

3次元情報可視化におけるクラッタリング回避手法の提案と応用

宮崎麗子[†] 伊藤貴之[†][†]お茶の水女子大学大学院

あらまし 3次元CGを用いた情報可視化技術の増加、および可視化データの大規模化に伴い、可視化結果の遮蔽(クラッタリング)を回避することが課題のひとつとして考えられるようになった。そこで我々は、最小二乗法を用いて3次元オブジェクトのディスプレイ上での重複を回避・調整する手法を提案する。またこの手法を用いて、メタ情報による階層構造を形成する音楽ファイル群を3次元に可視化した結果を示し、本手法によるクラッタリング回避の効果を評価する。

キーワード: 可視化, 3次元表示, クラッタリング, 最小二乗法

Summary We propose a 3D visualization technique which realizes browsing of hierarchical or metadata-based link structures intelligibly while avoiding 3D cluttering. This paper describes the algorithm which relocates the metaphor of contents mapped onto a 3D space so that cluttering is avoided on a 2D projection space by applying the least squares method. Also, we evaluate the effect of avoidance of cluttering using proposed technique.

Key words: Visualization, 3D Representation, Cluttering, least squares method.

1. はじめに

科学データ, マルチメディアコンテンツデータ, Web上のリンクデータなど, 大規模データの可視化の需要に伴い, 画面効率のよい情報可視化手法や, 3次元CGを用いた情報可視化手法がこれまでに多く提案されている。近年では, ただ可視化結果を表示するだけでなく, 可視化結果に対するインタラクティブな操作によって, ユーザが注目するデータをより見やすく表示するような機能を提供するものが多い。このようなインタラクティブな情報探索操作は, コンピュータの演算処理速度の向上に伴い, 3次元情報可視化においても提供されるようになった。

しかし, この情報探索操作がより効率よく機能するためには, なにも操作をしない初期の状態でも, なるべく多くの情報をわかりやすく見せる必要がある。特に3次元情報可視化は, 2次元に比べてより多くの情報を表示することができる一方, データ同士の遮蔽の起こりやすさに起因して, 目的とするオブジェクトをポイントするために視点を移動したり, オブジェクトを回転・拡大す

るなど, 操作が複雑になる。このような背景から, 特に3次元情報可視化においては, 可視化結果の遮蔽(クラッタリング)を回避することが課題になった¹⁾。

そこで本論文では, 重複のない2次元可視化手法よりも小さい面積でクラッタリング回避処理を行うことを目標とする。具体的には, 最小二乗法を用いて3次元オブジェクトのディスプレイ上での重複を回避する手法を提案する。本手法では, 3次元オブジェクトに対して「外接矩形」と呼ばれる2次元投影面上での表示領域を定義する。そして, 互いに重複し合う隣接外接矩形の対について, 重複を解消できる理想的な移動先位置を算出し, 移動先位置との差を最小二乗法によって最小化することで, 3次元オブジェクトの重複を低減する。

筆者らは本手法を, 入れ子状に階層構造を構築する可視化手法「平安京ビュー²⁾³⁾」に適用した。外接矩形の重複具合から算出される重複率は, 画面内に投影されるオブジェクトの数と比例に近い関係にある。筆者らの重複回避手法は, 3次元オブジェクトの配置領域を拡大するため, 全ての3次元オブジェクトに対して一様に重複回避を適用すると, 画面内の単位領域に投影されるオブジェクト数が減少し, 結果として密度の低い可視化結果となってしまう。筆者らはこのトレードオフ問題の対策として, 階層構造を形成する大量のデータを一度に見るときに,

¹⁾ “A Cluttering-Avoided Visualization Technique of 3D Visualization” by Reiko MIYAZAKI, and Takayuki ITOH, (Graduate School of Humanities and Sciences, Ochanomizu University).

ユーザは特定の部分に注目をおきながら、全体像もとらえることを並行に行う、という点に注目した。提案手法では、この点に配慮して重複回避手法を段階的に適用し、注目箇所該当する階層に重みをおいて重複回避を適用しながら、その周囲にも重複回避を適用する、という改善を試みた、これにより、できるだけ多くのオブジェクトをある程度クラッタリングを抑制したまま画面内に収まるように表示させ、可視化データ全体を俯瞰できる状態を保ちつつ、注視領域中のオブジェクトは特にクラッタリングのない状態で表示させる、という可視化結果を実現した。

筆者らは本手法のアプリケーション例として、メタ情報による階層構造(上位から順に、ジャンル アーティスト アルバム トラック)を形成する音楽ファイル群を可視化するミュージックブラウザ⁴⁾⁵⁾を実装した。本論文では、このミュージックブラウザを用いて重複率の削減や可視化に必要な領域サイズを測定し、本手法を評価した結果を提示する。

以下、2章では関連研究について、3章では提案手法について述べる。4章では実装したアプリケーションについて、5章ではその評価について説明する。最後に6章で本論文をまとめる。

