

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

宮崎 麗子 伊藤 貴之

お茶の水女子大学大学院

E-mail: reiko@is.ocha.ac.jp

3次元を用いた可視化技術の増加、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、重み付き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。

A Cluttering-Avoided Visualization Technique Using the Weighted Least Squares Method

Reiko MIYAZAKI Takayuki ITOH

Graduate School of Humanities and Sciences, Ochanomizu University

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 weighted least squares method. And we evaluate the effect of avoidance of cluttering using proposed technique.

1. はじめに

大規模データの可視化や3次元を用いた可視化の増加に伴い、可視化結果から必要なデータを抽出することや、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として考えられるようになった[1][2]。特に3次元可視化においては、2次元に比べて表示できる情報量は多いものの、クラッタリング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とが考えられる。このような問題を解決するために、力学モデルによってクラッタリングを回避する手法が提案されている[3]。

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する技法を提案し、そ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した[4]。具体的には、入れ子状に階層構造を構築する視覚化手法「平安京ビュー」[5]に対して、3次元のパイル構造を適用し、同一面積に対する各ノードの大きさを拡大するとともにこの3次元のノード群に対して「認識矩形」と呼ばれる2次元投影面上での表示領域を定義し、この認識矩形のすべての対について、最小二乗法による重複回避処理を行った。

しかし多くの場合において、ユーザは可視化結果の全体を眺めるというより、特定の部分を集中的に眺めてい

る。本稿では、この特定の部分を「フォーカス」と呼ぶ。文献[4]による可視化結果を用いて選曲を行うと、フォーカス中の認識矩形についてはクラッタリングが起こらない状態が望まれるが、全認識矩形についてクラッタリング回避処理を行うと、多くのオブジェクトが画面の外にはみ出る。この場合は、多少の重複を許しても、できるだけ多くのオブジェクトが画面内に収まっていた方が、「情報を俯瞰しつつフォーカス中のオブジェクトはクラッタリングのない状態で表示させたい」という当初の目的を達成できると考えられた。

そこで本稿では、新たに「重み付き最小二乗法」を用いたクラッタリング回避技法を実装し、この問題の解決とその評価を試みる。

2. 平安京ビュー

「平安京ビュー」は図1に示す通り、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おいた手法である。「平安京ビュー」と同様の目標を有する可視化手法として、Treemaps[6]の拡張手法であるQuantum Treemap[7]があげられる。文献[3]では、Quantum Treemapを含む関連手法と「平安京ビュー」の定量的評価結果がまとめられている。この評価結果の中で「平安京ビュー」は、長方形領域のアスペクト比、類似データにおける表示結果の類似度において関連手法よりも良好な結果を示している。

「平安京ビュー」の適用事例として、計算機ネットワーク不正侵入履歴、クレジットカード不正利用履歴、薬物実験情報、遺伝子や蛋白質などの生命情報、医療シミュレーション、原子力システムの計測情報、新聞記事データベース、写真コレクション、といった分野での可視化の事例が報告されている。

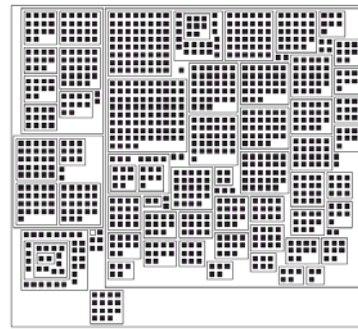


図1. 平安京ビューによる階層型データの可視化

3. 提案手法

提案手法の処理の流れを図2に示す。

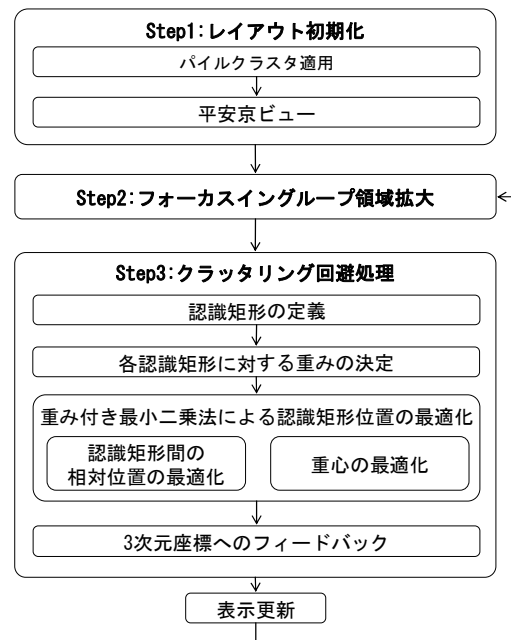


図2. 提案手法による処理の流れ

Step 1 では、平安京ビューにおいて最も内側に表示される枠内の葉ノードを、パイル状（組積状）に重ねてひとつのクラスタとし3次元で表現する。図5(A)に平安京ビューによる可視化結果、図5(B)にパイルクラスタを適用した可視化結果を示す。パイルクラスタ適用時は、ノード数の多い順にクラスタをソートしたあとに、平安京ビューによるレイアウトを行う。これにより、最下層の枠内（グループ内）では、奥から背の高いクラスタが並ぶため、簡易的にクラッタリングを抑制できる。

