える.

- 比較的構造が単純な階層構造 A に対しては, 統計的な有意差は認められないものの, Spiral Tree を利用する場合の方がタスク達成時間が減少している.

- より複雑な構造を持つ階層構造 B に対しては, Spiral Tree を利用する場合の方がタスク達成時間が減少しており, 統計的な有意差も認められる.

表 5 Spiral Tree と Cone Tree でのタスク達成時間の違い

階層構造 A	Spiral Tree	Cone Tree
平均タスク達成時間 (秒)	6.44	7.46
分散	10.79	17.47
有意差	なし ($P = 0.07$)	
階層構造 B	Spiral Tree	Cone Tree
平均タスク達成時間 (秒)	12.63	14.94
分散	43.14	52.24
有意差	あり ($P = 0.02$)	

ドラッグ数については階層構造の A,B 共に統計的な有意差が認められ, Spiral Tree の方がドラッグ数が少なかった. この結果については表 6 にまとめる.

表 6 Spiral Tree と Cone Tree でのドラッグ数の違い

階層構造 A	Spiral Tree	Cone Tree
平均ドラッグ数 (回)	0.32	0.69
分散	0.32	0.56
有意差	あり ($P < 0.01$)	
階層構造 B	Spiral Tree	Cone Tree
平均ドラッグ数 (回)	1.04	1.41
分散	1.36	2.06
有意差	あり ($P = 0.03$)	

クリック数については, タスク達成時間と同様の傾向が見られた. すなわち, 以下の 2 点が言える.

- 比較的構造が単純な階層構造 A に対しては, 統計的な有意差は認められないものの, Spiral Tree を利用する場合の方がクリック数が減少している.

- より複雑な構造を持つ階層構造 B に対しては, Spiral Tree を利用する場合の方がクリック数が減少しており, 統計的な有意差も認められる.

詳細を表 7 にまとめる.

表 7 Spiral Tree と Cone Tree でのクリック数の違い

階層構造 A	Spiral Tree	Cone Tree
平均クリック数 (回)	0.18	0.25
分散	0.31	1.47
有意差	なし ($P = 0.61$)	
階層構造 B	Spiral Tree	Cone Tree
平均クリック数 (回)	0.04	0.19
分散	0.04	0.55
有意差	あり ($P = 0.05$)	

以上のタスク達成時間, ドラッグ数, クリック数に関する結果をまとめると,

- Spiral Tree を使用した方が平均タスク達成時間が減少している. 特に, 階層構造 B においては統計的な有意差も認められる.
- Spiral Tree を使用した方が平均ドラッグ数が減少している.
- Spiral Tree を使用した方が平均クリック数が減少している. 特に, 階層構造 B においては, 統計的な有意差も認められる.

となる. この結果から, 構造が比較的複雑な階層構造 B の方が Spiral Tree によるノード探索タスクへの利点が顕著に現れると考えられる.

ドラッグによる回転に関する結果と考察 最後に, ドラッグによる回転がどの程度使用されるのかを検討する. 表 8 において, 一試行当りの平均ドラッグ数と平均クリック数を比較するとドラッグ数の方が多い. 今回の実験中, 全ての被験者はドラッグを好んで使う傾向があることも観察された. 極端な被験者は全ての回転作業をドラッグで行なった. これらのことからノード探索タスクを行う際に,

自動的に手前に移動するアニメーションよりも自分で自由に回転角度を決めることができるドラッグを被験者が好むことが分かる。

表 8 ノード探索タスクにおけるドラッグとクリックの使用回数

		Spiral Tree (順序なし)	Spiral Tree (順序あり)	Cone Tree
階層構造 A	平均ドラッグ数	0.49	0.32	0.69
	平均クリック数	0.15	0.18	0.25
階層構造 B	平均ドラッグ数	1.14	1.04	1.41
	平均クリック数	0.25	0.04	0.19

実験 2 に関するまとめ 実験 2 に関する結果と考察をまとめる。

- 順序の有無による統計的な差はほとんど見られなかったが、順序がある方がタスク達成時間やタスク達成に必要な作業数(ドラッグ数やクリック数)が減少する傾向が見られる。
- 視覚化される階層が複雑な階層構造 B の方が Spiral Tree によるタスク達成時間やタスク達成に必要な作業数の減少が顕著に見られる。
- 自動アニメーションによる回転よりもドラッグによる回転の方が使用頻度が高い。

3.3 Spiral Tree に関するまとめ

本章では順序付き階層構造の三次元視覚化手法 Spiral Tree を提案し、2つの評価実験を行なった。その結果、順序付きの階層構造を視覚化する際に Spiral Tree を用いることで、順序の利用が可能になり、注目階層の一覧性が向上し、従来手法の Cone Tree より素早く目的の情報を見つけることが可能となることが分かった。また、複雑な階層構造において、Spiral Tree における利点が顕著に現れていることから、Spiral Tree は複雑な階層の三次元視覚化にも利用できることが分かった。

4. 三次元視覚化における運動立体視と両眼立体視の利用

前章までで、階層構造情報の三次元視覚化手法 Spiral Tree について述べた。このような三次元視覚化手法により視覚化された三次元物体をユーザに提示する際の奥行き提示手法としては透視投影表示のみを用いる例が多い。この章では三次元表示における奥行き提示手法として両眼立体視と運動立体視(視点追従による運動視差を利用した立体視)に着目し、ユーザが階層構造をブラウジングする時に両眼立体視や運動立体視がどのように利用されるかを実験により検証する。

4.1 Fish Tank Virtual Reality

Ware らは提示される透視投影表示の視点を利用者の視点に厳密に一致させて、実時間で更新する両眼立体視画像を Fish Tank Virtual Reality と名付け、ヘッドマウンテッドディスプレイを用いた没入型 Virtual Reality と比較して、以下の利点を挙げている [15].

