

XML によるインタラクティブな統計グラフ

— Web ベースの統計環境への活用 —

山本 義郎¹ · 藤野 友和² · 飯塚 誠也³

(受付 2006 年 7 月 3 日 ; 改訂 2006 年 11 月 28 日)

要 旨

Web 上での統計グラフの表示方法として, XML テクノロジーを利用し, 2 次元グラフの表示のために SVG, 3 次元グラフの表示のために X3D の利用を提案する. SVG と X3D はともに, Web ブラウザで閲覧することにより, Web ページの一部としてベクターグラフィックスを描画できるグラフ記述言語であり, ビューアの機能を用いて拡大・縮小などができ, プログラムにより更なるインタラクティブな機能を実装することが可能である.

本論文では, SVG および X3D により実現できるインタラクティブな統計グラフの作成方法について紹介するが, XML ベースのグラフィックスフォーマットである SVG と X3D の理解のためには, XML についての理解が不可欠であるため, まず XML の概要を述べ, SVG および X3D の特徴について概説する. その後で, SVG, X3D によるインタラクティブ統計グラフを紹介するとともに, 著者らが作成した XML ベース統計グラフの作成ツールについても紹介する. 最後に XML ベースのグラフィックの活用方法として, Web を利用した教育ツールを例に, SVG を利用したティーチウェアおよび XML ベースのインタラクティブテキストについて提案を行う.

キーワード: SVG, X3D, Web ベースグラフ.

1. はじめに

インターネットの普及と利便性の向上により, 多くの情報が Web 上に公開され, 官公庁のみならず多くの統計情報についても, 紙媒体による情報の公開から Web での公開へとシフトしている. Web における統計情報の公開においては, 単なる表の形のデータの公開だけでなく, データを理解するためのグラフの提示も必要である. Web への要求の高まりに応じ, Web ページも単なるテキストによる表現だけでなく, 動きを伴った利用者の視覚に訴える表現へと変化している. 統計グラフについても事前に作成された静的な(表示の変更ができない)グラフを配置するだけでなく, 利用者の要求に応じてグラフを生成・表示したり, 時々刻々変化する株価などを表示するグラフ(ダイナミックグラフ)や, ユーザの要求に応じてラベルの表示・非表示を変えるなどインタラクティブ(対話的)な処理を行うことが可能なグラフ(インタラクティブグラフ)を活用しているサイトも数多く見られるようになってきた. その種のグラフは,

¹ 東海大学 理学部数学科: 〒259-1292 神奈川県平塚市北金目 1117

² 福岡女子大学 人間環境学部: 〒813-8529 福岡市東区香住ヶ丘 1-1-1

³ 岡山大学大学院 環境学研究所: 〒700-8530 岡山市津島中 3-1-1

Java や Flash により実現されているものが多く、統計解析システムにおいても、S-PLUS の graphlet のように、作成したグラフがインタラクティブな機能が利用できる形で Web 上へ公開できるものも開発されている。その種のグラフの実現方法として、近年注目されている XML にもとづく (XML ベースの) グラフィックスフォーマットである SVG (Scalable Vector Graphics, W3C, 2003b) および X3D (Extensible 3D, Web3D, 2004) を用いることも可能である。本論文では、Web 上でインタラクティブなグラフ表現を実現するために、SVG および X3D の XML によるグラフィックスフォーマットを用いる方法について紹介する。そのため、まず XML についての概要を与え、SVG および X3D の概要を紹介する。一般のインタラクティブグラフについては、Symanzik (2004) が詳しい。

2. XML ベースのグラフィックスフォーマット

XML ベースのグラフィックスフォーマットである SVG および X3D を利用して統計グラフを作成する方法について述べる前に、まず XML についての理解のために XML の概要を与え、統計科学関連の領域において現在どのように XML が用いられているかについて紹介する。また、SVG および X3D の歴史や仕様などの概説を与える。

2.1 XML の概要

XML (Extensible Markup Language) は ISO (国際標準化機構) が、多様なソフトウェア環境で文書互換がとれるように開発した、タグにより記述する言語 SGML (Standard Generalized Markup Language) を、インターネットに対応させた言語であり、W3C (World Wide Web Consortium) が標準化を行っている (W3C, 2003c; W3C, 2004)。インターネットを通じた情報公開や、情報の交換の利便性を高めるため、さまざまな領域においてデータやインタフェースの標準化が行われ、W3C が XML に準拠した形での標準化を推進し、多くの仕様が各種団体において策定されている。