2. 関連研究

2.1 3次元情報可視化手法

3次元情報可視化手法のうち、インタラクティブな操作による情報探索機能を提供する代表的な研究を挙げる。

2.1.1 3次元空間においてフィルタリングを行う手法

ConeTrees⁶⁾は、階層データを3次元の木として表示するもので、フォーカスしているパスが手前に表示されるように回転する。3次元空間の奥行きを利用することで、フォーカスしていないサブツリーを画面上に表示することができる。

InformationCube⁷⁾は、半透明表示を効果的に利用した3次元可視化手法である。UNIXディレクトリに代表される階層構造を3次元の入れ子状の立方体として視覚化する。各立方体はディレクトリを表し、子ディレクトリの立方体は親ディレクトリの立方体に包含されている。半透明表示を利用することにより、各立方体は内部が透けて見える。この透明度は内側の立方体ほど低くなるように制御されているため、ある程度の深さまで中の情報を見ることができるとともに、必要以上に深いディレクトリは隠され、図の複雑さを増さない効果が得られる。ConeTreeのように階層構造を木として表示した場合、画面からの溢れを避けることができないが、InformationCubeでは着目したディレクトリから下は必ず一定の範囲に収まる。

H3⁸⁾は、3次元双曲型空間(3D hyperbolic space)でのグラフィック・ナビゲーション機能を提供するライブラリで、20,000ノードを越えるグラフのレイアウトが可能である。注目するノードを中心とした近隣のノードが見やすいようにグラフを変形して表示できる。

2.1.2 3次元空間から2次元に投影してフィルタリングを行う手法

SDM⁹⁾は、オブジェクトセットを選択し操作することができる視覚化環境を提供する。オブジェクトセット(あるいは選択されていないオブジェクト)のスケールを変更し目立たせる方法や、オブジェクトセットだけを選択し、前方に情報を引出す手法などが実現されている。

Reconfigurable Disc Trees¹⁰⁾は、前述のConeTreesに対する拡張および改良である。ConeTreesよりも多くのノードを表示でき、またパラメータを調整することで2次元に投影した時に重複を避けることができる。

GeoTime¹¹⁾は、位置および時刻に関する情報を持つ情報を、3次元空間上に表現するシステムである。位置情報を xy 座標に、時間情報を z 座標に写像し、3次元空間上にイベントの位置と時間を同時に視覚化できる。3次元表示にリンクしたタイムチャート(時間および位置による2次元チャート)を重ねて表示できる。

2.2 従来手法の問題点

このように、従来の研究では情報可視化におけるfocus+context操作を実現するために、3次元空間において対象となるオブジェクトを手前に移動させたり、拡大表示させたり、2次元に投影することによって、ユーザが注目するデータをより見やすく表示していた。これに対し、全体を俯瞰して次に注目すべき情報を探す操作も情報探索において重要である。しかし従来研究の多くにおいて、現在注目していない他のデータの視認性については特にコントロールされないため、画面外や3次元空間内の奥などに表示されて見えづらいことが多い。これを解決するためには、例えば同じ画面内に2つの表示領域を持たせ、一方は全体の俯瞰、もう一方は注目したデータの詳細、というように複数の表示領域を見比べる手法などが考えられる。しかし、同一データを複数の可視化方法で見ることが、単一の画面で見ることよりもユーザ負荷が高く、また画面サイズに制約があるためそれぞれの可視化領域も小さくなってしまふ。

3. 提案手法

本論文では、従来手法の問題点を解決するために、同一空間におけるfocus+context操作を効率よく行うことを考える。ここで言う、focus+context操作とは、focus操作はデータ中のユーザの注目部位を見やすく表示させ

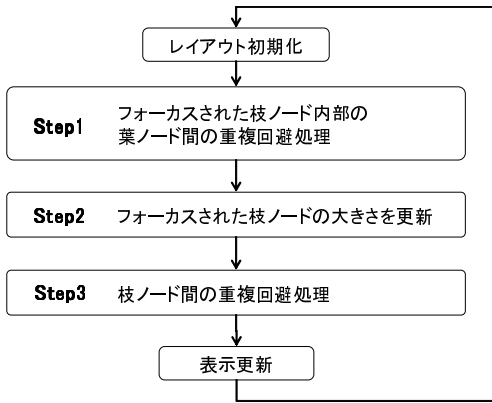


図1 手法全体の処理の流れ
Fig. 1 Total processing flow.