- 高解像度の画面表示が可能.
- ユーザが凝視する作業範囲が限定されるので、ぼやけたような背景を置くことで奥行きの広がりを感じさせることができる.
- 眼球運動による視点のずれの影響を受けにくい.
- 現実世界との自然な統合.

さらに Ware らは重複する二本の木における任意の末端がどちらの木に属するかを判別するタスクによって、視点追従による運動立体視の効果、両眼立体視の効果、その組合せによる効果を調べた。結果として運動立体視と両眼立体視の組合せによる表示が最もエラー率を減少したと報告し、エラー率の減少や、被験者の主観から両眼立体視による効果よりも視点追従による運動立体視の効果の方が大きいと結論づけている。

Spiral Tree によって支援される階層構造情報のブラウジングは

1. マウスによるインタラクティブな操作,
2. 文字の認識と検索,

を伴う。Spiral Tree をより立体的に知覚させて構造の理解を促進するために、Fish Tank Virtual Reality の適用を試みた。文字情報を扱うので高解像度を特徴とする Fish Tank Virtual Reality は都合が良いと考えたからである。

しかしながら、Spiral Tree に対する Fish Tank Virtual Reality の適用においては、Ware らが示しているような視点追従による運動立体視の優位が保たれるとは限らない。これは視点の移動がインタラクティブな操作や文字の認識と相互作用を及ぼすことが考えられるからである。すなわち、ユーザは情報を素早く見つけることができても、それを操作するためのポインティングに手間取ってしまう可能性がある。

次節以降では実験用の Cone Tree, Spiral Tree へ Fish Tank Virtual Reality の適用を試み、実験により運動立体視や両眼立体視がノード探索タスクにおいてどのように利用されるかを検討することでそれぞれの奥行き提示手法のノード探索タスクに対する有効性について論じる。

4.2 Cone Tree における奥行き提示手法の比較

この節では実験用 Cone Tree を用いて両眼立体視と運動立体視について調べるために行った実験について述べる。まず、Spiral Tree の提案以前に行った実験について述べる。次にその実験に対する追加実験について説明する。最後に、視覚化する階層構造情報の複雑さを変えての両眼立体視と運動立体視に関する実験について述べる。

4.2.1 実験 3 :運動立体視と両眼立体視の比較実験

実験 まず、実験用 Cone Tree に

1. 運動立体視モード
2. 両眼立体視モード

をそれぞれ独立に選択できる機能を追加した。これらの有無を組合せて4通りの奥行き提示方法での表現を可能にし、それぞれの奥行き提示手法がノード探索タスクに与える影響を調べるための実験を行なった。

運動立体視は多関節アーム型頭部位置入力装置を用いて頭部の位置を測定し、被験者の視点と透視投影表示の理論上の視点を一致させて運動視差を与えることで実現した。両眼立体視は液晶シャッター眼鏡により、理論上の視点位置を両眼間隔だけずらして計算した二枚の透視投影表示をそれぞれ対応する目に提示して両眼視差を与えることで実現した(図5)。



図5 実験の様子

実験の操作方法を統一するために、両眼立体視でも左右どちらかの利き目のみにマウスカーソルを提示し、画面上を平面的に移動するものとした。選択するこ

とができるノードは視点とマウスマウスカーソルを結ぶ直線上にあるノードで、選択できるノードの枠を強調表示した。このため片目にしかマウスマウスカーソルが表示されていなくても、カーソルが指示しているノードの確認は容易に行なえる。提示手法ごとに、練習試行 30 試行、本試行 100 試行を行なった。被験者は本学の学生 3 名であった。

結果と考察 以上の実験におけるタスク達成時間とクリック数の平均値を表 9 にまとめる。

表 9 Cone Tree における運動立体視と両眼立体視の比較

タスク達成時間	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	7.90 (15.45)	7.79 (14.96)	なし ($P = 0.84$)
運動立体視あり (分散)	6.66 (6.97)	6.12 (5.81)	なし ($P = 0.13$)
有意差	あり ($P = 0.01$)	あり ($P < 0.01$)	
クリック数	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	1.76 (0.43)	1.58 (0.38)	あり ($P = 0.05$)
運動立体視あり (分散)	1.18 (0.17)	1.13 (0.09)	なし ($P = 0.29$)
有意差	あり ($P < 0.01$)	あり ($P < 0.01$)	

まずタスク達成時間に注目する。両眼立体視の有無に関わらず、運動立体視がある場合の方が、ない場合より 1 秒以上短い時間でタスクが達成できている。一方、運動立体視がない場合、両眼立体視の有無による差はほとんどなく、運動立体視がある場合でもタスク達成時間の差は 0.5 秒程度であり、わずかであった。

クリック数についても運動立体視の有無によるクリック数の差に統計的な有意差が認められ、運動立体視が利用できる場合の方がクリック数が減少している。両

眼立体視の有無については運動立体視無しの場合にのみ統計的有意差が認められる。

以上より, 次のように考えられる。

- 運動立体視はタスク達成時間とクリック数の減少に有効。
- 両眼立体視はタスク達成時間とクリック数の減少にあまり効果がない。

と考えられる。

この実験において使用した実験用 Cone Tree では, ノードが移動する時のアニメーション時間を一定にするという設定であった。しかし, 後に土本らは, Cone Tree のアニメーションにはノードの移動速度を一定にする方がユーザにとって最も好ましいという結果を実験によって示した [16]。また, この実験に使用したマウスカーソルは被験者に不評で, 両眼立体視の効果を減少させている可能性がある。