Step 2 では、この表示に対して、マウスのカーソルの当たったグループを拡大する。この状態を「フォーカスイン状態」と呼ぶ。これとは反対に、カーソルの当たらない状態を「フォーカスアウト状態」と呼ぶ。フォーカ

スイン状態のときだけ、そのグループを拡大するには、フォーカスアウト状態ではグループ内のクラスタ間の間隔は詰めることで表示面積を必要最小限にとどめ、フォーカスイン状態では間隔を広げてクラッタリングを抑制するねらいがある。

Step 3 では、最小二乗法を用いたクラッタリング回避処理を行う。このステップにおいて、マウスポインタからの距離により定義された重みを考慮した場合(図 5(D))と考慮しない場合(図 5(C))を比較することが、本稿の主旨である。以下、このクラッタリング回避処理の詳細について述べる。

3.1. 認識矩形の定義

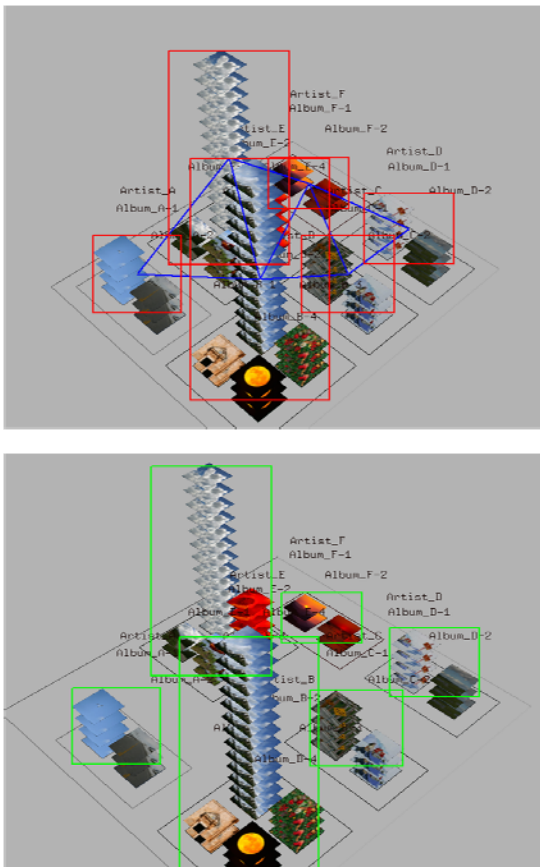


図 3. (上)クラッタリング回避処理前 (下)クラッタリング回避処理後

図 5(B)の例では、枠で表現されたグループがジャンル、パイルクラスタがアーティスト、アイコン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、異なるパイルクラスタに分類される。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

クラスタに分類される。本稿では、2次元投影面上において重複を回避するためのひとつの単位を、このうちのグループとする。

図 3 に、各グループ内のパイルクラスタを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。これを「認識矩形」と呼ぶことにする。この認識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次元クラッタリング回避の問題は、2次元矩形の重複回避の問題として近似できる。

3.2. 各認識矩形に対する重みの決定

マウスのカーソル位置から各認識矩形への距離を正規化した値を算出し、これを重み付き最小二乗法における重みとして用いる。これにより、カーソル位置に近いグループに、よりクラッタリング回避処理が反映され、反対にカーソル位置から遠いグループに対する重複は許すようにする。カーソルが異なるグループをポイントするタイミングでクラッタリング回避処理が行われる。

3.3. 重み付き最小二乗法による認識矩形位置の最適化

3.1 節で得られた各認識矩形について、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、認識矩形間の重複を抑えた配置を算出する。この手法はアイコンの重複を解消する Watanabe らの手法[8]に類似するものである。

1. 認識矩形の中心を入力点群としたドロネー三角形分割を行う。
2. ドロネー辺で結ばれた認識矩形の対において、重なり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、各認識矩形の対について、重なりがない場合には現在の位置を、重なりがある場合には重ならなくなるように両者を結ぶ直線上に平行移動した位置を、理想的な相対位置とする。このとき、各認識矩形を重複しない位置に最短距離でずらせるよう、重複部の xy 方向の長短も考慮する。
3. 2 で得られた認識矩形の座標を自由変数として、認識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、画面の中心を算出される認識矩形全体の重心にすることで表示領域を固定する。

以上を定式化したものを以下に示す：

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

$$\sum_i |w_i| = 1$$

v_i と v_j はドロ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. n はドロネー辺を構成する頂点数, つまり認識矩形数を示す. c は全認識矩形の重心, つまり本手法では画面の中心座標である. w_i は 3.2 節で述べたマウスのカーソル位置から各認識矩形への距離を正規化した値である. この算出値を最小化するように v_i の各位置を求める.