る操作，context 操作とはユーザの注目の有無にかかわらず，データを全体的に見やすく表示させる操作を示す。そこで，操作をしない状態でも視認性よく可視化結果を表示させる手段として，3次元可視化における重複回避手法を提案する。本手法では，3次元空間における可視化結果の重複を段階的に回避させることで，注目するデータは最下層までクラッタリングのない状態で表示させ，注目していないデータもその親階層はクラッタリングのない状態で表示させる。

3.1 処理の流れ

図1に提案手法全体の処理の流れを示す。説明のため，階層型データの親子関係を，枝ノード(親)および葉ノード(子)と呼ぶことにする。まず，ユーザの操作によりフォーカスされた枝ノード内の葉ノード間の重複を回避する(Step1)。このとき葉ノード間に重複があれば，その重複回避によってこの枝ノード全体の表示面積が広がる(Step2)。次に，その拡大された枝ノードを含むすべての枝ノード間の重複を回避する(Step3)。

なお，本手法では重複回避処理の前に，各ノードの配置を初期位置に戻す。この初期化処理を省略すると，重複回避処理後の配置結果に対して次々にノードの移動処理が行われることで，結果として同じ枝ノードをフォーカスしても毎回異なるレイアウトを生じてしまい，ユーザに混乱を与える原因となる。

3.2 重複回避処理

本節では，重複回避処理の具体的な流れを示す(図2)。

3.2.1 外接矩形の定義

図3(a)に，3次元の可視化結果を2次元投影面上で囲むような四角形を定義した様子を示す。以降，これを「外接矩形」と呼ぶことにする。図1のStep1ではこの外接矩形はフォーカスされた枝ノード内部の葉ノードを

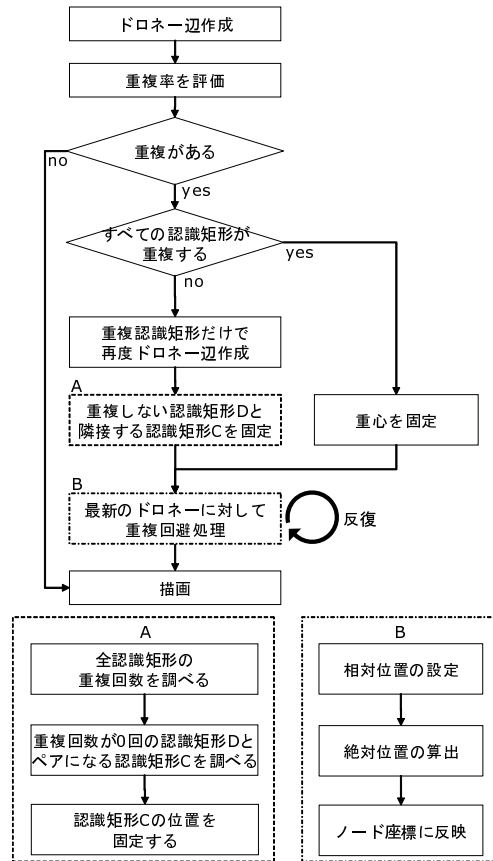


図2 重複回避処理の流れ
Fig. 2 Processing flow of avoidance of cluttering.

囲むように，図1のStep3ではこの外接矩形は枝ノードを囲むように設定する。この外接矩形の位置と大きさは3次元空間のノードの座標から求められる。これにより，3次元重複回避の問題は，2次元矩形の重複回避の問題として近似できる。

3.2.2 最小二乗法による外接矩形位置の最適化

3.2.1節で得られた各外接矩形について，2次元投影面の座標系において以下の処理を行うことにより，外接矩形間の重複を抑えた配置を算出する。この処理の流れを図2に示す。

1. 外接矩形の中心を入力点群としたドローン三角形分割を行う。
2. ドローン辺で結ばれた外接矩形の対において，重複する対の数とその重複面積を求める。
3. 2において重複がある場合，重複がなくなるような理想的な相対位置を求める。具体的には，各外接矩形の対について，重複がない場合には現在の位置を，重複がある場合には重ならなくなるように両者を結

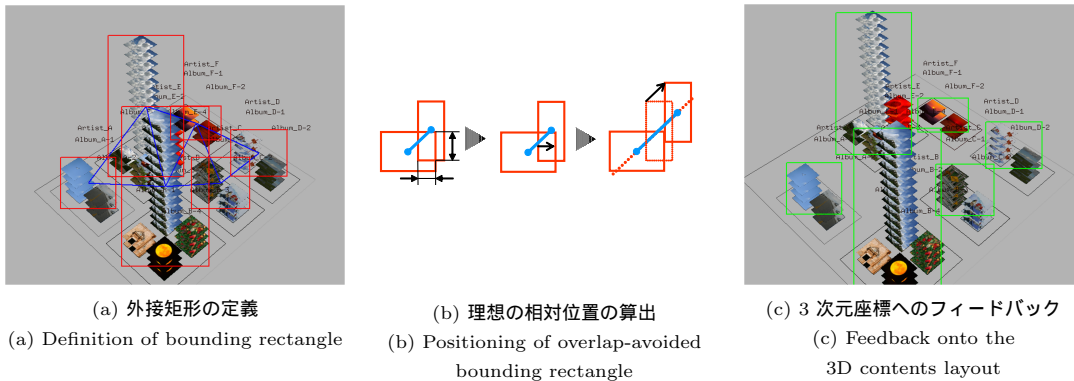


図 3 重複回避の流れ

Fig. 3 Avoidance of cluttering.