そこで, 実験用 Cone Tree を

- ノードの移動速度を一定にする。
- マウスカーソルは両眼に立体的に提示する。

という 2 点について変更し, 追加実験を行なった。

4.2.2 実験 4 :運動立体視と両眼立体視に関する追加実験

実験 この実験では, 一人あたりの試行数を 30 回とし, 分析には後半 25 回分のデータを使用した。統計的な有意差の検定には分散分析 ($\alpha = 0.05$) を用いた。被験者は 16 名であった。

結果 運動立体視の有無がノード探索タスクに与える影響については, 運動立体視が使用できる場合は, 運動立体視が使用できない場合に比べて, タスク達成時間は増加し, クリック数は減少する。それぞれ統計的な有意差が認められる。

両眼立体視の有無がノード探索タスクに与える影響については, 両眼立体視が使用できる場合のタスク達成時間, クリック数は, 両眼立体視が使用できない場

合に比べて減少している。クリック数については統計的な有意差が認められるが、タスク達成時間については認められない。

この結果を表 10 にまとめる。

表 10 運動立体視と両眼立体視に関する追加実験の結果

タスク達成時間	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	6.85 (14.29)	6.74 (10.98)	なし ($P = 0.64$)
運動立体視あり (分散)	8.76 (30.44)	8.17 (21.28)	なし ($P = 0.10$)
有意差	あり ($P < 0.01$)	あり ($P < 0.01$)	
クリック数	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	2.56 (2.55)	2.22 (1.78)	あり ($P < 0.01$)
運動立体視あり (分散)	1.78 (1.68)	1.53 (0.96)	あり ($P < 0.01$)
有意差	あり ($P < 0.01$)	あり ($P < 0.01$)	

考察 クリック数が減少していることから運動立体視、両眼立体視ともにノードの発見を容易にすると考えられるが、運動立体視を伴う場合タスク達成時間が増加していることから操作性に影響を与えている可能性がある。

以上のことから、Cone Tree で視覚化された階層構造に対するノード探索タスクにおいては、以下のようにまとめられる。

運動立体視 クリック数、すなわちユーザの操作の手間を減少させる。ただし操作対象が動いてしまうため、ノードを操作するための時間が余分にかかってしまう。

両眼立体視 ノード探索タスクにおいてクリック数を減少させる。タスク達成時間には影響を与えない。

しかし、この結果は単純な階層構造についての実験結果から明らかになったことであり、より複雑な構造においては画面が混雑するため、両眼立体視の重要性が増すことが考えられる。そこで次に構造の複雑さを二段階に分けて実験を行なった。

4.2.3 実験 5 : 構造の複雑さの違いに関する実験

実験 この実験では視点追従による運動立体視と両眼立体視の組合せによる4通りを複雑さの異なる二つの階層構造についてノード探索タスクを行なった。今回の実験ではエラー率(間違えた目標を選択する回数/全体の試行回数)をとるために、ノードの操作のマウスボタンと目標選択用のマウスボタンを別のボタンに割り当てた。また、ドラッグも使用できる。実験に使用した階層構造は3.2.2節での実験で使用した階層構造A,Bである。被験者は5名で、各モードにつき10試行を行ない、これを1セッションとした。1回の実験で10セッションを行ない、これを5回(計50セッション)行なった。各回の間隔は半日から一日であった。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使用した回数)である。実験の5回のうち前半2回は練習とし、後半3回分の実験データを分析に用いた。また実験後、被験者に各モードの主観順位をたずね、運動立体視と両眼立体視の嗜好に関するアンケートを実施した。

運動立体視の有無に関する結果と考察 まず、運動立体視の有無がノード探索タスクにおけるタスク達成時間、クリック数、ドラッグ数にどのような影響を与えるかについて順に述べる。

運動立体視の有無について、タスク達成時間については、階層構造Bにおいて両眼立体視が使用できない場合にのみ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。それ以外の場合の有意差は認められない。傾向としては階層構造Aにおいては運動立体視が使用できる場合の方が時間が若干短い、階層構造Bにおいては運動立体視がある場合の方が時間が長くなっている。(表11参照)。

表 11 複雑さの異なる階層構造におけるタスク達成時間の比較

階層構造 A	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	6.73 (16.18)	6.54 (13.80)	なし ($P = 0.68$)
運動立体視あり (分散)	6.50 (12.35)	6.53 (15.18)	なし ($P = 0.94$)
有意差	なし ($P = 0.60$)	なし ($P = 0.97$)	
階層構造 B	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	13.32 (56.17)	13.17 (76.53)	なし ($P = 0.88$)
運動立体視あり (分散)	16.09 (118.03)	14.38 (105.52)	なし ($P = 0.19$)
有意差	あり ($P = 0.01$)	なし ($P = 0.30$)	

ドラッグ数については、階層構造 A で両眼立体視なしの場合にのみ、運動立体視の有無によるドラッグ数の差に統計的な有意差が認められた。階層構造 A においては運動立体視が使用できる場合の方が若干ドラッグ回数が少ない。階層構造 B においてはどちらとも言えない。(表 12 参照)。

表 12 複雑さの異なる階層構造におけるドラッグ数の比較

階層構造 A	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	0.55 (0.50)	0.40 (0.39)	なし ($P = 0.08$)
運動立体視あり (分散)	0.36 (0.39)	0.35 (0.44)	なし ($P = 0.90$)
有意差	あり ($P = 0.02$)	なし ($P = 0.50$)	
階層構造 B	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	1.18 (1.94)	0.93 (1.67)	なし ($P = 0.13$)
運動立体視あり (分散)	1.17 (3.08)	0.95 (2.12)	なし ($P = 0.28$)
有意差	なし ($P = 0.97$)	なし ($P = 0.88$)	