以上を, カーソルが移動してフォーカスイングループが切り替わる度に, 認識矩形の重複がなくなるまで数回繰り返す. 図 5(D)はこれらの処理を 10 回繰り返した結果である.

3.4. 3次元座標へのフィードバック

3.3 節で得られた理想の認識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, 各認識矩形に対応するグループの新たな座標とする.

1. 投影面座標における理想の認識矩形の中心位置を (x',y',z') とする. (x',y' は 3.3 節で求めた座標)
2. $y=($ 各グループで最も高いパイルクラスタと最も低いパイルクラスタの高さの平均) という方程式の平面を 3 次元空間に描画し, (x',y') における z' の値をデブスバッファから取得する. (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 OpenGL の関数で算出する.

4. 評価

本手法の有効性を確認するために定量評価を行った. 本稿の目的として, 「3 次元で本手法によるクラッタリング回避処理をした場合, クラッタリング回避処理をしない場合に比べてクラッタリング率が低く, かつ表示領域は 2 次元の可視化結果よりも小さい」と言えることが望ましい.

4.1. 評価方法

今回行った評価では, 以下 A から D までの 4 つの可視化結果において, 後述の 4 つのポイントを数値評価し比較した.

■ 比較した可視化結果

(A) 平安京ビュー

文献[3]の平安京ビューの可視化結果.

(B) パイルクラスタ適用

クラッタリング回避処理を行う前段階の可視化結果.

(C) クラッタリング回避処理適用 (重みなし最小二乗法)

各認識矩形への重みを考慮しないクラッタリング回避処理をした可視化結果. 数式で示すと以下の通りである:

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

この算出値を最小にするように v_i の各位置を求める.

(D) クラッタリング回避処理適用 (重み付き最小二乗法)

本稿で示した各認識矩形に重みを考慮してクラッタリング回避処理をした可視化結果.

■ 評価ポイント

① フォーカスイン認識矩形のクラッタリング面積占有率: e_1

フォーカスしたグループに対応するフォーカスイン認識矩形 (図4 黄色の矩形) のうち, 重複領域 (図4 斜線部) の占める割合を示す. この値が低いほど, フォーカス中のオブジェクトの遮蔽が小さい, とすることができる.

② 全認識矩形のクラッタリング面積占有率: e_2

フォーカスイン認識矩形とフォーカスアウト認識矩形 (図4 水色の矩形) の占める面積のうち, 重複領域の占める割合を示す. この値が低いほど, 全体的にオブジェクトの遮蔽が小さい, とすることができる.

③ 表示領域拡大率: e_3

全認識矩形を囲む最小の四角形 (図4 点線の四角形) の面積が, パイルクラスタ適用のみの可視化結果に比べて何パーセント拡大したかを示す. 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.

④ 表示領域に対する隙間の割合: e_4

表示領域から、全認識矩形の占める領域以外の面積（図4 点線の四角形内で白色の部分）を示す。この値が低いほど、隙間なく画面を有効に使っていると言える。

ウィンドウ

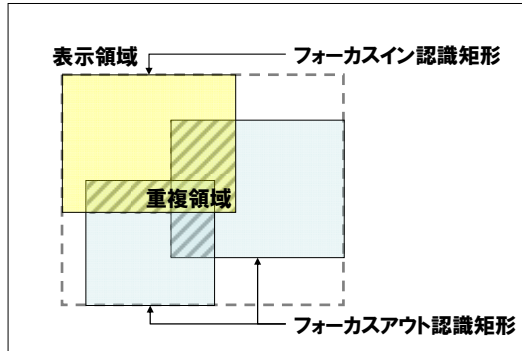


図4. 評価した領域の種類

4.2. 評価結果

数値評価を表1に示す。なお、図5(A)の平安京ビューについては、ジャケット写真の大きさが他の3つのビューと同等になるまで拡大表示した場合の評価を行った。また、クラッタリング回避処理を行う(C)と(D)については、可視化した5つのグループをフォーカスしたときに得られた $e_1 \sim e_4$ の評価結果の平均値を算出した。

