

東京大学大学院新領域創成科学研究科

人間環境学専攻

修士論文

WEB3D を用いたマルチメディア情報
の可視化に関する研究

2008 年 1 月 31 日提出

指導教員 岡本 孝司 教授



学生証番号 66812

藤井 公輔

目次

第1章 序論	4
1.1. はじめに.....	4
1.2. 研究目的.....	7
第2章 先行研究	9
2.1. 情報可視化.....	9
2.2. WEB3D.....	20
第3章 3次元表示を用いたグラフ表現	23
3.1. 概要.....	23
3.2. 目的.....	25
3.3. システム.....	26
3.3.1. システムアーキテクチャ.....	26
3.3.2. 機能.....	28
3.4. 考察.....	34
3.5. 課題.....	37
第4章 3次元表示の動画検索への応用	40
4.1. 概要.....	40
4.2. 問題点.....	45
4.3. 目的.....	46
4.4. 設計コンセプト.....	47
4.5. システム.....	51
4.5.1. システムアーキテクチャ.....	51
4.5.2. 機能・特徴.....	53
4.6. 評価.....	62
4.6.1. 評価手法.....	63
4.6.2. 各タスクの狙い.....	65
4.6.3. 自由記述アンケートの狙い.....	67
4.6. 結果・考察.....	67

第5章 結論.....	87
参考文献.....	89
謝辞.....	91

第1章 序論

1.1. はじめに

近年、IT 技術が劇的に進歩したことにより、人々の生活様式が変化を見せてきている。音楽、写真、ビデオといった情報が PC 内に蓄積されるようになり、CD、ネガ、ビデオテープといった記憶媒体を利用することが少なくなった。また従来、そういった情報は個人的に保持してきたが、最近ではインターネット上にアップロードすることにより、みなが共有する傾向が強まってきている。

膨大かつ多様な情報が生成、蓄積すると同時に、情報を利用しやすくするにはどうしたらよいか、という問題も生じてきた。蓄積された情報量に比例して、その中から必要なものだけを取り出すことは難しくなる。例えば、過去に受信したメールをチェックしたい時や、あるファイルの保管場所を忘れてしまった時などがそうである。また、インターネット上の情報に関して言えば、オンラインショップなどで無数にある商品の中から目的に沿う商品を探し出す時などがそうである。

複雑で大量の情報を分かりやすい形で提示するための手法として、「情報可視化」という技術が発達してきた。情報可視化とは、ある情報に対してユーザーが何らかの洞察を行うことができる視覚的なユーザーインターフェースである。わかりやすく言えば、大量の情報を効果的に計算機画面に表示することにより、ユーザーが情報を理解したり操作したりするのを助けることができるしくみのことである¹⁾。

情報可視化が実際に用いられる例をあげてみる。図 1.1 はアメリカの国勢調査における郡ごとの調査結果データであり、図 1.2 は図 1.1 のデータ中の学歴と収入の 2 要素を平面にプロットした図である。つまり図 1.2 は図 1.1 を視覚化したものであるが、これにより誰もが簡単に学歴と収入間の相関を発見することができる。こういった洞察はデータそのものが

あらかじめ保持しているものではなく、視覚的なパターン認識により推察されるものである。

	A	B	C	D	E	F	G	H
1	Name	State	Population	% College Grad	Income per Capita	Median Rent	Employees	Retail Sales
2	Aleutians East	AK	2,305	12.9	20,114	518	1,410	6,693
3	Aleutians West	AK	5,259	14.8	20,298	428	4,321	24,233
4	Anchorage	AK	251,335	26.9	26,619	528	93,037	2,612,640
5	Bethel	AK	15,525	13.1	15,493	460	2,269	59,687
6	Bristol Bay	AK	1,023	18.9	29,699	464	392	9,653
7	Dillingham	AK	4,360	15.3	22,616	518	1,060	23,077
8	Fairbanks North Star	AK	83,374	25.2	19,149	471	17,746	702,135
9	Haines	AK	2,181	17.6	26,413	405	371	15,154
10	Juneau	AK	29,378	30.7	26,066	587	8,369	289,620
11	Kenai Peninsula	AK	46,151	17.9	22,761	410	8,934	329,438
12	Ketchikan Gateway	AK	14,422	20.2	28,789	533	5,583	152,834
13	Kodiak Island	AK	14,987	21.5	28,888	609	4,234	94,172

図 1.1 アメリカ国勢調査データ

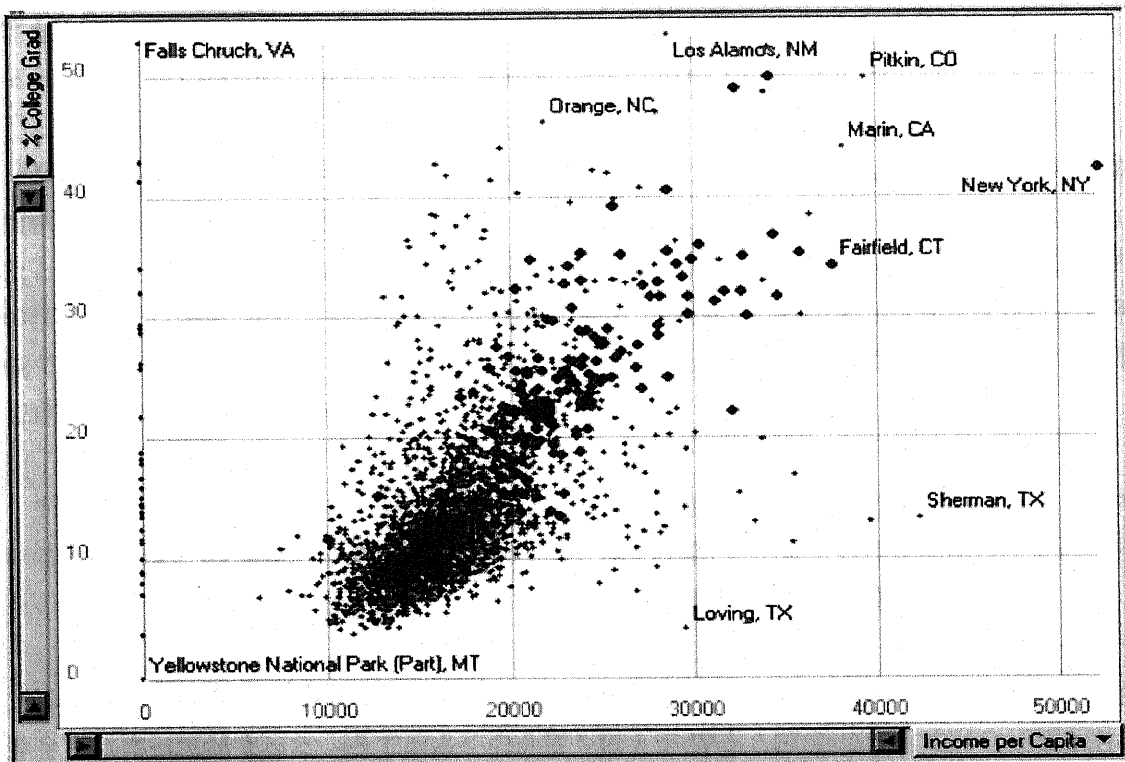


図 1.2 国勢調査データにおける学歴と収入をプロットした図

この例は非常にシンプルなものであるが、データ形式が複雑になったり、データ量が膨大になるほどその視覚化方法をうまく設計することが重要になってくる。様々なデータ形式に対応した可視化手法の例を第2章で紹介する。

このように情報を可視化することによってユーザーが予期しない新たな発見を促すことができる。今後も増え続けるであろう多種多様な情報をより効率よく管理するための手法

として情報可視化のその可能性が注目されている。

また、インターネットの発展に伴い最近では「Web2.0」と呼ばれる新たなトレンドが生まれている。この言葉自体は2004年にTim O'Reillyによって提唱された言葉である²⁾が、特定の技術やコンセプトを指している言葉ではなく、「次世代のWeb」を漠然と総称する言葉である。そのため使う人によって多少認識が異なるところはあるが、多くの人が合意するその最も大きな特徴の一つは、コンピューターにおけるOSのようにWebが一種のプラットフォーム（基盤）として振舞うようになり、その上で情報や機能が製作者の手を離れて組み合わせられたり加工されたりするという点である。

従来のWebは製作者が作った状態で完結しており、利用者は単にそれを利用するだけの関係であったが、Web2.0ではWebサイトのもつ情報や機能を外部のサイトやソフトウェアなどから参照したり呼び出したりすることができる。それを実現するためのインターフェースとなっているのがAPI(Application Programming Interface)である。

具体的にWeb上ではどのようなAPIが公開されているのかの例をあげる。Web APIが提供する機能として最も多いのが、企業のもつデータベースを公開するというものである。たとえば、米Googleの「Google Maps API」では、同社の地図データベース「Google Maps³⁾」の情報を別のWebサイトに取り込んで表示したり、地図上に目印をつけたりすることが可能である。米Amazonの「Amazon E-Commerce Service⁴⁾」のように、商品データを参照できるデータベースもある。

データベースだけでなく、プログラムの機能を提供するタイプのWeb APIも提供されている。例えば、「Simple API⁵⁾」というサイトでは、URLを指定することによってそのサイトのサムネイル画像を自動生成してくれる機能を提供するAPIを公開している。

Web APIの公開は2002年ごろ、GoogleやAmazonなどの先進的なインターネット関連企業によって始まった。今では多くの企業や個人がWeb APIを公開しており、しかもその大半は無償で使うことができる。なぜ企業は無償でAPIを公開するのであろうか。一つは、APIを公開することによってより優れたサービスが生まれる可能性が高まる、ということである。企業外の一般開発者がそのAPIを利用することによって、当企業が思いつかなかったようなユニークなサービスを生んでくれる可能性がある。また、APIを利用してもらうことで、間接的にせよ自社のサービスのユーザーが増える、というメリットもある。

このようなWeb APIの充実とともに、複数のWeb APIを組み合わせ、あたかも1つのWebサービスにする開発手法である「マッシュアップ」という言葉も生まれた。既存のWeb

サービスを組み合わせて、短期間でITの深い知識がなくても、アプリケーション開発を行うことができることから、新しい開発手法として注目されている。事例としては、Google MapsのAPIによる地図情報と、HotpepperのAPIによる飲食店情報などを組み合わせたWebサービスが代表的である⁶⁾。図1.3にその例を示す。



図 1.3 マッシュアップ事例

マッシュアップに関しては様々なコンテストも開かれ始め、その作品の中には情報の関係性や構造を視覚的インターフェースで表現するものもあり、情報可視化手法がWebの世界にも浸透し始めたことを予感させられる。今後インターネット上に蓄積されるデータベースはますます増大していくであろうが、それらをより効率よく、かつ、楽しく魅せるしくみを発明するのは我々一般開発者なのかもしれない。

1.2. 研究目的

インターネットの発展に伴い今後ますますWeb上の情報は多様化し、効果的な情報提示のあり方が求められることは間違いない。3次元表示はその1つの可能性であり、有効性

や課題を明らかにすることは重要なことである。また Web 上であることから誰もがアクセスして簡単に扱えるインターフェースであることが求められている。

1. 1でも述べたように情報可視化は近年注目を集め、様々な研究がなされてきたわけであるが、さらなる改善が求められている。特に様々な属性値をもつ多次元の情報に関しては、2次元平面での可視化よりも3次元空間での可視化の方が多くの軸を設定でき、より多様な属性を表現できると考えられる。

そこで、本研究では情報を可視化するにあたり3次元表示を利用し、多属性を持つ情報、特にネットワーク上から利用できるAPIを使用することで得られる、XML形式の情報の効果的な表示・洞察が可能な可視化システムを開発した。また、このシステムを用いて評価実験を行うことでXML情報の理解を促すユーザ支援ツールとしての3次元表示のあり方について考察を行う。

第3章では、iTuneにより書き出されるXML形式のメタデータを元に、音楽データベースを3次元空間で可視化するシステムを実装し、考察を行った。その際、色のグラデーションを用いることで3次元空間の位置把握を容易にした。

第4章では、YouTubeのAPIを利用し、Web上のデータベースの可視化を試みた。動画は音楽以上に多様な情報が混在するマルチメディアであり、効果的な可視化手法が期待されている分野である。第3章で得られた知見を生かし、効果的と思われる可視化システムを開発し、またユーザー検証による評価も行い考察を加えた。