クリック数については、階層構造 A で両眼立体視ありの場合にのみ、運動立体視の有無によるクリック数の差に統計的な有意差が認められた。階層構造 A については運動立体視がある方がクリック数が減少する傾向があるが、階層構造 B についてはどちらとも言えない。(表 13 参照)。

表 13 複雑さの異なる階層構造におけるクリック数の比較

階層構造 A	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	0.20 (1.10)	0.27 (0.83)	なし ($P = 0.53$)
運動立体視あり (分散)	0.07 (0.11)	0.07 (0.09)	なし ($P = 0.98$)
有意差	なし ($P = 0.17$)	あり ($P = 0.01$)	
階層構造 B	両眼立体視なし	両眼立体視あり	有意差
運動立体視なし (分散)	0.16 (0.46)	0.34 (1.36)	なし ($P = 0.11$)
運動立体視あり (分散)	0.27 (0.89)	0.14 (0.42)	なし ($P = 0.19$)
有意差	なし ($P = 0.27$)	なし ($P = 0.08$)	

以上の結果をまとめる。

- 構造が複雑な時、運動立体視ありの場合の方がタスク達成時間が長い。
- 構造が簡単な時、運動立体視ありの場合の方がドラッグ数が少ない。
- 構造が簡単な時、運動立体視ありの場合の方がクリック数が少ない。

この結果より、実験 4 で明らかになった運動立体視の性質について、さらに以下のことが分かる。

- タスク達成に必要とする作業数を減少させる性質：構造が簡単な時に顕著

- タスク達成時間を増大させる性質 :構造が複雑な時に顕著

これは、構造が簡単であれば運動立体視の利点を活かして作業数を減少させることができるが、構造が複雑になり画面が混雑すると、

- 画面上の位置が動くノードのマウスでの選択がより困難になる、
- 隠れたノードを覗くすき間が少なくなる、

などのように、運動立体視の欠点が顕著に現れる状況が起こるためと考えられる。

両眼立体視の有無に関する結果と考察 次に両眼立体視の有無についても同様に述べる。タスク達成時間、クリック数、ドラッグ数の全ての場合において両眼立体視の有無による有意差は確認できなかった。階層構造 A におけるタスク達成時間は両眼立体視ありの場合の方が若干長い、階層構造 B におけるタスク達成時間は両眼立体視ありの場合の方が短い(表 11参照)。特に、運動立体視ありの場合について、両眼立体視の有無の差が顕著に現れている。ドラッグ数については階層構造 B の方が減少する傾向にあった(表 12参照)。クリック数についてはどちらも言えない(表 13参照)。

この結果から、構造が複雑になった時に両眼立体視の利点が顕著に現れることが分かる。これは両眼立体視が、両眼視差(奥行き距離によって決まる量)から奥行き感を得るものなので構造の複雑さに依存しないため、構造が複雑になって画面が混雑して見づらくなると、相対的に両眼立体視の利点が顕著になるためと考えられる。

エラー率 また、表 14に今回測定したエラー率についてまとめる。この表から運動立体視がエラー率の減少に効果があるという傾向が見られる。

表 14 エラー率 (単位 %)

階層構造 A		両眼立体視なし	両眼立体視あり
	運動立体視なし	4.7	6
	運動立体視あり	4	5.3
階層構造 B		両眼立体視なし	両眼立体視あり
	運動立体視なし	8.7	8.7
	運動立体視あり	8.7	7.9

主観評価の結果 次に被験者の主観による各奥行き提示方法の平均順位を表 15 に示す。主観順位は階層構造 A,B に関わらず両眼立体視、運動立体視ともに使用できる場合が最も評価が高かった。

表 15 Cone Tree での各奥行き提示方法の平均順位

階層構造 A		両眼立体視なし	両眼立体視あり
	運動立体視なし	3.4	2.6
	運動立体視あり	2.2	1.8
階層構造 B		両眼立体視なし	両眼立体視あり
	運動立体視なし	3.8	3.2
	運動立体視あり	2.6	1.2

被験者に対して行なったアンケートの結果は、

- 運動立体視単独の効果は
 - － 重要である :3 人
 - － 必要無し :1 人
 - － その他 :1 人 (ノード数が多い時は必要なし.)

- 両眼立体視単独の効果は
 - 重要である :4 人
 - 必要なし :1 人
- 両方の組合せを使用するか？
 - 使用する :4 人
 - 使用しない :1 人
- 運動立体視と両眼立体視どちらがより重要か
 - 運動立体視 :2 人
 - 両眼立体視 :2 人
 - その他 :1 人 (ノード探索タスクには運動立体視が有効だが, 見やすさに関しては両眼立体視の方が貢献している.)

となっている. このアンケートにより被験者の主観からは運動立体視と両眼立体視のどちらも重要であり, 特に両方を併用することが好まれることが分かる.

構造の複雑さに関する実験のまとめ 以上, 実験用 Cone Tree に運動立体視や両眼立体視を適用した時にノード探索タスクにこれらの立体視がどのように影響を及ぼすかを確認する実験を行ない, 検討した. 結果として以下のことが分かった.

運動立体視 構造が比較的簡単な時は, 操作数やエラー率の減少に効果があるが, 構造が複雑になるとタスクの達成時間を遅くする.

両眼立体視 運動立体視に比べてノード探索タスクに与える影響は小さいが, 探索時間や操作数を減少させる傾向にある. 構造の複雑さには影響を受けない.