ぶ直線上に平行移動した位置を、理想的な相対位置とする。重複部の x 軸方向の長さや y 軸方向の長さを比較し、より重複の長さの短い方向に重複がなくなるように外接矩形を平行移動させる (図 3(b))。

4. 3 で得られた外接矩形の座標を自由変数として、外接矩形間の現在の相対位置と理想的な相対位置との差を最小二乗法で最小化する。このとき、2 において全外接矩形に重複がある場合は、全外接矩形の重心を処理前の全外接矩形の重心に合わせるようにする。

以上を定式化した、外接矩形位置の最適化のための評価式を以下に示す：

$$\arg \min \{ \sum_{i,j} |v_i - v_j - dv_{ij}|^2 + |(\frac{1}{n} \sum_i v_i) - c|^2 \} (1)$$

v_i と v_j はドローネー辺の始点と終点、 dv_{ij} は v_j からみた v_i の理想的な相対位置を示す。つまり、 v_i および v_j の平行移動後の位置を v'_i および v'_j とすると、 $dv_{ij} = v'_i - v'_j$ である。 n はドローネー辺を構成する頂点数、つまり外接矩形数を示す。 c は全外接矩形の重心である。式 (1) の第 1 項と第 2 項の和を最小化するように v_i の各位置を求める。この式において第 1 項は、「2 頂点の相対位置 $v_i - v_j$ 」と「重複のない理想の相対位置 dv_{ij} 」との差を最小化して、理想の状態に近づける意味を持つ。第 2 項は、第 1 項だけの制約だと v_i の処理後の位置が定まらないので、処理後の外接矩形の重心位置を処理前の位置から動かさないようにするためのものである。この手法はアイコンの重複を解消する Watanabe らの手法¹²⁾に類似するものである。

なお、上記処理手順 2. において、一部の外接矩形間だけが重複する場合は、重複する外接矩形間のみ重複回避を適用する。重複しない外接矩形にはこの処理を適用せず、重複回避処理前後でその位置が動かないように固

定する (図 4)。

以上 3.2.1 節~3.2.2 節の処理を、外接矩形の重複が十分少なくなるまで反復処理する。ただし筆者らの実装では、この反復回数を最大 10 回としている。

3.2.3 3 次元座標へのフィードバック

3.2.2 節で得られた理想の外接矩形の投影面座標を、以下の処理を行うことにより 3 次元空間座標に変換し、各外接矩形に対応するグループの新たな座標とする (図 5)。

1. 投影面座標における理想の外接矩形の中心位置を (x', y', z') とする。ここで、 (x', y') は式 (1) におけるベクトル変数 v_i の x 座標値および y 座標値である。
2. $y = (\text{各グループで最も背の高いオブジェクトと最も背の低いオブジェクトの高さの平均})$ という方程式で与えられる平面を 3 次元空間に描画し、 (x', y') における z' の値をデスバツファから取得する (ただしこの平面は表示しない)。
3. (x', y', z') に対応する 3 次元空間座標値 (x, y, z) を算出する。

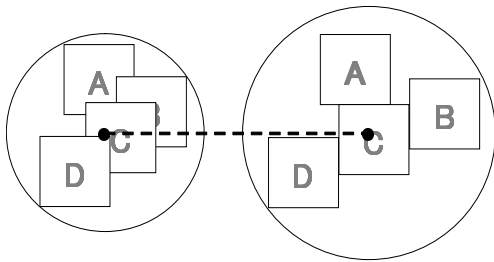
4. 実装

本章では、3 章で述べた手法を既存の 3 次元情報可視化手法に応用した例を示す。

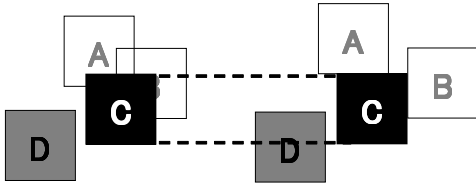
4.1 前提とする 3 次元可視化方法

4.1.1 平安京ビュー

本論文で提案手法を適用する 3 次元可視化手法について述べる。本論文では、階層構造をもつデータを可視化する手法「平安京ビュー²⁾³⁾」を用いる。平安京ビューは階層型データの葉ノードを長方形のアイコンに、枝ノードを長方形の枠に対応させ、階層構造を 2 次元の長方形群の入れ子構造で表現し、その全体を一画面に表示することを目標とした手法である。計算機のファイルシステムに例えるなら、葉ノードはファイルに、枝ノードはディ



すべての認識矩形が重複する場合:
認識矩形A~Dの重心を固定



一部の認識矩形が重複する場合:
重複しない認識矩形Dと隣接する認識矩形Cを固定

図4 外接矩形位置の固定

Fig. 4 Pinning of bounding boxes.

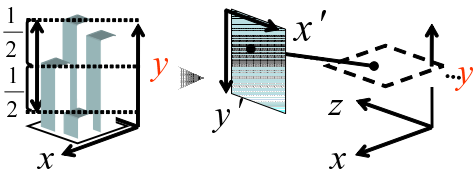


図5 3次元座標へのフィードバック

Fig. 5 Feedback to the 3D space.