第2章 先行研究

本章では、情報可視化における先行研究の紹介を行う。

2.1. 情報可視化

序論でも述べたように情報可視化(Information Visualization)とは、抽象的なデータに対してコンピューターの力を借りてインタラクティブでビジュアルな表現を行うことによって、人間がそのデータに対して何らかの洞察や理解を加えることができる手法や概念のことである。この情報可視化という言葉が使われ始めたのは比較的新しく、本来、可視化と言えば、シミュレーションによる科学技術計算の結果を画面に表示するサイエンティフィックビジュアリゼーション(Scientific Visualization)のことを指すのが一般的であった。

しかし、Webの発展や、デジカメ・iPodに代表されるような記憶装置を内蔵したメディアの普及などにより個人でも大量のデータを扱う機会が増え、また、パーソナルコンピューター(PC)の高機能化・低価格化に伴い個人でもそういった大量のデータを処理・表示することが可能となったため、情報可視化技術に対する需要が高まっている。

サイエンティフィックビジュアリゼーションでは視覚化の対象が物理現象であることが多いため、物理的制約により可視化手法がある程度決まってしまう場合が多い。つまり、実際に存在する物体や現象を再現することが目的であるため、可視化結果は実世界のある物や現象に限りなく近いものを目指せばよい。一方、情報可視化の場合ははっきりした形をもたない抽象的なデータを視覚化の対象とするため、表示手法は定まっておらず、その情報の形式やデータ量などに応じて可視化手法を吟味する必要がある。

そこでまず、情報そのものがどういった構造を持ちうるのかということ把握しておくことが重要である。以下、表 2.1 に情報の形式をおおまかにまとめる。

表 2.1 情報構造の分類表

情報の構造	例・解説
線形	アルファベットオーダーのリスト、プログラムのソースコード、時系列データ（プロジェクト管理データ、等）
階層	ファイルシステム、家系図、組織図
ネットワーク	コミュニティ構造、物流、Web ページのリンク構造
多次元	飲食店データ（店名、価格、距離、ジャンル、等） メタデータ（タイトル、作成者、作成日時、サイズ、等）

次に、それぞれの情報構造に応じた情報可視化手法についての先行研究を紹介する。

● 線形情報の可視化手法

◇ Perspective Wall⁷⁾

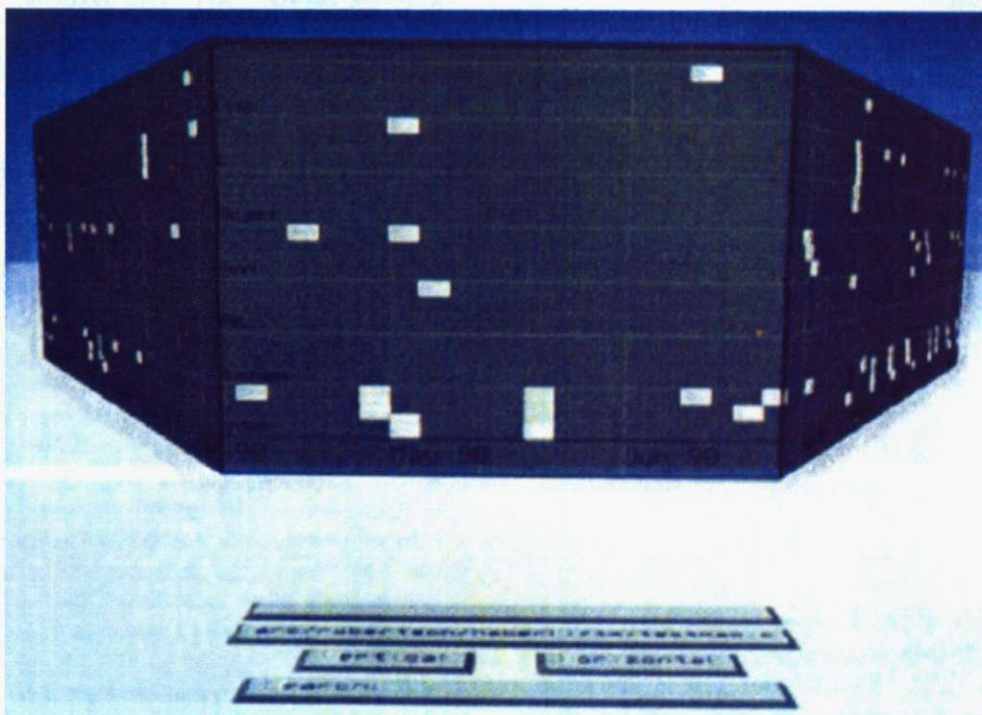


図 2.1 Perspective Wall

J. D. Mackinlay, G. G. Robertson, S. K. Card (1991)

Perspective Wall は、プロジェクト管理データに代表される線形データを3次元の壁に表示するシステムである。この3次元の壁は途中2箇所折れ曲がり、中央部分は計算機画面と平行で左右部分はそれぞれ端が奥行き方向に遠ざかっている。この壁に時系列データを表示すると、中央の壁に表示されるデータはその詳細を見ることができ、一方、左右の壁に表示されるデータはその存在だけを把握することができる。従来の2次元可視化で大規模な時系列データを見る場合、着目点近傍を詳細に見ようとすると全体構造が見えなくなり、逆に、全体を見ようとすると個々の詳細が見えなくなる。これに対し Perspective Wall は、3次元グラフィックスの透視投影図法(Perspective View)を利用することによって、局所的詳細と大局的概略を統合した表示を可能としている。

◇ SeeSoft⁸⁾

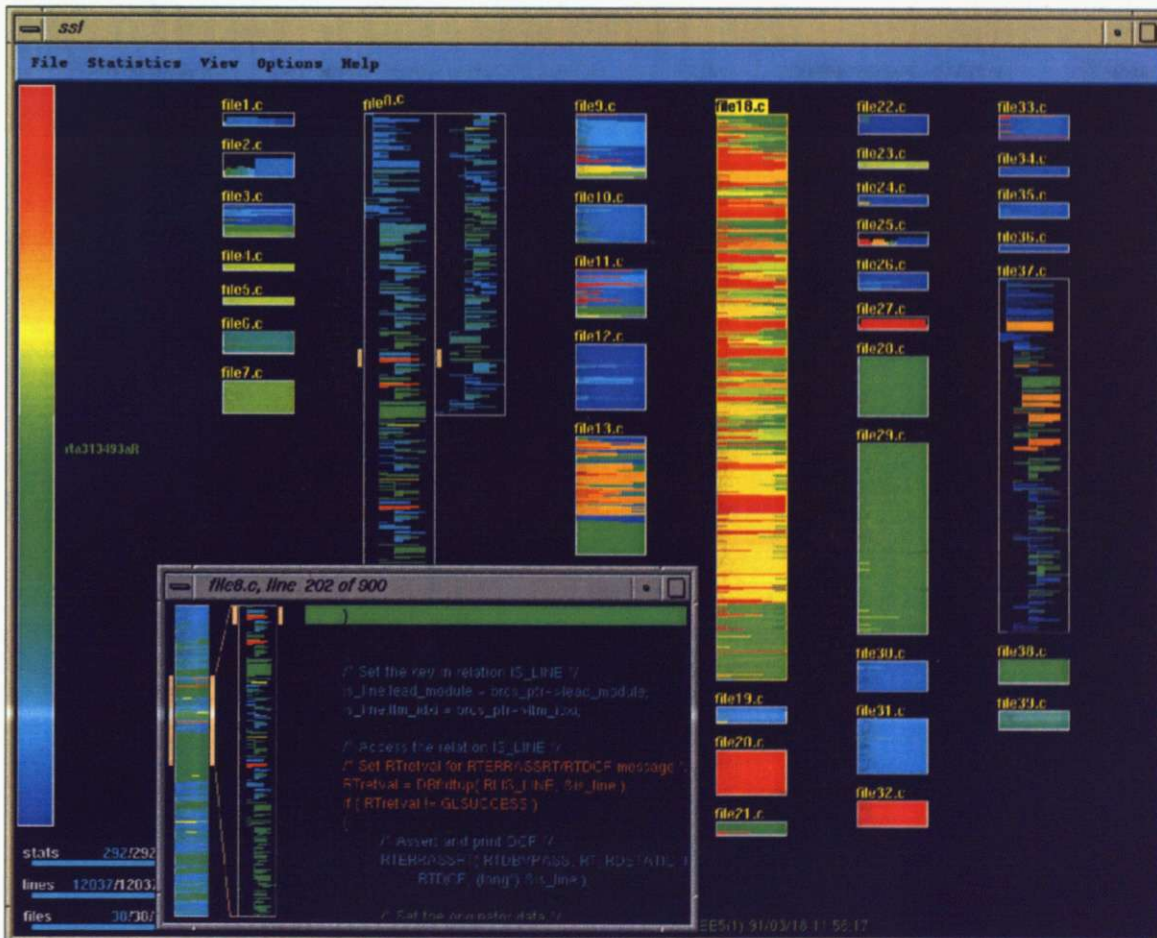


図 2.2 Seesoft

J. L. Steffen, S. G. Eick, E. E. Sumner Jr. (1992)

AT&T で開発された Seesoft は、プログラムに関する全体情報の把握を目的としたシステムで、プログラムの各行は色の付いた 1 本の線として表示される。この色はコードの変更履歴といった統計データを表し、ユーザーはプログラムの概略を把握するとともに、プログラムの履歴情報を獲得することができる。図 2.2 はあるディレクトリ内に含まれるソースコードファイルを可視化している。また、この図では赤が最も新しいコード、青が最も古いコードを表している。

● 階層情報の可視化手法

◇ Cone Trees⁹⁾

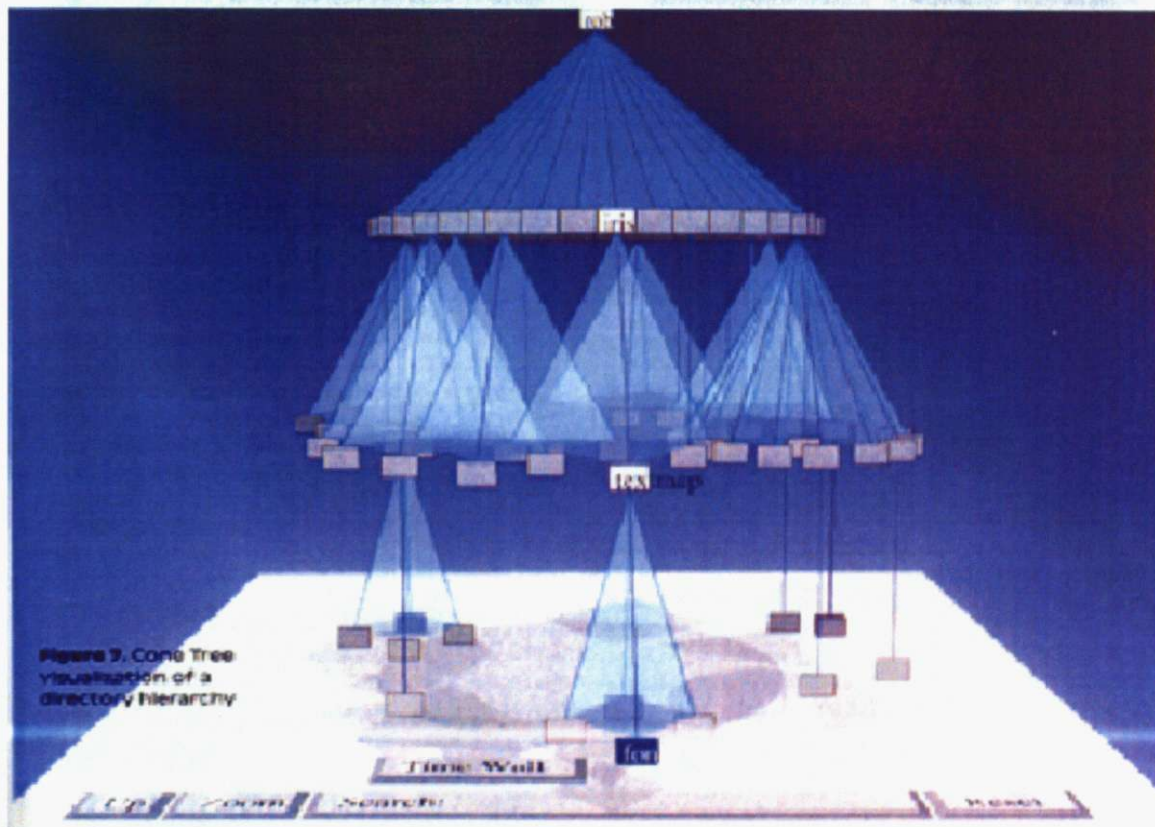


図 2.3 Cone Trees

G. G. Robertson, J. D. Mackinlay, and S. K. Card (1991)

Cone Trees は Xerox PARC が開発したシステムで、ファイルシステムに代表される階層データを 3次元の木として表示する。木の各レベルにおいて子ノードは親ノードを頂点とする円錐の底面円周上に配置される。2次元表示では大規模な木は用意に表示領域から溢れてしまうが、Cone Trees は 3次元の奥行き方向を利用することによって、効果的により大きな階層データを画面溢れを起こすことなく表示することができる。