また, 主観評価としては両方の立体視の併用が好まれることが分かった.

次節では Spiral Tree に対して視点追従による運動立体視と両眼立体視を適用した場合のノード探索タスクへの影響を調べる実験について述べる.

4.3 Spiral Tree における運動立体視と両眼立体視

4.3.1 実験 6 : ノード探索タスクによる実験

前述したように Spiral Tree の形状は Cone Tree より注目階層の一覧性が高いという特徴を持つ. この特徴により階層構造をブラウジングする時の運動立体視の使用量が減少する可能性がある. そこで Spiral Tree に運動立体視を適用し, Cone Tree に対して運動立体視のみを適用したものとの比較を行なった. この実験においては運動立体視の使用時には頭部の移動量を測定した. 統計的な有意差の検定には分散分析 ($\alpha = 0.05$) を用いた. 結果を表 16 にまとめる.

表 16 Spiral Tree と Cone Tree における運動立体視使用の比較実験の結果

階層構造 A		運動立体視あり	運動立体視なし	有意差
Spiral Tree	平均時間	6.05	5.72	なし
	(分散)	(8.76)	(12.26)	($P = 0.40$)
	平均クリック数	0.10	0.13	なし
	(分散)	(0.17)	(0.24)	($P = 0.60$)
	平均ドラッグ数	0.20	0.22	なし
	(分散)	(0.23)	(0.24)	($P = 0.81$)
	頭部移動量	12.41	—	
Cone Tree	平均時間	6.50	6.73	なし
	(分散)	(12.26)	(16.07)	($P = 0.60$)
	平均クリック数	0.07	0.20	なし
	(分散)	(0.11)	(1.10)	($P = 0.17$)
	平均ドラッグ数	0.36	0.55	あり
	(分散)	(0.38)	(0.50)	($P = 0.02$)
	頭部移動量	15.11	—	
階層構造 B		運動立体視あり	運動立体視なし	有意差
Spiral Tree	平均時間	14.23	11.80	あり
	(分散)	(50.85)	(44.19)	($P = 0.01$)
	平均クリック数	0.16	0.04	あり
	(分散)	(0.32)	(0.04)	($P = 0.03$)
	平均ドラッグ数	0.98	0.89	なし
	(分散)	(1.09)	(1.28)	($P = 0.52$)
	頭部移動量	19.95	—	
Cone Tree	平均時間	16.09	13.32	あり
	(分散)	(117.17)	(55.76)	($P = 0.01$)
	平均クリック数	0.27	0.16	なし
	(分散)	(0.88)	(0.46)	($P = 0.27$)
	平均ドラッグ数	1.17	1.18	なし
	(分散)	(3.06)	(1.93)	($P = 0.97$)
	頭部移動量	32.73	—	

階層構造 A に注目すると Spiral Tree を用いる場合は Cone Tree を用いる場合

と違い、運動立体視がある方がタスク達成時間がかかっている。また、階層構造 B に注目すると運動立体視がある場合の方がタスク達成時間、クリック数、ドラッグ数の全てが増加する傾向にある。また、頭部の移動距離は Cone Tree に比べて Spiral Tree の方が少ない。

運動立体視の効果に関する被験者による主観的判断は Cone Tree のほうが運動立体視の効果が大いだと答えた者が 3 名、差がないと答えた者が 2 名であった。

これらの結果から Spiral Tree の方が階層の一覧性が高いため、頭部を動かして見る必要が少なかったため、対象が動いてクリックしづらいという欠点がより顕著に現れたと考えられる。

4.3.2 実験 7：相違点指摘タスクによる実験

Spiral Tree と Cone Tree における運動立体視と両眼立体視の使われ方を見るために、階層構造のブラウジングを別の角度から捉えたタスクを用いて被験者 1 名を使用して実験を行なった。以下この実験について述べる。

ノード探索タスクは階層構造情報にたいしてユーザが頻繁に行なう作業を想定しており、抽象的ではあるが目標の相対的な位置情報をフルパス名と言う形で与えられる。ところが階層構造のブラウジング時には、目標の位置情報を全く持たない場合もある。このような場合には階層情報を局所的にではなく、全体的に捉える必要がある。階層構造全体を捉える必要のあるタスクとして相違点指摘タスクを設定し、このタスクを用いて運動立体視や両眼立体視がどのように利用されるかを検討した。

相違点指摘タスク 相違点指摘タスクは被験者に構造の違う二つの階層構造を提示して、その相違点を指示させるものである。二つの階層構造の相違点はある一つのノードの位置である。具体的には以下のようにして被験者に提示する二つの階層構造を作成する。

1. もととなる階層構造を用意する (Tree A).
2. Tree A から葉 (子の階層を持たないノード) L と L の親以外の節 (子の階層を持つノード) M を一つずつランダムに選出する。

3. L を M の子の階層へと移動する. 移動後の階層構造を Tree B とする.

試行中は, 画面上部に Tree A, 下部に Tree B を提示する (図 6). 被験者は Tree A, TreeB をブラウジングし, Tree A におけるノード L, Tree B におけるノード L を指示する. この指示作業を 1 試行とする. Tree B は試行毎に Tree A から生成される.

このタスクは, 以下の 2 点がノード探索タスクとは異なる.

- ブラウジングの開始時においてどこに目標があるかという情報が全く与えられない.
- 木の全体をブラウジングせずには, タスクを達成することができない.

被験者を観察した結果, このタスクを達成する方針として, 被験者はタスクをの 2 つのサブタスクに分けて実行することが分かった.