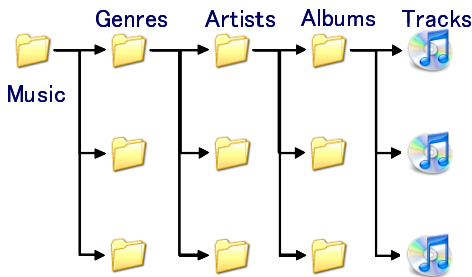


図6 音楽ファイルの階層構造

Fig. 6 Hierarchical structure of music metadata.

レクトリに相当する。平安京ビューは、階層型データ中の葉ノードと枝ノードの親子関係よりも、階層型データ全体に分布する葉ノード群を全て一画面に表現することに主眼を置いた手法である。また平安京ビューには、葉ノ

ードもつパラメータ情報をノードの高さに割り当て、3次元で表示する機能がある。本論文ではこれを応用する。

4.1.2 平安京ビューの3次元化による音楽ファイル構造の可視化

図6に示すような従来の音楽ファイルの階層構造を、平安京ビューを使って可視化する(図8(v1)参照)ことを考える。平安京ビューは基本的に2次元表示であるが、本論文ではこの平安京ビューの葉ノードをパイル状に積み重ねることで3次元の可視化結果を生成する(図8(v2)参照)。以下、この可視化手法における重複回避手法の議論を進める。なお、この表示におけるオブジェクトの最小単位を「ノード」、ノードを積み重ねた単位を「パイル」、パイルを枠で囲んだ単位を「フレーム」と呼ぶことにする。

筆者らはこの3次元情報可視化手法を応用して、階層構造を形成する音楽ファイル群を可視化するミュージックブラウザを実装した。図8のv3に示す可視化結果例では、フレームで表現されたグループがジャンル、パイルがアーティスト、ノードがトラックを表している。よって、コンピレーションアルバムなど同一アルバム内に異なるアーティストのトラックがある場合は、異なるパイルに分類される。また、同一アーティストによる複数個のアルバムは、同じパイルに分類される。

4.2 ミュージックブラウザのGUI

筆者らが実装したミュージックブラウザのGUI(図8(v3)参照)について説明する。このミュージックブラウザでは画面全体に音楽ファイル群を表示し、マウスにより選曲操作をさせる。画面左上部には、音楽ファイルに関するメタ情報をオーバーレイさせ、マウスにより選択された曲、およびそのメタ情報を表示する。開発言語はC++、グラフィックライブラリはOpenGLを使用している。

4.3 可視化データ

筆者らは、音楽データ管理ソフトウェア iTunes¹⁾に登録された各音楽ファイルを対象として、iTunesAPIを用いて、ジャンル名、アーティスト名、アルバム名、トラック名、音楽ファイルへのリンク、ジャケット写真へのリンクを抽出するプログラムを実装した。

そして、筆者の iTunes に登録された音楽ファイル群から、本論文のための可視化データ2種類(以下、データ1、データ2と称する)を作成した。さらにこれらの情報と、実装したミュージックブラウザにおける階層構造を、表1のように対応付けた。

1) iTunes は、Apple 社の登録商法である。



図7 レイアウト変更時のアニメーション
Fig. 7 Animation of changing layout.

表 1 可視化データの階層構造

Table 1 Hierarchical structure of visualization data.

階層	グループ	メタ情報	データ 1	データ 2
1	フレーム	ジャンル	5	16
2	パイル	アーティスト	39	134
3	パイルの一部	アルバム	30	52
4	ノード	トラック	188	430

4.4 操作仕様

マウス操作によってフォーカスされるフレームが切り替わると、本手法は重複回避処理を適用し、フォーカスされたフレームおよびその内部のパイルの重複を最大限回避するように、レイアウトを更新する。このとき、レイアウトの更新前と更新後の各オブジェクトの位置を、滑らかなイーザリング付きのアニメーションで補間することで、ユーザに急激な印象を与えないようにする(図7)。このとき、重複回避の処理時間はフレームの数とその重複数に比例し、例えば4.3節のデータ2を用いた場合には、重複があるかどうかの判定に約30msec、これに加え重複があった場合はその重複数に比例して最大141msecかかる。また、アニメーション表示の処理時間は、レイアウト更新前と更新後の補間に30フレームを費やしてアニメーション表示するのに、平均140msec(1フレームあたり約4.7msec)かかる。

また、マウスの右ボタンを押下しながら縦横方向にドラッグ操作をすると、縦横方向にビューポイントを回転することができる。このときも、重複回避処理の時間とアニメーション表示処理の時間は前述の通りフレーム数とその重複数に比例する。

ユーザは、このようにフォーカスを変えながら任意の曲が見つかったところで、対応するノードにカーソルをあててクリックすることで、曲を再生することができる。

5. 評価

提案手法の有効性を確認するために、4章で述べたミュージックブラウザを用いて定量評価を行った。1章でも述べたように、本論文の目標は重複のない2次元可視化手法よりも小さい面積でクラッタリング回避処理を行うことにある。それを実証するために本章では、「提案手法によりオブジェクト間の重複率が低下し、かつ可視化領域は2次元の可視化結果よりも小さい状態を保っている」ということを評価する。