同時に Cone Trees はアニメーションによってユーザーの認知負荷の低減を実現している。ユーザーが任意のノードをマウスで選択すると、ルートからそのノードへのパス上に存在する全ノードが最前面に一直線に並ぶように、各円錐は平行してスムーズに回転する。また、表示ノード数が増えた場合には、マウス操作によって着目している部分木の円錐だけを表示させたりすることもできる。

◇ TreeMaps¹⁰⁾

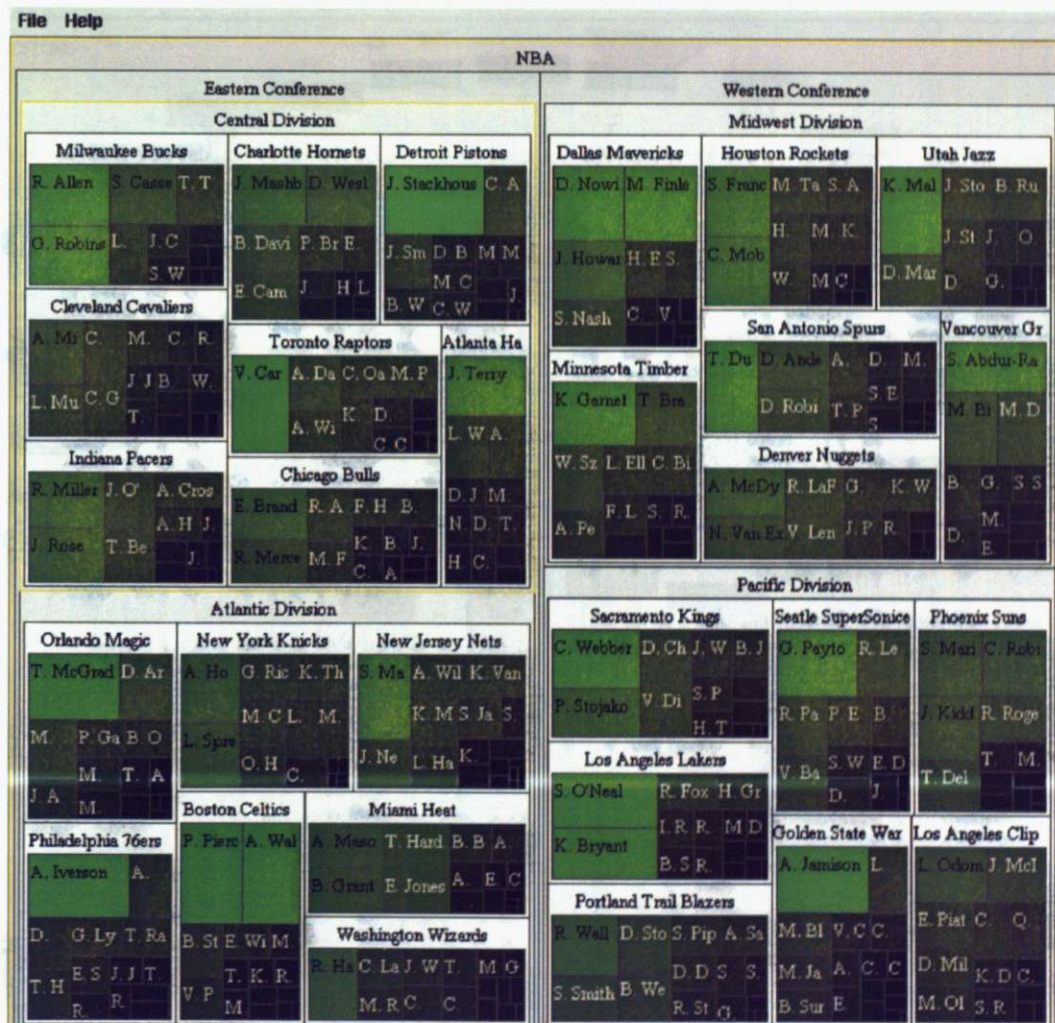


図 2.4 TreeMaps

Brian Johnson and Ben Shneiderman (1991)

TreeMaps は、深い階層のあるツリー構造のデータを縦／横に分割を繰り返すことにより 2次元平面上で可視化するシステムである。大きな長方形に包含される小さな長方形という形で階層構造の親子関係を表現している。各長方形の形は TreeMaps 内のアルゴリズムによって、最も空間効率がよくなるように決定される。現在この TreeMaps を応用したソフトウェアが多数存在する。

図 2.4 では NBA の選手をリーグ、地区、チーム、選手名という階層情報に基づいて TreeMaps で可視化したものである。

◇ Hyperbolic Browser¹¹⁾

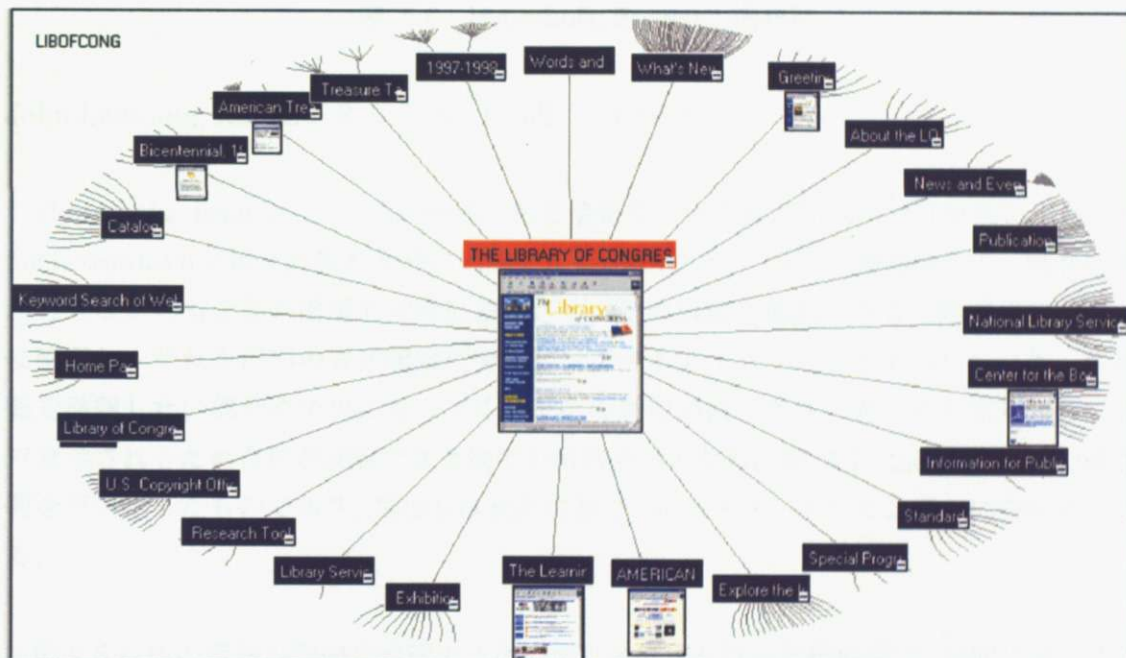


図 2.5 Hyperbolic Browser 初期状態

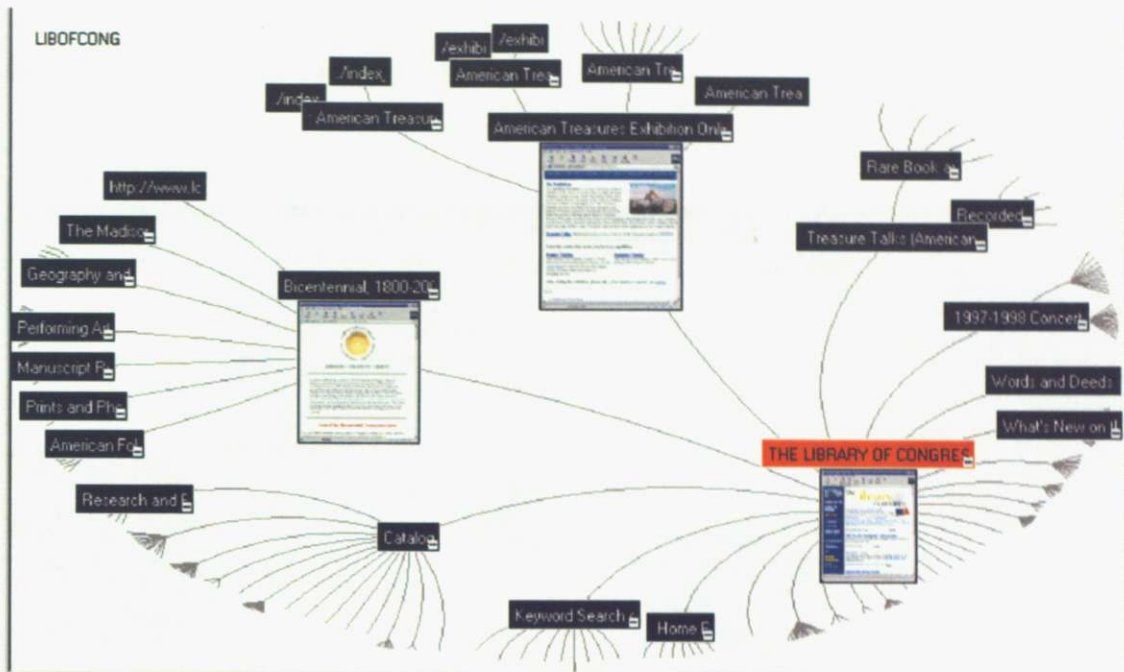


図 2.6 Hyperbolic Browser 操作後

John Lamping, Ramana Rao, Peter Pirolli (1995)

Hyperbolic browser は、深い階層のある情報を円形の表示領域内で可視化したもので、focus+context(全体と詳細を同時に確認できる)技法を用いている。階層構造の一部を拡大表示させながらも全体の階層も同時に表示させ、その概略を確認できる。階層情報を双曲面上に配置し、それを円形の表示領域にマッピングするアルゴリズムを使用している。詳細情報を確認したい部分を中央にもってくるとその周辺が拡大され、逆にその周外部分は歪んで表示されるため存在が確認できる程度の情報表示となる。図 2.5, 2.6 は図書館の蔵書情報を可視化したものであり、図 2.5 は初期状態で、ルートノードがセンターに表示されている。

図 2.6 は他の情報の詳細を確認するためマウス操作を行った後の図で、初期状態ではセンターに位置していたルートノードは円周近くに位置し、アイコンも小さく表示されていることが確認できる。