1. Tree A/B どちらかのノード L を両方の Tree を見比べながら見つける.
2. 先に見つけたノード L のノード名を手がかりにして他方の tree のノード L を見つける.

それぞれのサブタスクの手順は以下のとおり.

サブタスク 1 Tree A/B どちらかのノード L をブラウジングによって見つけ出す.

1. 上下の階層構造の各々の階層を同じ角度にそろえて眺める.
2. 階層の形状から違いを感じた部分に注目する.
3. 注目した階層内のノード名を順番に見比べて行き, 上下の階層構造で違うノードを発見する.

サブタスク 2 先に見つけたノード L のノード名を手がかりにしてに他方の Tree のノード L を探索する.

1. 形を眺め先頭と末尾の位置を確認する.
2. 先に見つけたノードの名前がありそうな場所に注目.

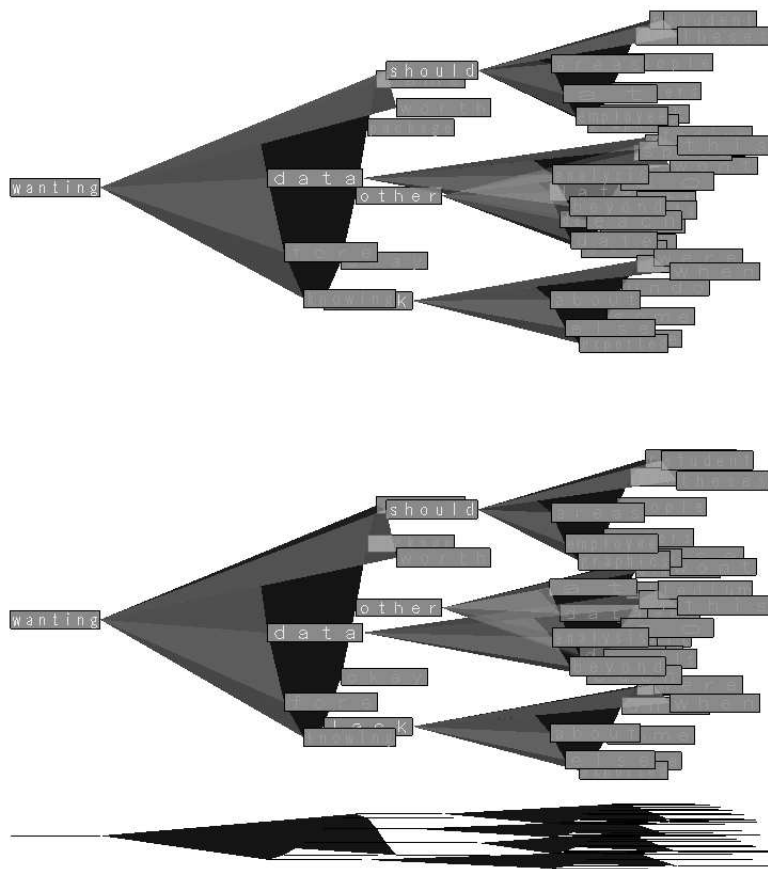


図 6 相違点指摘タスク中の画面

3. ノード名を確認

この観察から、サブタスク1とサブタスク2はほぼ独立した作業だと考えられ、定量的に評価する場合には分離して考える必要があると思われる。

結果 結果としてサブタスク毎の平均終了時間を表17,18に示す。

表17 サブタスク1の平均終了時間(単位:秒)

	両眼立体視あり	両眼立体視なし
運動立体視あり (分散)	32.32 (583.33)	31.55 (356.67)
運動立体視なし (分散)	30.63 (541.65)	23.24 (239.11)

表18 サブタスク2の平均終了時間(単位:秒)

	両眼立体視あり	両眼立体視なし
運動立体視あり (分散)	15.08 (91.52)	13.62 (60.39)
運動立体視なし (分散)	9.79 (17.58)	9.43 (30.17)

なお、タスク終了後に被験者に感想を求めたところ、

- 運動立体視があると画面全体が動くために上下を見比べにくい上に、カーソルでの指示が難しい。
- 両眼立体視は疲れる。
- 両眼立体視で立体的に表示されるとカーソルの動きの予測がつきにくい。

と報告した。

運動立体視、両眼立体視ともない場合が各サブタスクの達成時間が最も短い。被験者の感想と合わせて考えると、

- 相違点指摘タスクでは両眼立体視は被験者を疲労させ、各サブタスクの作業時間を増大させる。これはノード探索タスクに比べて視線の移動距離が増えるため、焦点距離の移動が必要な両眼立体視は疲労しやすかったと考えられる。
- 相違点指摘タスクでは運動立体視は各サブタスクの時間を増大させる。これは「二つの物を見比べる」という本タスク特有の作業では、比較する二つの対象を固定した方が見やすいため、運動立体視の利用が困難であったことが原因と考えられる。

ただし、この実験は被験者の各立体視に対する慣れも十分でなく、被験者数も1名で行ったため、今後十分な検討を要する。

4.4 運動立体視と両眼立体視に関するまとめ

本章では階層構造情報のブラウジング作業を支援する三次元視覚化アプリケーション Cone Tree や Spiral Tree に Fish Tank Virtual Reality 環境を適用した場合、視点追従による運動立体視と両眼立体視がどのような影響を与えるかを調べるため、ノード探索タスクや相違点指摘タスクを用いた実験を行った。その結果から運動立体視と両眼立体視が以下の特徴を持つことが分かった。

- 運動立体視
 - － ノードの探索時においてノードの発見を容易にし、作業に必要とするクリックやドラッグの数を減少させる。
 - － 操作対象の画面上での位置が動くのでマウスによる指示に影響を及ぼす。
 - － 複雑な構造を観察する時に画面が混雑するとかえって見づらくなることがある。
- 両眼立体視

- ノードの探索時においてノードの発見を容易にし、作業に必要とするクリックやドラッグの数を減少させる.
- 視点の移動量が増加するにつれて、利用者の疲労が大きくなるという欠点を持つ.