5.1 評価方法

4章で述べたミュージックブラウザを用いて、提案手法の有効性を評価した。図8に示す3つの可視化結果(v1, v2, v3)において、後述の3つの評価尺度(e1, e2, e3)を数値評価し比較した。

可視化手法

v1: 平安京ビュー

平安京ビューによる2次元可視化手法。

v2: 重複回避処理前

クラッタリング回避処理を行う前の3次元可視化手法。

v3: 重複回避処理後

提案手法を用いて、まずフォーカスされたフレーム内の各パイルに対応する外接矩形を重複回避させ、さらに全フレームに対応する外接矩形を重複回避させる。フォーカスするフレームにより評価結果が変わるのを避けるため、全フレームを順にフォーカスし、その測定結果の平均値を求める。

評価尺度

e1: フォーカスフレームの重複率(%)

フォーカスしたフレーム内部の全パイルに対応する外接矩形において、重複のない領域(図9中のグレー部)の面積を抽出し、全外接矩形に対する重複率を算出する。

$$(1 - \sum_i \frac{A_i}{R_i}) * 100 \quad (2)$$

A_i は各外接矩形における重複のない領域(図9中のグレー部)の面積、 R_i は各外接矩形の面積である。この値

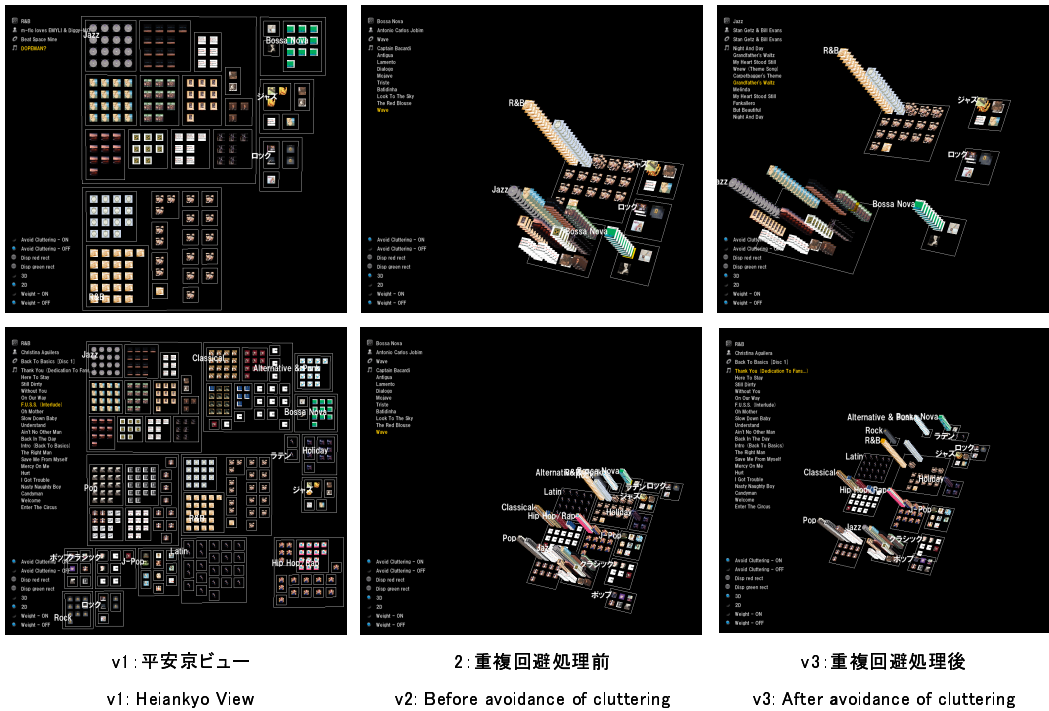


図8 可視化結果 (上段：データ1, 下段：データ2)
Fig. 8 Visualization results. (Top:Dataset1, Bottom:Dataset2)

が低いほど、フォーカスしたフレーム内部のパイル間の遮蔽が小さい、とすることができる。

e2: 全フレームの重複率 (%)

全フレームに対応する外接矩形において、e1と同様に重複率を算出する。この値が低いほど、全域にわたってフレーム間の遮蔽が小さい、とすることができる。

e3: 可視化領域拡大率 (倍)

全フレームに対応する外接矩形を囲む最小の四角形 (図9中の点線の四角形) の面積が、前述の可視化手法 v2「重複回避処理前」に比べて何倍拡大したかを示す。この値が低いほど、表示面積が小さくて済むとすることができる。

ウィンドウ

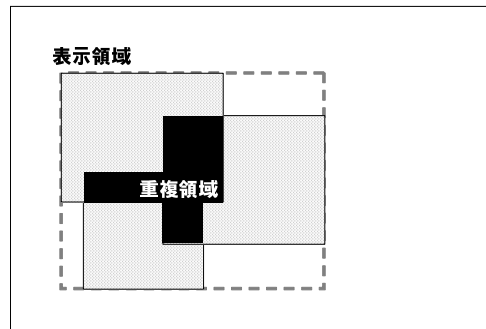


図9 評価した領域
Fig. 9 Evaluated area.