● ネットワーク情報の可視化手法

◇ mixiGraph¹²⁾

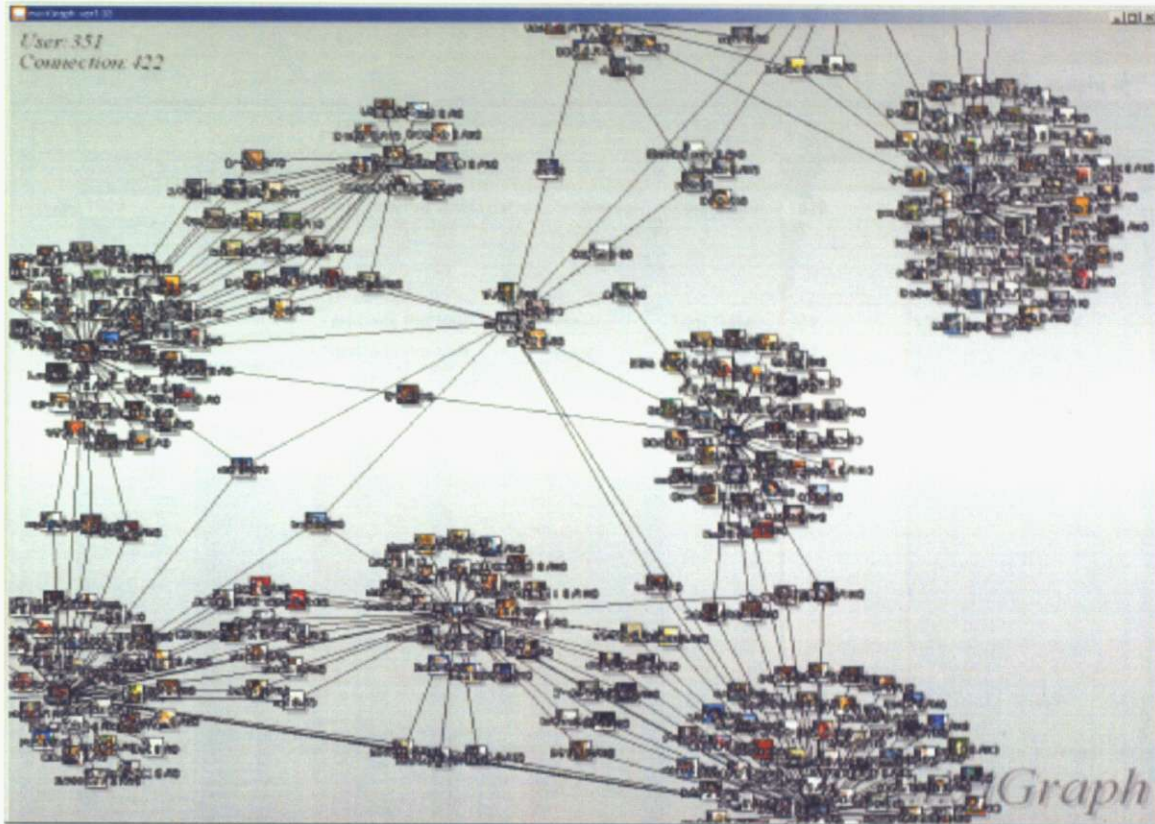


図 2.7 mixiGraph

Koji Sugimoto (2005)

mixiGraph はソーシャルコミュニティ Web サイト”mixi”での交友関係を表す”マイミクシー”登録関係を図示するシステムである。mixi ユーザーは、専用の Web メールを介してほかのユーザーとメッセージをやりとりでき、相手の了承を得ればそのユーザーを”マイミクシー”へと追加することができる。mixiGraph を起動したユーザーの写真が中央に、起動ユーザーの”マイミクシー一覧”に登録されている友人・知人の写真が周囲に表示され、起動ユーザーの写真と友人・知人の写真との間に関係線が引かれる。友人・知人の写真をダブルクリックすると、その人のマイミクシー一覧が追加表示され、関係のある人同士が線で結ばれていく。友人・知人の縁をたどっていくことで、思わぬ人とのつながりを図や写真を通じて発見することができる。

● 多次元情報の可視化手法

◇ Table Lens¹³⁾

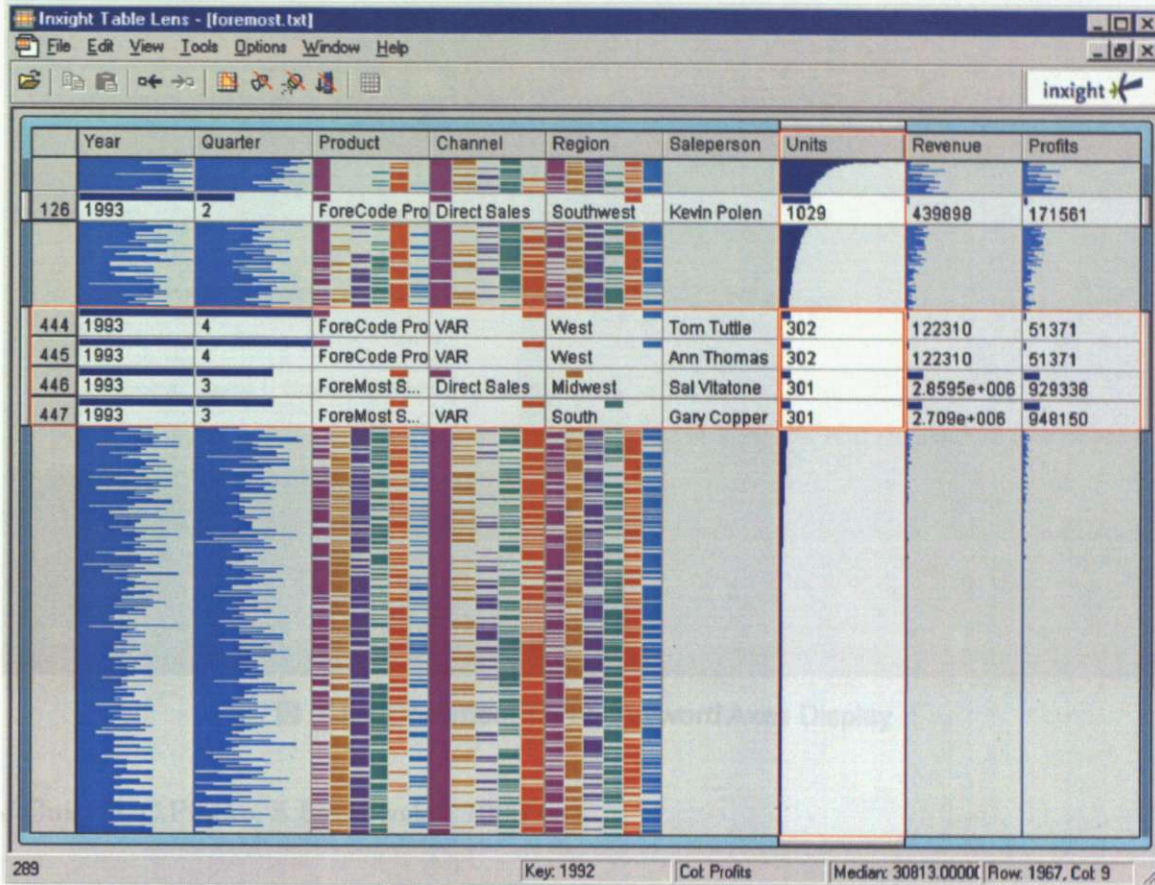


図 2.8 Table Lens

Ramana Rao, Stuart K. Card (1994)

Table Lens は focus+context(全体と詳細を同時に確認できる)技法を用いており、全体中の位置関係を認識しつつ、注目した情報の詳細表示を可能にしたシステムである。通常多次元のデータはテーブル形式のデータとして管理されることが多いが、非常に大きな表を小さな画面に表示するときには、目的の情報を探すのに煩雑なスクロール操作を繰り返さなければならない。Table Lens では数値データとして表示しきれないものは線の長さとして表現したり、項目名を表示しきれないものは色や点の位置で表現したりしている。表中の行や列を拡大すると数値や名前が見えるようになる。

◇ Document Three-Keyword Axes Display¹⁴⁾

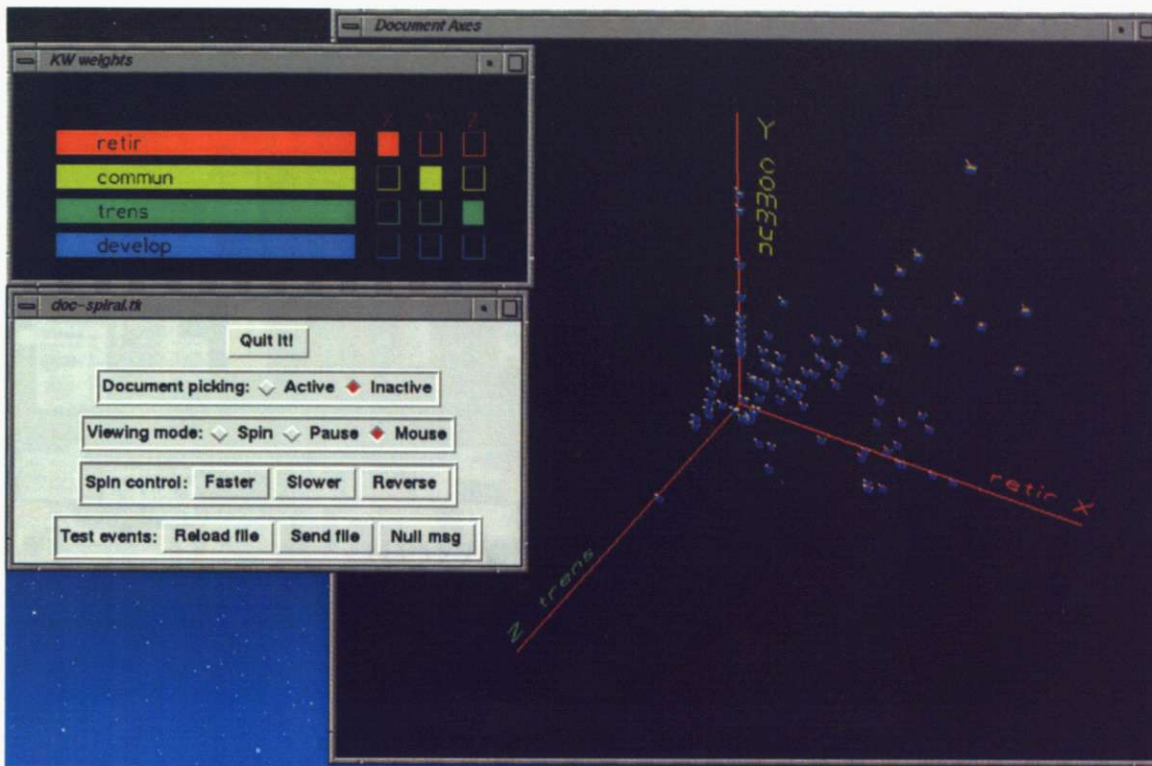


図 2.9 Document Three-Keyword Axes Display

J.Cuigini, C.Piatko, S.Laskowski (1996)

Document Three-Keyword Axes Display は文献の検索結果を 3 次元空間内に表示したシステムである。特にこのシステムでは 3 つ以上のキーワードを指定して検索することを前提としており、各キーワードとの一致度をそれぞれ X,Y,Z 軸にとり、3 次元空間でプロットしている。その際、結果として返ってくる文献をアイコンで表示しており、また、ユーザーは各軸にどのキーワードを割り当てるかを別ウィンドウのインターフェースで指示することができる。

ユーザーが複数のキーワードを指定する際、各キーワードの重要度には偏りがあることが多く、あるキーワードに重点をおいて検索したい時などには効果を発揮する。しかし、平面ディスプレイに仮想 3 次元空間を実装しているため、それぞれのアイコンが 3 軸のどの位置に対応しているのかがわかりづらいといった問題点も存在する。

第 3 章では、こういった問題点も踏まえ、3 軸で多次元の情報を可視化したシステムについての説明を加える。

● その他の可視化手法

◇ Data Mountain¹⁵⁾



図 2.10 Data Mountain

G.Robertson, M.Czerwinski, K.larson, D.Robbins, D.Thiel, M.van Dantzich (1998)

Data Mountain は Web ページのお気に入り (ブックマーク) 情報を可視化したシステムである。ブックマーク情報は Web ページのタイトルと URL のみを保持するもので、データ量を問わずデータ間に関連性は見られない。表 1 で示した情報構造のいずれにも当てはまらない情報である。

Data Mountain では各ブックマークを Web ページのサムネイル画像で表し、そのサムネイル画像を傾いた平面状にユーザーが自由に配置することができる。図 2.10 における 2 枚

の図はいずれも同じ Web ページ 100 件分のブックマークを、異なる 2 人のユーザーが可視化した例である。この例からもわかるように 2 人の配置は全く異なっており、その配置方法はユーザーの好みに委ねられる。これは実際に机や本棚で書類を管理するメタファーを用いており、ユーザーの空間記憶能力を利用した情報管理システムといえる。

2.2. Web3D

情報可視化の先行研究でも見られるように、可視化手法として 3 次元表示を利用したものが増えてきている。しかし、それらのシステムのほとんどはローカルな環境下で動作するものである。これに対し、最近では、インターネット上で 3D オブジェクトをインタラクティブに操作することができる Web3D という 3D 技術が普及しつつある。

Web3D では Web ブラウザ上に表示された 3D オブジェクトを自由に回転・拡大・縮小させたり、アニメーションさせたりすることができるため、平面的な情報しか持たない画像に比べると、マウス操作によりあらゆる角度から見ることで Web3D コンテンツは非常に豊かな表現が可能となる。現在その応用例として最も多いのが製品紹介である。