被験者の主観としてはノードの探索時には両方の併用が好まれた.

また, **Spiral Tree** を利用する場合には **Spiral Tree** の特徴である一覧性の高さのため, 運動立体視の必要性が減少し, 運動立体視の欠点が顕著に現れる傾向があるように思われる.

5. 図書検索への応用

Spiral Tree を実際に情報の検索などに利用する際の有効性や使用感を見るために、図書館の図書検索ツールへの適用を試みた (図 7, 8)。これは一般に図書館では図書を分野によって分類して管理しており、この分類が階層構造を形成しているため Spiral Tree での視覚化に適していると考えられるためである。

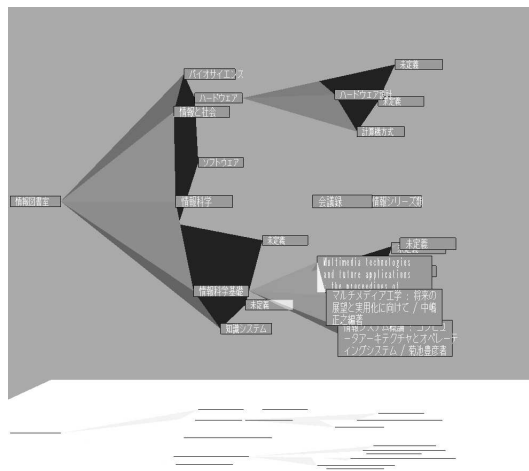


図 7 図書検索用 Spiral Tree

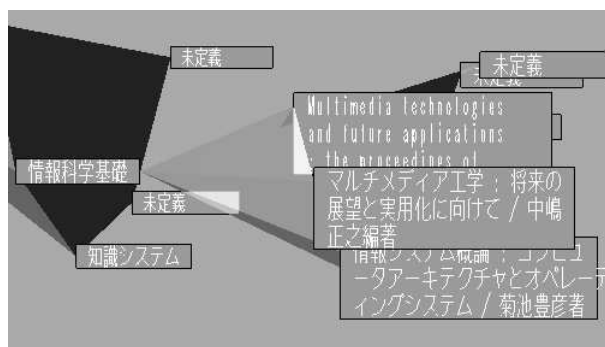


図 8 図 7 の部分拡大図

試作したツールではユーザは **Spiral Tree** によって視覚化された図書の書名リストを探索し、必要な図書のデータを取り出すことができる。現在取り出すことのできるデータは ISBN, 書名, 巻号, 著者名, シリーズ名である。ユーザの必要や嗜好に応じて運動立体視や両眼立体視による奥行き提示を利用することができる。プログラムに入力として与えるデータファイルは、書籍一冊の情報につき一行で、通し番号, ISBN, 書名, 巻号, 書名読み, 著者, 著者読み, 出版事項, シリーズ名, 所在, 請求記号一段目, 請求記号二段目, 請求記号三段目, の 13 項目をコンマで区切っている。従来はこのデータを用いてキーワード検索を行っていたが、書名などにキーワードがヒットしなければ検索結果として得られないため、求める内容の図書を検索するのは非常に困難であった。今回作成した **Spiral Tree** を利用する検索ツールを使用すると、関連分野名からたどっていき関連の深そうな図書の書名を一覧できるため、目的の図書を容易に見つけることができると考えられる。

また、現在本学では附属図書館の電子化が進められており、図書館の電子化が完了すると指定キーワードによる全文検索が可能となる [17]。そこで全文検索を **Spiral Tree** による視覚化と併用することで、ユーザの望む図書の探索が早く行なえるようになると考えられる。これは全文検索により得られた図書に対応するノードを強調表示することで、図書の所属する分野名と全文検索結果の両方の情報を図書の探索に使用できるからである。さらに、キーワードの登場頻度などを色情報などによってユーザに直観的に伝えることも可能である。**Spiral Tree** による図書ブラウジングシステムにより、将来の電子図書館利用時に目的の図書を探索することがより容易になるものと考えられる。

6. むすび

本論文では大規模な情報を利用するための順序つき階層構造情報の表示手法について、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 2 つの比較実験を行なった。これらの実験の結果、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分かった。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が原因であると考えられる。また、Cone Tree や Spiral Tree を利用しての階層構造情報のブラウジング作業時における、3DUI における運動立体視と両眼立体視の利用について検討するために、実験用の Cone Tree や Spiral Tree に運動立体視や両眼立体視を適用し、これらの立体視がどのように利用されるかを調べる評価実験を行なった。その結果、運動立体視はノード探索タスクの作業数を減少する効果があるが、作業時間を増加させることが分かった。このことから、運動立体視によりノードの発見は容易になるが、ノードの指示が困難になっているのではないかと考えられる。また、同実験から、両眼立体視はノード探索タスクにおける探索時間、作業数を減少させる傾向があることも分かった。また、運動立体視と併用することで互いにその欠点を補うことが出来ると考えられる。最後に将来の電子図書館システムのユーザインタフェースへの適用について準備段階の実装について述べ、より人にやさしい図書検索の可能性を論じた。