5.2 評価結果

表 2(a) に表 1 のデータ 1 (フレーム (1) ~ (5)) を、表 2(b) に表 1 のデータ 1 (フレーム (1) ~ (16)) を可視化手法 v1 ~ v3 を用いて可視化し、評価尺度 e1 ~ e3 について測定した結果を示す。なお v1 については、ジャケット写真の内容を目視で確認できる程度の大きさにそろえて、実際のユーザ操作に耐える拡大率で表示させた。具体的には、v1 のジャケット写真の 1 辺の長さと、v2, v3 のジャケット写真の手前の 1 辺の長さが同じになるようにした。

5.3 考察

まず、表 2(a) の e1「フォーカスフレームの重複率」を見ると、フレーム (1) ~ (5) のうちフレーム (3) ~ (5) のフォーカス時に内部のパイル間では重複がなかったため、重複回避処理は行われていない。フレーム (1) の内部では、一部のパイル間で重複が起きたため、それに対応する外接矩形間のみで重複回避処理が行われ、他のパイルの位置は処理前後で動かない。フレーム (2) の内部では、全パイル間で重複が起きたため、それに対応する全外接

表 2 重複回避処理前後の比較
Table 2 Comparison between before and after avoidance of cluttering.

(a) データ 1

	e1		e2		e3	
	処理前	処理後	処理前	処理後	v1/v2	v3/v2
	v2	v3	v2	v3		
(1)	42.69	3.70	47.11	8.70	2.42	1.43
(2)	78.03	20.38	57.78	8.02	2.35	1.73
(3)			52.60	6.01	3.07	1.57
(4)			52.60	6.01	3.07	1.57
(5)			52.60	6.01	3.07	1.57
平均	60.36	12.04	52.54	6.95	2.80	1.57

(b) データ 2

	e1		e2		e3	
	処理前	処理後	処理前	処理後	v1/v2	v3/v2
	v2	v3	v2	v3		
(1)	0.25	0.25	66.25	48.80	3.39	1.11
(2)	2.39	39.84	66.80	49.01	3.39	1.11
(3)	55.26	35.55	67.24	42.76	3.38	1.19
(4)	74.43	5.16	61.87	45.11	2.75	1.15
(5)	77.73	16.57	75.32	54.81	3.39	1.19
(6)	11	0.71	68.18	50.79	3.39	1.12
(7)	42.12	3.62	69.47	51.87	3.39	1.17
(8)	0.33	0.33	66.25	48.71	3.39	1.11
(9)			66.28	48.75	3.39	1.11
(10)			66.28	48.75	3.39	1.11
(11)			66.28	48.75	3.39	1.11
(12)			66.28	48.75	3.39	1.11
(13)			66.28	48.75	3.39	1.11
(14)			66.28	48.75	3.39	1.11
(15)			66.28	48.75	3.39	1.11
(16)			66.28	48.75	3.39	1.11
平均	32.94	12.75	66.98	48.86	3.35	1.13

e1: フォーカスフレームの重複率 (%)
 e2: 全フレームの重複率 (%)
 e3: 可視化領域拡大率 (倍)

矩形で重複回避処理が行われた。以上の結果から、フォーカスフレーム周辺の重複の度合いにかかわらず、重複回避効果を得ていることがわかる。

同様に、表 2(b) の e1「フォーカスフレームの重複率」を見ると、フレーム (1)~(16) のうちフレーム (9)~(16) のフォーカス時に内部のパイル間では重複がなかったため、重複回避処理は行われていない。フレーム (1)~(8) の内部では、一部のパイル間での重複あるいは全パイル間で重複が起きたため、対応する外接矩形間で重複回避処理が行われていることがわかる。

さらに、e1「フォーカスフレームの重複率」について

は、v3 のクラッタリング回避処理をしたものが、v2 の処理をしないものにくらべて重複率を大幅に下げることができた。また、e2「全フレームの重複率」についても、v3 のクラッタリング回避処理をしたものが、v2 の処理をしないものにくらべて重複率を大幅に下げることができた。この e1 と e2 の結果は e3「可視化領域拡大率」とトレードオフの関係にある。重複が回避されると、それだけ全体の表示面積が広がることになる。本論文の目的は、1 章で述べたとおり重複のない 2 次元可視化手法よりも小さい面積でクラッタリング回避処理を行うことである。e3 の各可視化手法の面積拡大率を見比べると、データ 1 の場合、本手法は 2 次元可視化手法の平安京ビューの約 6 割の面積で、データ 2 の場合、約 3 割の面積で可視化できることが示された。これより、「提案手法によりオブジェクト間の重複率が低下し、かつ可視化領域は 2 次元の可視化結果よりも小さい状態を保っている」という望ましい結果が得られたことがわかる。