図 2.11 はスバルの Web サイト¹⁶⁾であり、車の製品紹介に Web3D を適応した事例である。ユーザーは 3D の車オブジェクトを自由に回転・拡大・縮小させたり、ボディの色、車種を変更することができ、直感的にどういう製品かを現実とほぼ同程度に理解することができる。車の他にも不動産サイトでは、実際の物件を 3DCG で再現し、それを Web3D でインターネット利用可能としたものもある。これによりユーザーはその物件の間取りや部屋の内部の様子を実際に足を運ぶことなく理解することができる。このように、実際に 3D であるモノに対して Web3D を用いる例がほとんどであるが、そのようなコンテンツを作り上げるには、3DCG 作成ソフトでのモデリング技術や、インタラクティブ性を実現するためのプログラムスキル等、開発側の負荷が通常のサイトに比べ飛躍的に増大してしまうため、普及率は今一歩伸び悩んでいるのが現状である。



図 2.11 製品紹介におけるWeb3Dの適応例

それに対し、図 2.12 は日興コーディアル証券の Web サイト¹⁷⁾で、株価情報の表示に Web3D を適応した例である。形のない株価という情報に対して Web3D を用いて 3 次元表示を可能にしており、その点で Web3D と情報可視化とを融合させた比較的新しい試みと言える。このシステムでは、1つの株価情報は1つのパネル上に表示され、そのパネルが前後左右に配置されている。正面のパネルにてある株価の詳細情報を確認すると同時に、その周辺にある情報もぼんやりと認識できるようになっており、情報可視化ではしばしば用いられる、focus+context の技法を利用している。また、画面下のボタンインターフェースを操作することによって、各パネルはスムーズなアニメーションで移動する。

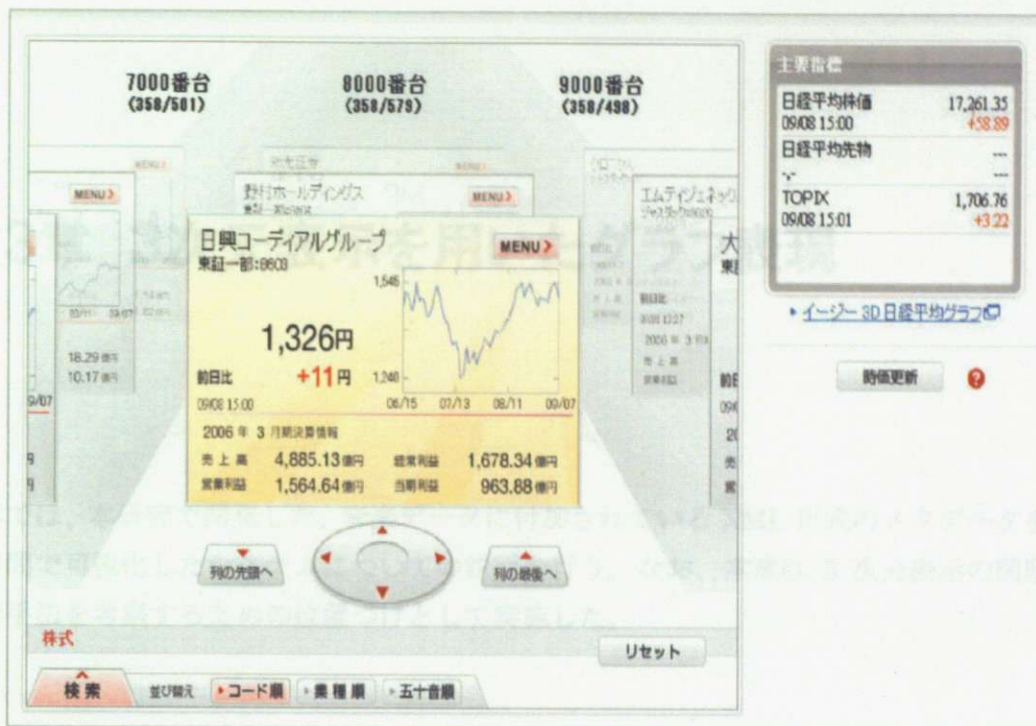


図 2.12 3D 証券ビューアー

このように、Web3D を用いることによってその表現手段の拡大により、斬新かつ魅力的なサイトを作れる可能性は広がる、しかし、上述した開発コストというデメリットの他にも普及を妨げている要因がいくつかある。まず、Web3D に関する企画が乱立している状態であることがあげられる。主なもので30種類近くあり、国内で利用できるものだけでも数10種類を数える。それぞれ表現力や実現できる機能が異なり、適材適所で使うソフトを選択しなければならない。早急な企画の統合が期待されている。また、もうひとつの大きな問題点として考えられるのが、Web3D オブジェクトを表示させるためには専用のプラグインのインストール作業が必要なことである。特に、あまり PC の利用頻度が高くないユーザーにとっては非常にハードルの高い作業であり、好んでわざわざ新しい機能を利用しようとはしない。中にはプラグイン不要のものもあるが、その分開発コストは急激に増大する。

今後 Web3D が発展していくためには、こういった問題点を克服していくとともに、単に商品を回転させるというような安易な発想を乗り越え、より創意工夫に富んだ発想力が求められる。そのためにも、多くのクリエイターが実験的で多彩なコンテンツを自由に公開しあったり、プロデューサーや経営者がそういった新たなコンテンツを積極的に発掘し、ビジネスに結び付けていくことが求められるだろう。

第3章 3次元表示を用いたグラフ表現

本章では、本研究で開発した、音楽データに付加されている XML 形式のメタデータを 3次元空間で可視化したシステムについての説明を行う。なお、本章は 3次元表示の問題点や改善手法を考察するための位置づけとして実施した。

3.1. 概要

現在、PC 上で音楽データを管理・再生するのに最も利用されているソフトとして iTunes¹⁸⁾があげられる。図 3.1 はその使用例である。音楽データには wave 形式の音楽データそのもののほかに、そのデータを説明するメタデータと呼ばれるデータが付加されている。音楽データにおけるメタデータとしては、アーティスト名、トラック名、アルバム名、ジャンル、再生時間、リリース年、等があり、これらは通常下図のようなテーブル形式のデータで管理されている。



図 3.1 iTunes による音楽ファイル表示例

しかし、このような表を眺めるだけではデータ間に存在する規則や相関、つまり、ユーザーの嗜好性や特徴といったものまでは発見することが難しく、そのようなものを発見するためには情報を視覚化する必要性がある。メタデータのような多属性の値をもつ多次元データの可視化に関する研究としては第2章で述べた、Table Lens や Document Three-Keyword Axes Display などがある。Table Lens は2次元平面での可視化システムであり、Document Three-Keyword Axes Display は3次元空間の可視化システムである。

多次元情報の可視化手法としては、1つの情報が多くの属性値をもつことから従来の2次元平面へのマッピング手法よりも、より多くの属性を表現できる3次元空間へのマッピングの方が効果的と考えられるが、Document Three-Keyword Axes Display ではマッピングされたデータがXYZ軸のそれぞれどの位置に対応しているのかがわかりづらいといった問題点があった。

また、Web上でグラフ表現を可能とするツールとして OpenOffice¹⁹⁾や Google Docs and Spreadsheets²⁰⁾などがあるが、いずれも2次元グラフィックスにしか対応しておらず、3次元でのインタラクティブなグラフ表現は実現していない。

本章では、多次元情報(XML)を3次元表示で可視化する手法において上記の問題点を解決

すべく、音楽に付加されたメタデータを、3次元空間内のXYZ軸でグラフ表現できるツールの開発を行った。その際、Web上での閲覧を可能とするため、Web3D技術の一つであるViewpointを用いて実装した。なお、用いるメタデータに関しては、iTuneが楽曲管理のために書き出すXML形式の外部ファイルを利用した。その生データの一部を図3.2に示す。

```
<dict>
  <key>Track ID</key><integer>1664</integer>
  <key>Name</key><string>Chubb_Rock_-_Treat_Em_Right-XXL</string>
  <key>Artist</key><string>07</string>
  <key>Album</key><string>Funkymix 07</string>
  <key>Genre</key><string>Dance</string>
  <key>Kind</key><string>MPEG オーディオファイル</string>
  <key>Size</key><integer>8837508</integer>
  <key>Total Time</key><integer>368222</integer>
  <key>Year</key><integer>1986</integer>
  <key>BPM</key><integer>116</integer>
  <key>Date Modified</key><date>2007-07-09T20:38:50Z</date>
  <key>Date Added</key><date>2007-07-09T17:20:58Z</date>
  <key>Bit Rate</key><integer>192</integer>
  <key>Sample Rate</key><integer>44100</integer>
  <key>Comments</key><string>.: Tastyt Team :.</string>
  <key>Persistent ID</key><string>55218F01A2AC61A8</string>
  <key>Track Type</key><string>File</string>
  <key>Location</key><string>
file:///localhost/C:/Documents%20and%20Settings/fujii/M/%20Music/iTunes/iTunes%20Music/07/Funkymix%2007/
Chubb_Rock_-_Treat_Em_Right-XXL.mp3</string>
  <key>File Folder Count</key><integer>4</integer>
  <key>Library Folder Count</key><integer>1</integer>
</dict>
<key>1665</key>
<dict>
  <key>Track ID</key><integer>1665</integer>
  <key>Name</key><string>Dark Circles</string>
  <key>Artist</key><string>07 Leacy &#38; Urban Fox</string>
  <key>Album</key><string>boty 2000</string>
  <key>Kind</key><string>MPEG オーディオファイル</string>
  <key>Size</key><integer>4821449</integer>
  <key>Total Time</key><integer>301322</integer>
  <key>Year</key><integer>2000</integer>
  <key>BPM</key><integer>128</integer>
  <key>Date Modified</key><date>2007-07-09T20:00:05Z</date>
  <key>Date Added</key><date>2007-07-09T17:20:58Z</date>
  <key>Bit Rate</key><integer>192</integer>
  <key>Sample Rate</key><integer>44100</integer>
  <key>Comments</key><string>.: Tastyt Team :.</string>
  <key>Persistent ID</key><string>55218F01A2AC61A8</string>
  <key>Track Type</key><string>File</string>
  <key>Location</key><string>
file:///localhost/C:/Documents%20and%20Settings/fujii/M/%20Music/iTunes/iTunes%20Music/07/Funkymix%2007/
Dark_Circles.mp3</string>
  <key>File Folder Count</key><integer>4</integer>
  <key>Library Folder Count</key><integer>1</integer>
</dict>
</dict>
```

図 3.2 iTunes による XML データ

3.2. 目的

多属性の情報を保有する XML 形式のデータを3次元のグラフ表現で可視化するにあたり、過去の研究の問題点を解決しうるシステムを実装するとともに、XML情報の可視化に3次元表示がどのように有効かを調査する。

本章で解説するシステムについては、多次元情報を3次元表示で可視化する有効性に焦点を絞って開発を行っており、第4章で用いるようなWeb APIは使用していない。基本的には個人の音楽データベースをiTunesの機能によってXMLファイルで書き出し、それらをローカルな環境で処理した上で可視化を行った。第4章では、本章で得られた知見を生かし、Web APIを用いたシステムの開発を行う。

3.3. システム

3.3.1. システムアーキテクチャ

本システムは、HTML ファイル、MTX ファイル（ブラウザに表示する 3D 情報を保持するファイル）によって構成され、可視化には Web3D の 1 つである Viewpoint²¹⁾を用いた。MTX ファイルは XML 形式の 3 次元オブジェクトデータを保持しており、Viewpoint の機能により HTML ファイルから MTX ファイルを呼び出すことができ、その結果、3 次元オブジェクトがブラウザに表示される。

MTX ファイルは音楽データから欲しい情報だけを抽出して作成した CSV ファイルをもとに作成している。本システムでは、XML 形式の生データから音楽の特徴を表すと考えられる、アーティスト名（文字列）、リリース年（数値）、BPM（数値）、ジャンル（文字列）の 4 つのパラメータを抽出した。MTX ファイルには、ジャンルを除く 3 つのパラメータを XYZ 軸に割り当て、3 次元空間にプロットした情報が保持されている。HTML ファイルからこの MTX ファイルを呼び出すことで、図 3.3 に示すような可視化結果が Web ブラウザ上に表示される。ユーザーはマウス操作によって可視化結果を自由に回転・拡大・縮小することができる。