本論文で提案した順序つき階層構造の三次元視覚化手法 Spiral Tree は従来手法 Cone Tree における配置方針を変更することで、ノードの探索効率を高めることができた。これはユーザがノードを探索する時に順序を利用することを考慮にいた配置方針を採用することで、ユーザがより容易に順序を利用できたことなどが考えられる。このように、三次元ユーザインタフェースを用いるアプリケーションでは、作業対象をどのように視覚化するかを、ユーザがどのように利用するかを考慮にいた上で決定することが重要である。

また、本論文で検討した運動立体視や両眼立体視の特徴については、三次元ユーザインタフェース一般に適用できる部分もあると考えられ、今後の三次元ユーザ

インタフェース研究の足掛かりになるものとして重要であると考えられる。このような奥行き提示手法の利用もユーザがどのように利用するかを考慮して採用するかどうかを決定すべきである。

今後の課題として *Spiral Tree* を 1 つのアプリケーションとして包含するような 3DUI 環境の構築や, 3DUI アプリケーション構築を支援するシステムの開発, これらを通しての 3DUI の特質の分析などが挙げられる。

謝辞

本研究の全過程を通して、直接、懇切なる御指導、御鞭撻を賜ったソフトウェア基礎講座 横矢 直和教授 に心より深謝致します。

本研究の遂行にあたり、終始、有益な御助言と御鞭撻を賜った像情報処理学講座 千原 國宏教授に深く感謝致します。

本研究の全過程を通して、直接、懇切なる御指導、御鞭撻を頂いた、ソフトウェア基礎講座 竹村 治雄助教授 に深く感謝致します。

さらに、日頃より様々な御指導と御助言を頂いた大阪大学基礎工学部情報工学科 萩原 兼一教授 に心より感謝致します。

そして本研究を通じて、有益な御助言を頂いたソフトウェア基礎講座 岩佐 英彦助手、並びに情報科学センター 片山 喜章助手 に厚く感謝致します。

最後に、物心両面において常に温かい御助言を頂いたソフトウェア基礎講座の諸氏、本研究における評価実験に被験者として御協力を頂いた本学学生諸氏、ならびに、ソフトウェア基礎講座事務補佐員 村上 和代さんに感謝致します。

参考文献

- [1] 水野, 青木, 辻: “データ可視化技法を用いた情報検索方式の提案”, SICE 第9回 HI シンポ論文集, pp.79–82, 1993.
- [2] Ahlberg, C. and Shneiderman, B.: “Visual information seeking: Tight coupling of dynamic query filters with starfield displays”, Proc. CHI'94: Human Factors in Computing Systems, pp.313–317, 1994.
- [3] Rao, R. and Card, S. K.: “The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information”, Proc. CHI'94: Human Factors in Computing Systems, pp.318–322, 1994.
- [4] 三輪, 植田, 西田: “3次元グラフィックスを用いたネットワーク型情報の多面的管理と可視化”, SICE 第8回 HI シンポ論文集, pp.613–616, 1992.
- [5] Card, S. K., Robertson, G. G. and Mackinlay, J. D.: “The Information Visualizer: An information workspace”, Proc. CHI'91: Human Factors in Computing Systems, pp.181–188, 1991.
- [6] Robertson, G. G., Card, S. K. and Mackinlay, J. D.: “Information visualization using 3D interactive animation”, Commun. the ACM, Vol.36, No.4, pp.57–71, April 1993.
- [7] Rao, R., Pedersen, J. O., Hearst, M. A., Mackinlay, J. D., Card, S. K., Masinter, L., Halvorsen, P-K. and Robertson, G. G.: “Rich Interaction in the Digital Library”, Commun. the ACM, Vol.38, No.4, pp.29–39, April 1995.
- [8] Mackinlay, J. D., Robertson, G. G. and Card, S. K.: “The perspective wall: Detail and context smoothly integrated”, Proc. CHI'91: Human Factors in Computing Systems, pp.173–179, 1991.

- [9] Robertson, G. G., Mackinlay, J. D. and Card, S. K.: “Cone Trees: Animated Visualizations of Hierarchical Information”, Proc. CHI'91: Human Factors in Computing Systems, pp.189–194, 1991.
- [10] 土本, 竹村, 片山, 萩原: “階層情報の 3 次元表示に関する実験的評価”, 日本ソフトウェア科学会大 11 回大会論文集, pp.277–280, 1994.
- [11] 大隈, 竹村, 片山, 岩佐, 横矢: “順序つき三次元視覚化手法の提案と評価”, 1995 情報秋期全大, 7U-3, 1995.
- [12] 大隈, 竹村, 片山, 岩佐, 横矢: “Spiral tree : 順序つき階層構造の三次元視覚化”, SICE 第 11 回 HI シンポ論文集, pp.369–374, 1995.
- [13] 大隈, 竹村, 片山, 岩佐, 横矢: “順序つき階層構造の三次元視覚化の一手法”, 情報研報, HI63–3, 1995.
- [14] Okuma, T., Takemura, H., Katayama, Y., Iwasa, H., Yokoya, N.: “3D visualization of hierarchical ordered information : Spiral Trees”, Proc. 1st Int. Conf. on Applied Ergonomics (ICAE '96), 1996 (to appear).
- [15] Ware, C., Arthur, K. and Booth, K. S.: “Fish tank virtual reality”, Proc. INTERCHI'93 : Human Factors in Computing Systems, pp.37–42, 1993.
- [16] 土本, 竹村, 片山, 萩原, 横矢: “順序つき三次元視覚化手法の提案と評価”, PRU95–32, 1995.
- [17] 今井, 羽田, 堀井, 山口, 佐藤, 竹村, 横矢, 千原, 嵩: “曼陀羅図書館の構築の試み”, 信学技報, PRU95–32, 1995