まず、 e_1 については、(D)の最小二乗法に重みを考慮した場合の方が、(C)の重みを考慮しない場合よりもクラッタリングを回避した。 e_2 を比較してみると、(B)のパイルクラスタを適用しただけの状態よりも、(C)と(D)のクラッタリング回避処理をした場合の方がクラッタリングは回避できているが、(C)と(D)では(D)の最小二乗法に重みを考慮した場合の方が、全体のクラッタリング回避率は低い。この結果は e_3 とトレードオフの関係に

ある。表示領域の比較では(C)よりも(D)の方が、面積が小さくて済むという結果が得られている。つまり、フォーカスアウト認識矩形におけるクラッタリング発生率が多少高くなっても、フォーカスイン認識矩形について重点的にクラッタリング発生を軽減させ、全体の面積は小さくなるようにする、という目標が達成されたという結果になった。またこの効果によって、若干ではあるが(C)よりも(D)の方が e_4 も小さくなった。

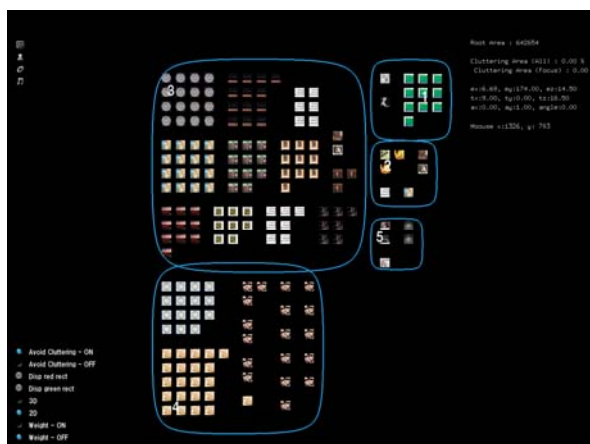
次にクラッタリングが起こらない2次元の平安京ビューと比較してみると、(D)の重みを考慮して面積を抑えた場合では、約4分の1の面積で可視化できることが示された。

以上により、「3次元で本手法によるクラッタリング回避処理をした場合、クラッタリング回避処理をしない場合に比べてクラッタリング率が低く、かつ表示領域は2次元の可視化結果よりも小さい」という望ましい結果が得られた。

4.3. 考察

今回の評価はノード数200、グループ数5の比較的規模の小さいデータ数で行ったため、今後はより規模の大きなデータで評価する必要がある。

また、 e_1 と e_2 については、(C)と(D)のどちらの可視化結果においても、5つのグループのうち3つではクラッタリングが完全に回避されたものの、残り2つのグループをフォーカスしたときにクラッタリング面積が大きくなってしまった。今後この問題については改善を目指す。



(A) 平安京ビュー



(B) パイルクラスタ適用



(C) クラッタリング回避処理適用（重みなし最小二乗法）



(D) クラッタリング回避処理適用（重み付き最小二乗法）

図 5. 可視化結果

表 1. 評価結果

	(A) 平安京ビュー	(B) パイルクラスタ適用	(C) クラッタリング回避処理適用（重みなし最小二乗法）	(D) クラッタリング回避処理適用（重み付き最小二乗法）
① フォーカスイン認識矩形のクラッタリング面積占有率	-	-	20.29 %	14.53 %
② 全認識矩形のクラッタリング面積占有率	0 %	19.15 %	5.03 %	8.40 %
③ 表示領域拡大率	9.38	1.0（基準）	2.57	2.23
④ 表示領域に対する隙間の割合	0 %	0 %	45.59 %	44.50 %

参考文献

[1] Chuah M. C., Roth S. F., Mattis J., Kolojchick J., SDM: Selective Dynamic Manipulation of Visualizations, In Proceedings UIST'95 User Interface Software and Technology, ACM, November 1995.

[2] Elmqvist N., Tudoreanu M. E., Evaluating the Effectiveness of Occlusion Reduction Techniques for 3D Virtual Environments. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2006, pp. 9-18, 2006.

[3] 吉田, 高橋, 西田, 嶋田, 運転経路が遮蔽されないカーナビゲーションシステム. Visual Computing / グラフィクスと CAD 合同シンポジウム, 87-92, 2005.

[4] 宮崎, 伊藤, 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), 113-114, 2007.

[5] 伊藤, 山口, 小山田, 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, 26, 6, 51-61, 2006.

[6] Johnson B., Schneiderman B., Tree-Maps: A Space Filling Approach to the Visualization of Hierarchical Information Space, IEEE Visualization '91, 275-282, 1991.

[7] Bederson B., Schneiderman B., Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies, ACM Transactions on Graphics, 21, 4, 833-854, 2002.

[8] Watanabe N., Washida M., Igarashi T., Bubble Clusters: An Interface for Manipulating Spatial Aggregation of Graphical Objects, ACM User Interface Software and Technology, 37-43, 2006.