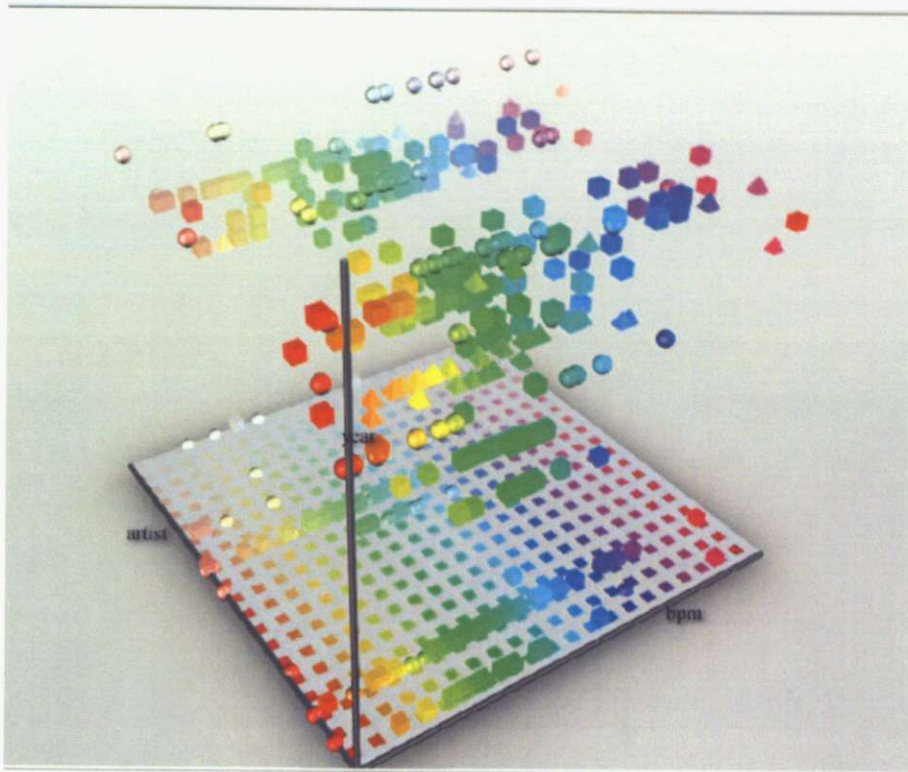


図 3.3 音楽データの可視化例

図 3.3 に示すような可視化結果を得るまでに行ったユーザーPC 内の処理のフローを図 3.4 に示す。以下、フローチャートの詳細説明を述べる。

- (1) iTunes が自動的に書き出す XML 形式の音楽データベースを取得
- (2) 図 3.3 に示すような XML 形式のデータから、アーティスト名、リリース年、BPM、ジャンルの 4 つのパラメータを抽出し CSV 形式のファイルとして書き出した。その際、XML を処理するプログラムは Perl で書いた。
- (3) 得られた CSV ファイルを元に 3D 情報を保持した MTX 形式の可視化用ファイルを作成。その際用いたプログラム言語は Perl で、CSV ファイルを読み込み、それぞれのパラメータに格納された値に応じて 3 次元空間内の座標を決定するプログラムを実行した。
- (4) 作成された MTX ファイルを HTML ファイルから呼び出し、ブラウザ上で可視化結果を閲覧。

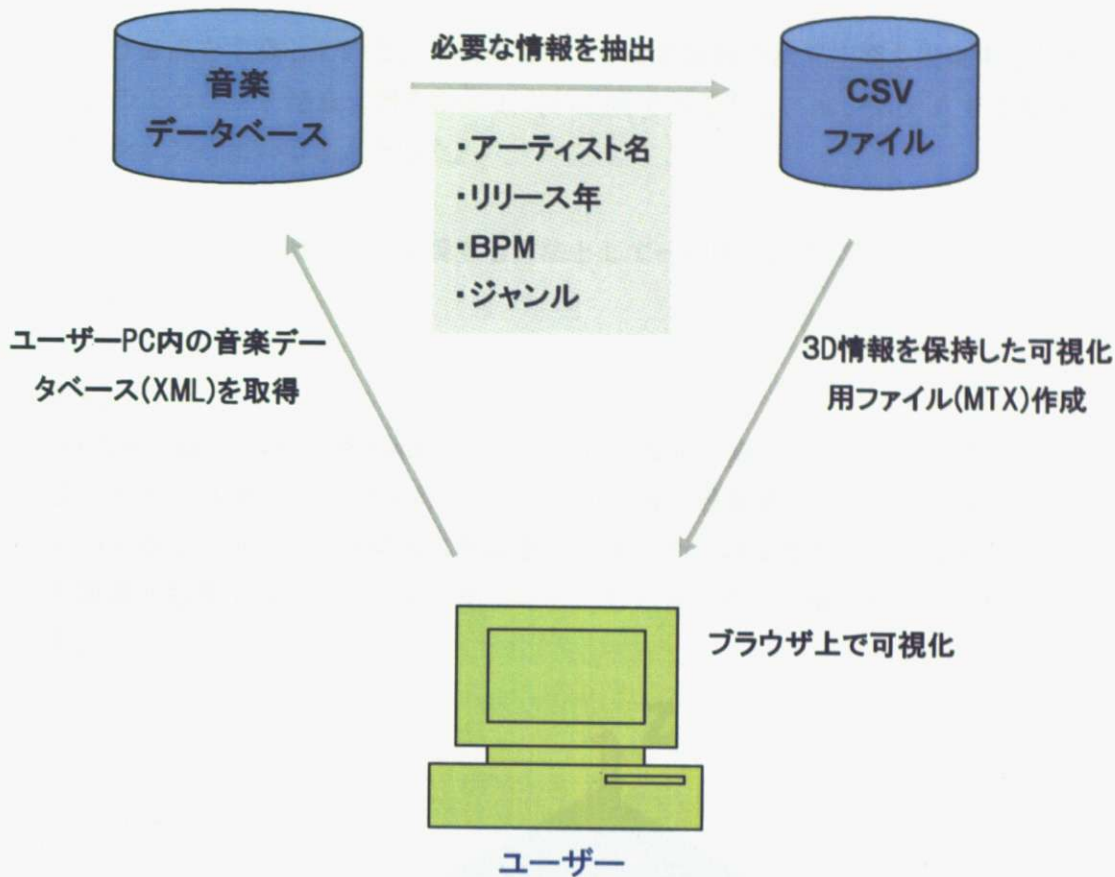


図 3.4 システムアーキテクチャ

3.3.2. 機能

以下、本システムの機能を詳細に説明する。

(1) カラーグラデーションによる空間的位置把握

従来の3次元表示を用いたグラフ表現システムでは、プロットされた点が3軸のどの位置に対応しているのかがわかりづらいという問題点があった。これは空間的位置把握のための手がかりの少なさに原因があると考えられる。J.J.Gibson は生態学的視覚論²²⁾の中で人間の空間認識について述べており、

- ・ 面（サーフィス）と肌理（テクスチャ）の変化（勾配）によって空間を知覚する。

- ・ 網膜は点ではなくパターンを認識する器官である。

という2点を主張している。本来の2次元平面に疑似3次元空間を実現するためには、上記のような点を踏まえ何らかの工夫が必要であると、本システムではカラーグラデーションを用いることとした。

コンピュータ内で色を実現する方法として一般的にはRGB系とHSV系があり、その説明を述べる。

RGB

赤(Red)、緑(Green)、青(Blue)の三つの原色を混ぜて幅広い色を再現する加法混色の一様である。RGBカラーモデルでは、赤・緑・青の各要素がどれだけ含まれているかで記述することができ、各要素は輝度最小(闇)から輝度最大までの範囲をもつ。もし各要素とも最小なら表示結果は黒になる。もし各要素とも最大なら表示結果は白になる。

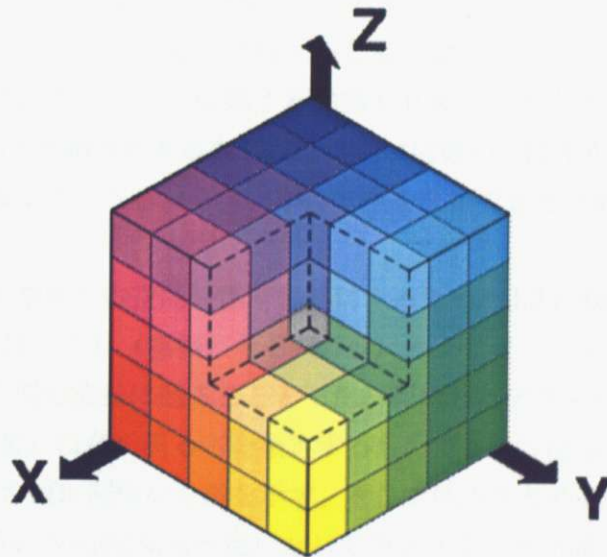


図 3.5 立方体状にマッピングされた RGB カラーモデル

HSV

色相(Hue)、彩度(Saturation)、明度(Value)、の三つの成分からなる色空間。

- ・ 色相 — 色の種類 (赤、青、黄色のような)。0~360° の範囲。
- ・ 彩度 — 色の鮮やかさ。0~100%の範囲。
- ・ 明度 — 色の明るさ。0~100%の範囲。

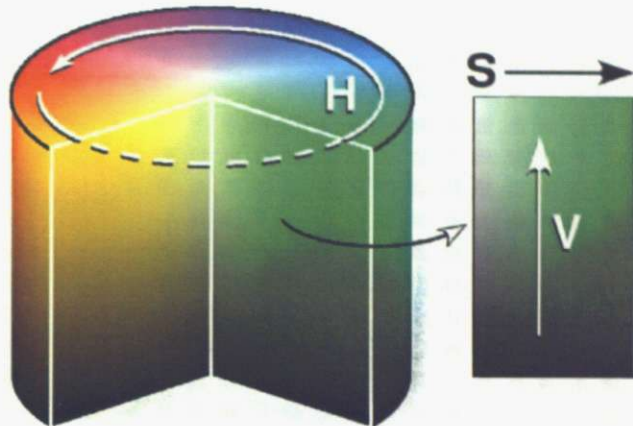


図 3.6 円柱状にマッピングされた HSV カラーモデル

Photoshop などのアプリケーションでユーザーが色を選択するような場合、HSV のカラーモデルがよく用いられる。これは HSV モデルのあり方が人間が色を知覚する方法と類似しているためである。RGB は加法混色によるモデルであり、原色の組み合わせによって色が定義されるため、目的の色を実現するためにはどの色をどれくらい混色すればどういった色になるのかということに関する知識が必要となり直感的ではない。それに対し HSV はより人間と親和性のある内容、この色は何色か・鮮やかさはどのくらいか・明るくしたり暗くするにはどうしたらよいか、で色を決定できるため直感的である。

そこで、仮想 3 次元空間での空間認識を容易にするため、J.J.Gibson の主張にある 2 点を考慮し、HSV における H (色相) と S (彩度) のグラデーションを用いて作成したテクスチャを 3 次元空間の底面に貼ることとした。利用したテクスチャを図 3.7 に示す。HSV において V (明度) は色の明るさを指定するものであり、図 3.6 からもわかるように V=0 の時、H と S の値に関係なく黒を示す。そこで他の色との違いを最も明確に判別することができるように V=100% (一定) としてテクスチャを作成した。

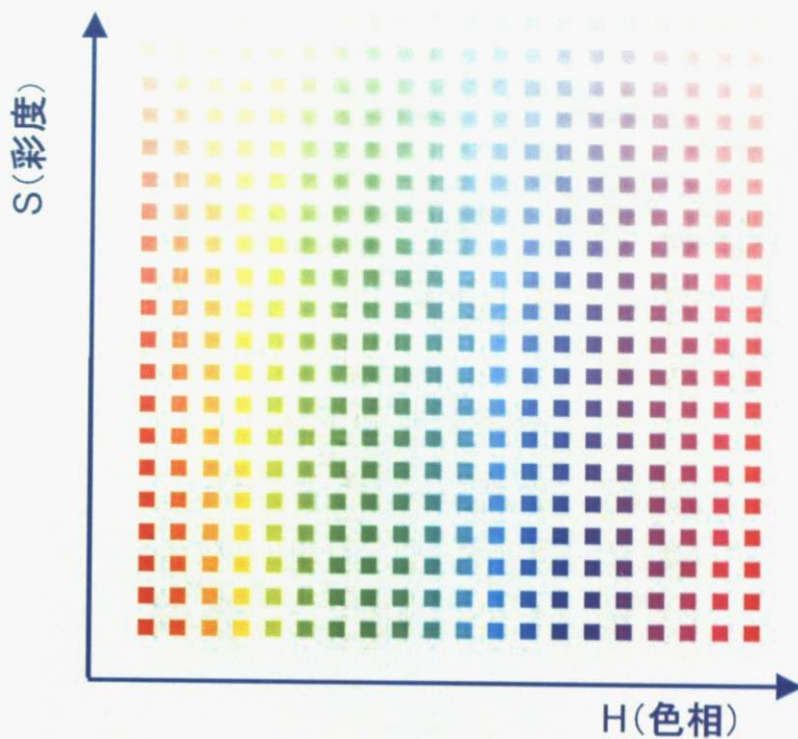


図 3.7 空間位置把握のためのテクスチャ

図 3.7 に示すテクスチャを仮想 3 次元空間内の底面 (XY 平面) に貼り付け、空間内にプロットされるオブジェクトが保持する X、Y 座標値を元に各オブジェクトのカラーを決定する。Z 座標値に関しては V=一定のため、通常の高さだけで表現した。図 3.8 と図 3.9 にカラーグラデーションの効果を示す。なお、各 XYZ 軸に割り当てた情報は以下の表のとおりである。