6. む す び

本論文では、3 次元可視化手法において「外接矩形」と呼ばれる 2 次元投影面上での表示領域を定義し、最小二乗法を用いてこの外接矩形間の重複回避処理を行うことで、可視化結果のクラッタリングを回避する手法を提案した。また、本手法の有効性を確認するために、階層型データ可視化手法「平安京ビュー」に本手法を適用し、音楽ファイルの階層構造を可視化するミュージックブラウザを実装し、その重複回避の効果を評価した。この評価結果から、「提案手法によりオブジェクト間の重複率が低下し、かつ可視化領域は 2 次元の可視化結果よりも小さい状態を保っている」という効果を実証された。

今後の課題としては、まず大規模データへの対応が考えられる。5 章で用いたデータよりも大規模なデータを本手法に適用した場合、重複回避処理によってパイルやフレームの配置領域が非常に大きくなり、可視化データ全体を一画面に収めることが難しくなる可能性がある。このような大規模なデータにおいても focus+context 操作を実現させるためには、ユーザが目目していない部分においては低階層部分を表示せず、高階層部分のみを小さく表示する、などの方法が考えられる。

また、本手法を大規模なデータにおいて実用した際に、パイルの歪みにより可視化データが見づらくなる可能性が考えられる。そこで、透視投影によるこのような歪みに関する検証と、平行投影による実装との比較評価も課題である。

さらに、重複回避の効果を向上する一手段として、パイルやフレームの 2 次元投影面上の形状に、より近い外

接形状を導入することが考えられる。例えば外接形状に、凸多角形や楕円などを適用することで、重複回避のための外接形状間の理想位置をより近くし、小さい配置領域に可視化データ全体を収められる可能性がある。

(2009年2月20日受付)

(2009年10月21日再受付)

宮崎麗子



2002年お茶の水女子大学理学部情報科学科卒業。2004年お茶の水女子大学大学院人間文化研究科博士前期課程修了。同年ソニー(株)入社。お茶の水女子大学大学院人間文化研究科博士後期課程在学中。情報処理学会会員。

伊藤貴之



1992年早稲田大学大学院理工学研究科修士課程修了。同年日本アイ・ビー・エム(株)入社。1997年博士(工学)。2000年カーネギーメロン大学客員研究員。2003年京都大学大学院情報学研究科 COE 研究員(客員助教授相当)兼任。2005年お茶の水女子大学理学部情報科学科助教授(現在准教授)。ACM, IEEE CS, 他会員。

参考文献

- 1) Elmqvist, N. and Tudoreanu, M. E.: Evaluating the Effectiveness of Occlusion Reduction Techniques for 3D Virtual Environments, *the ACM Symposium on Virtual Reality Software and Technology 2006*, pp. 9–18 (2006).
- 2) 伊藤貴之, 山口裕美, 小山田耕二: 長方形の入れ子構造による階層型データ可視化手法の計算時間および画面占有面積の改善, *可視化情報学会論文集*, Vol. 26, No. 6, pp. 51–61 (2006).
- 3) Itoh, T., Takakura, H., Sawada, A. and Koyamada, K.: Hierarchical Visualization of Network Intrusion Detection Data, *IEEE COMPUTER GRAPHICS AND APPLICATIONS*, Vol. 26, No. 2, pp. 40–47 (2006).
- 4) 宮崎麗子, 伊藤貴之: クラッタリングを回避する3次元情報可視化手法を用いたコンテンツブラウザ, 第15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(WISS), pp. 113–114 (2007).
- 5) 宮崎麗子, 伊藤貴之: 重み付き最小二乗法を用いたクラッタリング回避技法の提案, 第131回情報処理学会グラフィクスとCAD研究会, 研究報告 2008-CG-131, pp. 25–30 (2008).
- 6) Robertson, G. G., Mackinlay, J. D. and Card, S. K.: Cone Trees: animated 3D visualizations of hierarchical information, *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 189–194 (1991).
- 7) Rekimoto, J. and Green, M.: The Information Cube: Using Transparency in 3D Information Visualization, *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, pp. 125–132 (1993).
- 8) Munzner, T. et al.: H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space, *Proceedings of the 1997 IEEE Symposium on Information Visualization*, pp. 2–10 (1997).
- 9) Chuah, M. C., Roth, S. F., Mattis, J. and Kolojejchick, J.: SDM: Selective Dynamic Manipulation of Visualizations, *UIST'95 User Interface Software and Technology*, ACM, pp. 61–70 (1995).
- 10) Jeong, C. and Pang, A.: Reconfigurable disc trees for visualizing large hierarchical information space, *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pp. 19–25 (1998).
- 11) Kapler, T. and Wright, W.: GeoTime Information Visualization, *INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*, pp. 25–32 (2004).
- 12) Watanabe, N., Washida, M. and Igarashi, T.: Bubble Clusters: An Interface for Manipulating Spatial Aggregation of Graphical Objects, *ACM User Interface Software and Technology*, pp. 173–182 (2007).