- X 軸 : BPM (70 ~ 140)
- Y 軸 : アーティスト名 (A ~ Z)
- Z 軸 : リリース年 (1960 ~ 2010)

図 3.8 と図 3.9 は全く同じデータを同じアングルからキャプチャしたもので、図 3.8 はグラデーションなしの場合、図 3.9 はグラデーションありの場合の図である。

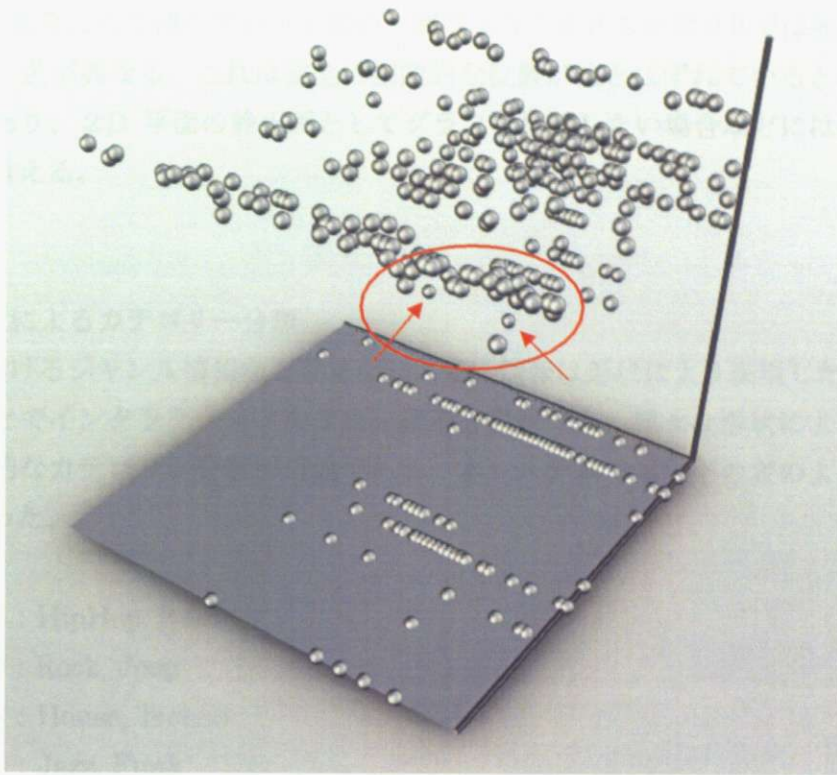


図 3.8 グラデーションなし

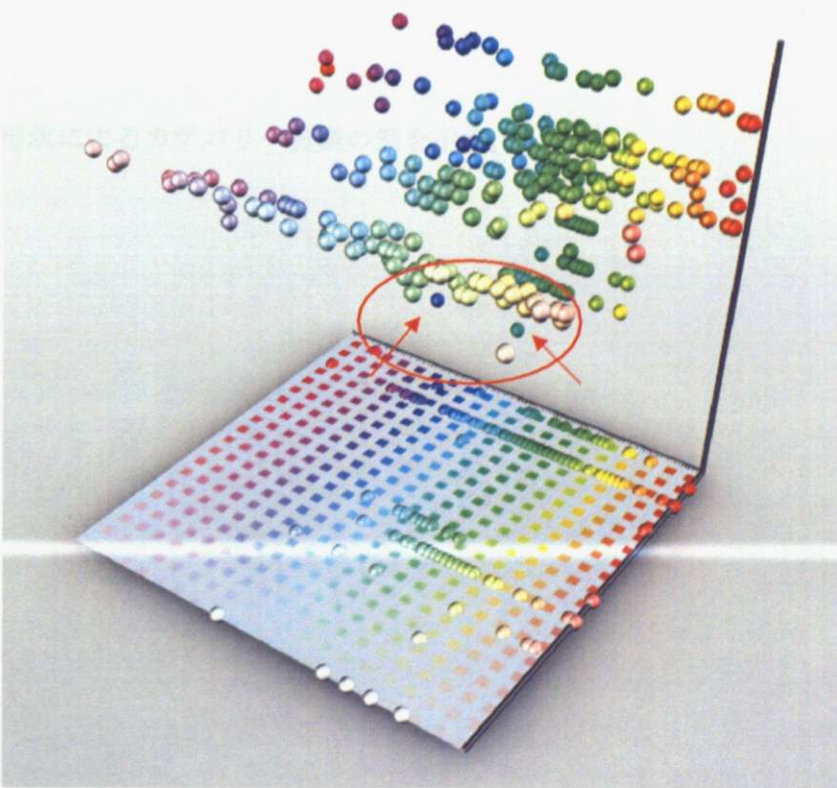


図 3.9 グラデーションあり

図 3.8 と図 3.9 の赤で囲まれ、かつ、矢印で指されているオブジェクトに注目する。図 3.8 では一見それらは他のプロット群の一部のように見えるが図 3.9 では他のプロット群とは明確に色が異なる。これは完全に空間的な位置が他とはずれているということを示すものであり、2D 平面の静止画としてグラフを提示したい場合などには非常に効果的であると言える。

(2) 3D 形状によるカテゴリー分類

音楽におけるジャンル情報など数値化しにくい情報は形状により表現した。Web3D を用いたことでインタラクティブな 3 次元表示を実現でき、様々な形状により、より多様かつ直感的なカテゴリー分類が可能である。本システムでは以下の表のようなジャンルわけを行った。

立方体 : HipHop, R&B
球体(大) : Rock, Jpop
四角錐 : House, Techno
円柱 : Jazz, Funk
球体(小) : その他

図 3.10 に形状によるカテゴリー分類の例を示す。

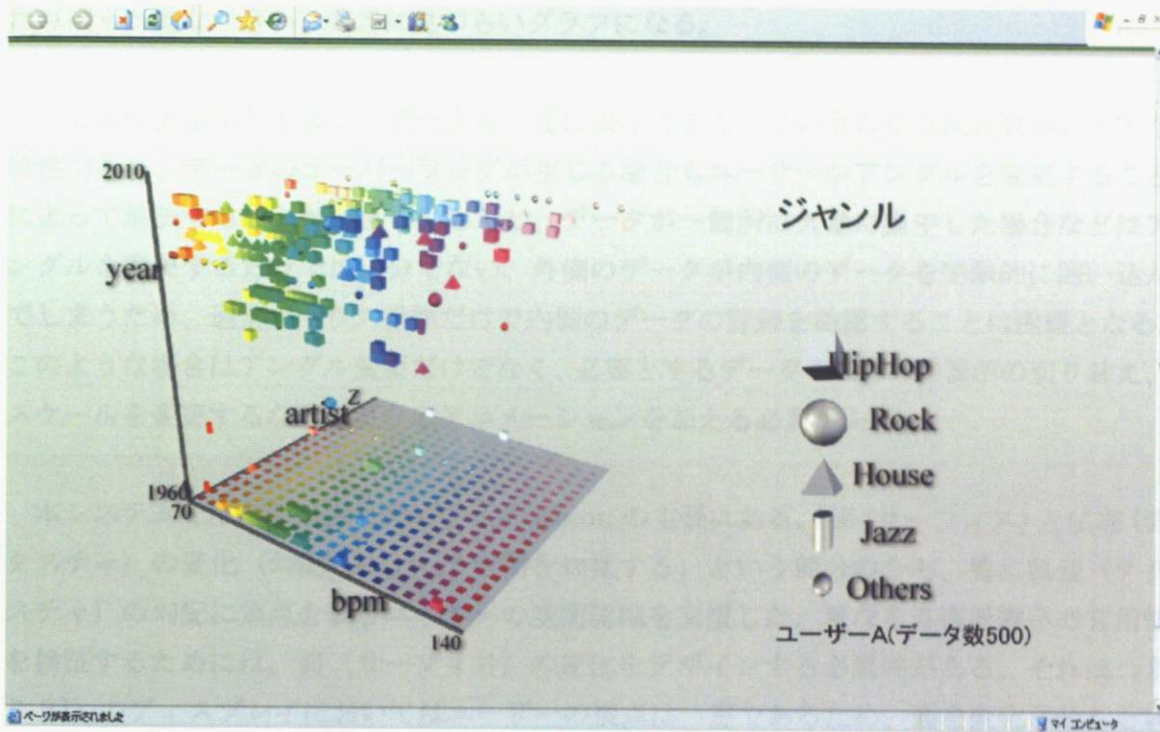


図 3.10 形状によるジャンル分類例

(3) マウスオーバーによる詳細情報の表示

プロットされたオブジェクト上にマウスカーソルをのせるとテキストベースの詳細情報が表示される。情報可視化では大量の情報を扱うことが多く、ユーザーのニーズに応じて新たな情報を提示することが重要となる。

3.4. 考察

インタラクティブな3次元表示を統計的なグラフ表現に応用したことで、どのような知見が得られたか考察をまとめる。

▶ データ表示量

本システムでは、約 3000 のデータを一度にストレスなく表示することができた。3000 というデータ数をエクセルの表形式のデータとして表示させようとすれば約 90 画面分の表示領域が必要となる。また、従来の 2D 平面へプロットする場合には 3次元空間に用いる 3平面をそれぞれグラフにする必要があり、特に、データ数が多くなるとプロットされ

た点同士が重なり合い、非常に見づらいグラフになる。

「2次元表示よりも多くのデータを一度に表示できる」というのは3次元表示の大きな特性であり、データのオーバーラップが生じる場合もユーザーがアングルを変更することによって解決することができる。しかし、データが一箇所に大量に集中した場合などはアングルを変更するだけでは十分でない。外側のデータが内側のデータを閉鎖的に囲い込んでしまうため、通常のマウス操作だけで内側のデータの詳細を確認することは困難となる。このような場合はアングル変更だけでなく、必要とするデータの表示・非表示の切り替え、スケールを変更するなどの更なるアニメーションを加える必要がある。

本システムでは空間認識においてJ.J.Gibsonの主張にある、「面(サーフィス)と肌理(テクスチャ)の変化(勾配)によって空間を知覚する」という部分のうち、特に肌理(テクスチャ)の勾配に焦点を絞りユーザーの空間認識を支援した。更なる3次元表示の有用性を検証するためには、面(サーフィス)の変化をデザインする必要がある。それはつまり2Dのディスプレイにおいてはユーザーの視点は一定であるため、面を変化させるということはアニメーションをデザインするということである。

本システムにおいて2次元表示よりも多くの情報を一度に表示できた、という点は評価できるポイントではあるが、今後の発展のためには、大規模データを表示することによるデータのオーバーラップの解消、仮想3次元表示における空間認識、を実現するためのツールとしてアニメーションの利用は非常に重要な視点である。

▶ クラスタによるデータ特性の判断

本システムを用いて、あるユーザーの音楽データベースを表示させた場合にクラスタが確認される場合があった。音楽という情報においてクラスタが生じる、つまり、データに偏りが生じるということは、そのユーザーの音楽に対する嗜好性が現れていると考えられる。

図3.11と図3.12に二人のユーザーの可視化結果を示す。データ数は両者ともに500である。

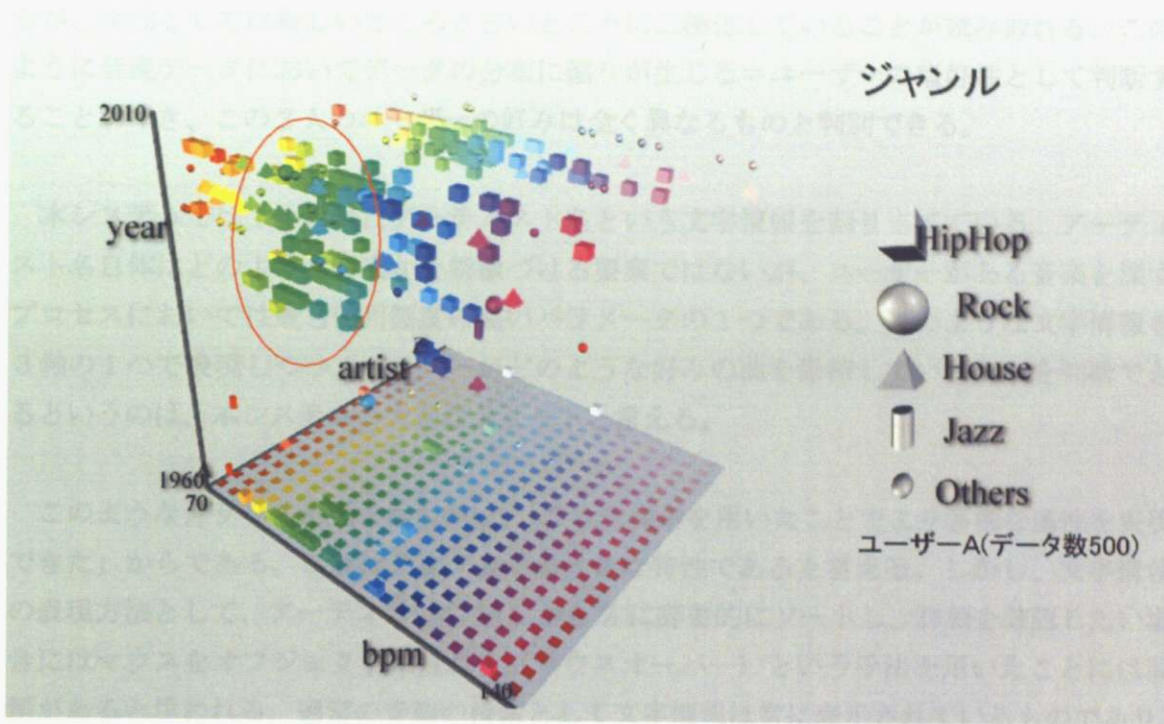


図 3.11 ユーザーAの可視化例

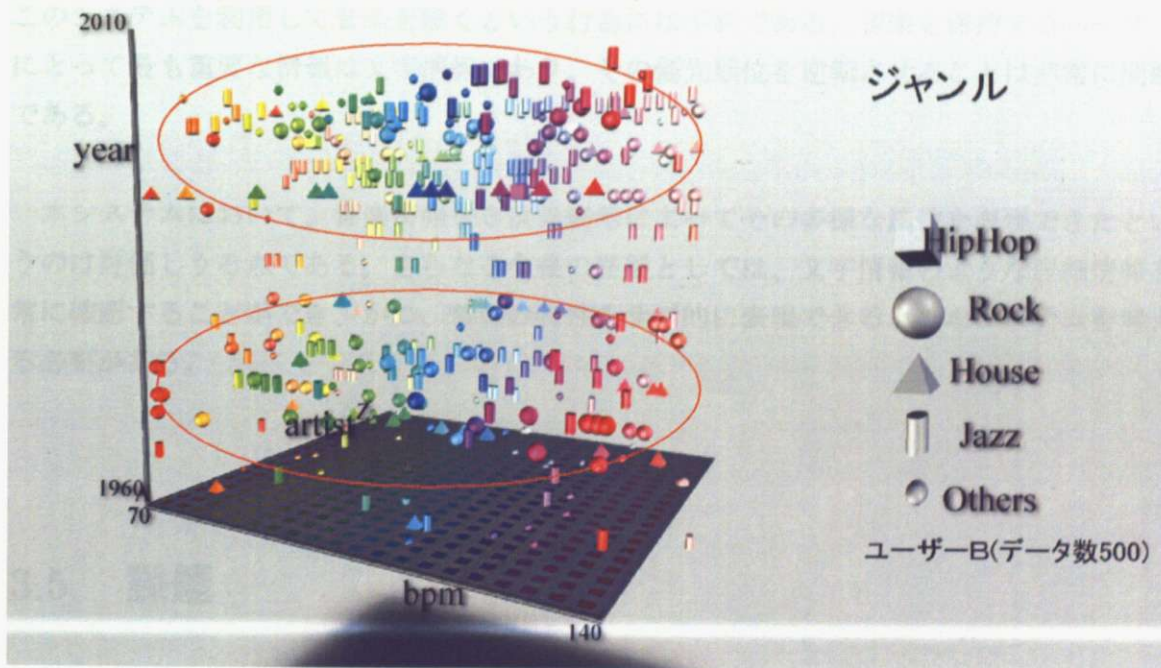


図 3.12 ユーザーBの可視化例

ユーザーAの可視化例からは、まず形状として四角が圧倒的に多いことからヒップホップを好み、かつ、BPM(テンポ)は遅く、年代は新しめのところにオブジェクトが集中している。それに対し、ユーザーBでは、形状としては円柱が多く、BPMは均等に分散してい

るが、年代としては新しいところと古いところに二極化していることが読み取れる。このように音楽データにおいてデータの分布に偏りが生じる＝ユーザーの嗜好性として判断することができ、この2人のユーザーの好みは全く異なるものと判別できる。

本システムでは1つの軸にアーティスト名という文字情報を割り当てている。アーティスト名自体はどのような音楽かを特徴づける要素ではないが、ユーザーがある音楽を探すプロセスにおいては最も使用頻度の高いパラメータの1つである。このような文字情報を3軸の1つで表現しつつ、ユーザーがどのような好みの曲を保持しているのかを判断できるというのは、本システムの大きなメリットと言える。

このようなメリットが生まれたのは、「3次元表示を用いたことでより多様な属性を表現できた」からである。これは3次元表示の大きな特性であると言える。しかし、文字情報の表現方法として、アーティスト名をAからZに辞書的にソートし、詳細を確認したい場合にはマウスをオブジェクトに重ねる（マウスオーバー）という手法を用いたことには問題があると思われる。通常の音楽の検索として文字情報は常に表示されているものであり、音楽の特徴などはその音楽を保持するユーザーが一番理解しているものである。本システムはユーザーがどのような音楽を保持しているのかを知りたいマーケット側には適するが、このシステムを利用して音楽を聴くという行為には不利である。音楽を保持するユーザーにとって最も重要な情報は文字情報であり、その優先順位を逆転させることは非常に問題である。

本システムにおいて、音楽情報を3次元表示によってその多様な属性を表現できたというのは評価しうる点である。さらなる今後の発展としては、文字情報のような詳細情報を常に確認することができ、かつ、情報の特性をサブ的に表現できるようなシステムを考える必要がある。

3.5. 課題

最初に、考察によって得られた3次元表示に関する知見をまとめておく。

3次元表示の利点

- ・ 2次元表示よりも多くのデータを一度に表示できる
- ・ 複数の軸を設定できることにより多様な属性を表現できる

また、3次元表示の利用に関して以下の課題もあげられた。

課題

- ・ データのオーバーラップの解消、かつ、仮想3次元表示における空間把握の実現方法として、アニメーションの効果的な利用を考える。
- ・ 文字情報のような詳細情報を常に確認することができ、かつ、情報の特性をサブ的に表現できるような手法を考える。

以上のような3次元表示の利点・課題を踏まえ、Web3DとAPIを用いてネットワーク上のデータベースを可視化することを考える。

現在、様々なデータベースを公開するAPIが提供されている。Google Mapsでは地図データベース、Amazonでは商品データベース、YouTube²⁹⁾では動画データベースなどがAPIによって利用できる。音楽データベースに関してはSeeqPodという企業がネット上の音楽ファイルを検索できるサイトを運営しており、そのAPIも公開している。これらのサイトでは全て、自分の目的の情報を得るためのインターフェースとしてキーワードを入力するフォームが用意されている。そして、返ってくる検索結果は2Dで一覧表示され、ユーザーは画面をスクロールさせながら目的の情報を探す、というプロセスを辿る。

自分の探したいものが明確にわかっている場合、複数のキーワードで絞り込み、少数の検索結果を吟味すればよい。しかし、自分の探したいものに対しての知識が明確でない場合は、漠然としたキーワードで検索するため、返ってくる結果も大量となることが多い。この中からスクロールやページ切り替えを何度も行い、目的の情報を探し出すのは非常に骨の折れることである。実際、現行のYouTubeのサイトでは検索結果を一画面にせいぜい5つ程度しか表示することができていない。その例を図3.13に示す。

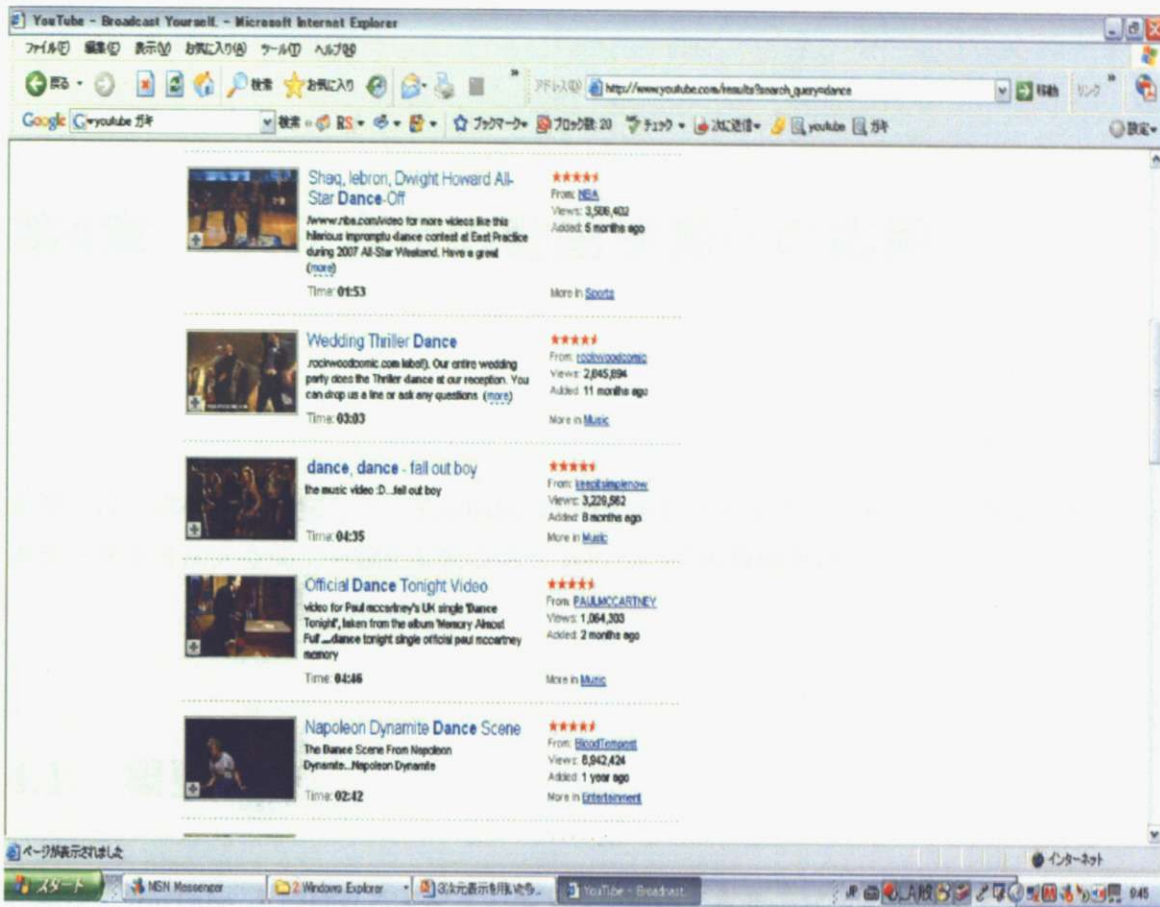


図 3.13 YouTube による検索結果表示例

図 3.13 からわかるように YouTube に投稿された動画には、

- ・ タイトル (文字)
- ・ 詳細説明 (文字)
- ・ 再生回数 (数値)
- ・ 再生時間 (数値)
- ・ サムネイル画像 (画像)

というような情報が含まれており、音楽以上に多様な形式の情報が混在するマルチメディアである。よって、さらに効率の良い表示が求められるデータベースであると言える。そこで、ここにはまだまだ改良の余地が残されていると見なし、次の目的を以下のように設定する。

「YouTube の API を利用し、動画に含まれる複雑な XML 形式のメタデータを仮想 3 次元空間で効率よく可視化する。」