ホームページ記述のための言語である HTML の、XML による標準化である XHTML や、数式の記述に用いられる MathML (Mathematical Markup Language, W3C, 2003a) などの仕様は、XML のタグを規定することからタグセットと呼ばれたり、特定領域における XML であることからボキャブラリと呼ばれる。Web を表現する方法は、HTML だけを利用する形態から、多種の XML タグセットを利用する形態へとシフトしている。図 1 に W3C Day Japan (2003/11/14) における Steve Bratt の資料 (<http://www.w3.org/2003/Talks/1114-W3CDay-Japan/steve-foundation/>) を参考に作成した、今後の Web 表現についての XML の関連図を与えた。

XML 文書は ASCII テキスト形式で記述され、1) XML 宣言、2) DTD (文書型定義、省略可)、3) XML インスタンス (本体) からなる。XML インスタンスはルート要素の下に複数の子要素をもつツリー構造であるため XML ツリーとも呼ばれる。

Web を利用する際に HTML でなく XML を利用することの利点は、1) 1 つの XML 文書内で複数のタグセットをネームスペースという概念を用いて使い分けることができる、2) XHTML において DOM (Document Object Model, W3C, 1998) を利用することによりいくつかの XML ドキュメントの構成要素に対してアクセス可能であること、があげられる。1) の性質を利用して、XHTML ドキュメントにおいて、MathML を用いて数式を記述した Web ページが実現できる (方法の詳細は後述)。2) に関しては、XML ドキュメントの内部要素にアクセスするためのプログラミングインタフェース仕様である DOM の利用により、XML ドキュメントに対して、記述内容の変更やスタイルの変更などが可能となり、表現力を大幅に高めるものである。更なる利点として、XML を利用したシステムを作成する際には、XML の構文解析などの処理方法が同じであるため、システム開発における負担が軽減できることがあげられる。

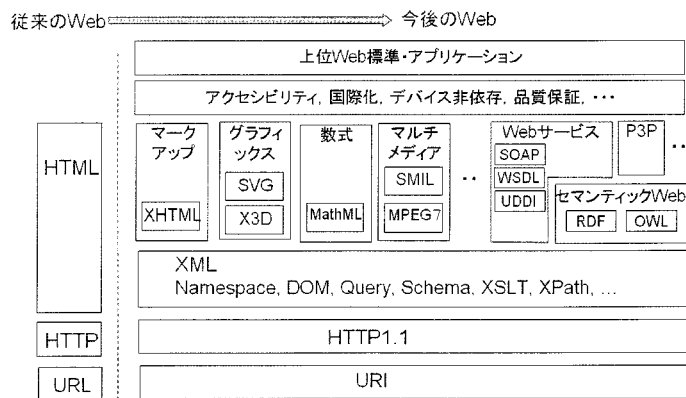


図 1. 従来の Web と将来の Web の概念図.

2.2 統計科学領域における XML の利用

統計科学の領域においては、いくつかの独自のタグセットを定義して用いる試みが提案されている。日本においては、統計データの記述形式として DandD (横内・柴田, 2001; DandD Project IV, 2004) が提案されている他、メタデータの記述 (大津, 2004) や、DoSS@d というシステムにおけるデータおよびデータ解析記述方法としてのタグセット (Mori et al., 2004) などが提案されている。統計ソフトウェア R において、XML によりデータを記述し R や MATLAB, Octave においてデータの互換性を保つ試みを提案している StatDataML (Meyer et al., 2004) も独自のタグセットの 1 つである。これらの XML の活用の提案は、データの説明であるメタデータの記述方法として独自の書式ではなく、XML に則った記述をすることにより利用者の利便性に配慮している。

統計科学に関連するボキャブラリとしては、データマイニングソフトウェアのモデルの記述方法に利用されている PMML (Data Mining Group, 2005) があり、IBM や SAS なども構成員である Data Mining Group により策定されている。

コンピュータ上でグラフィックスを表現する方法としては、ラスターグラフィックスとベクターグラフィックスの 2 通りがある。Web 上で統計グラフを表示するためには、通常あらかじめ作成したグラフを、Web ページで表示可能な GIF, JPEG, PNG などの形式で表示させるのが一般的であるが、これらのラスターグラフィックスは、画像を、色のついた点(ドット)の羅列として表現したもので、画像に描かれている内容については一切の情報を持たないため、拡大・縮小により画像の劣化が生じる。一方、ベクターグラフィックスは、点とそれを結ぶ線や面の方程式のパラメータ、および塗りつぶしなどの描画情報によって画像を表現したものであるので、これらの情報を利用して、拡大・縮小・変形を行う際に、ソフトウェアの機能で再描画することによって、画質の劣化を防ぎ、高品質なグラフが提供できる。

XML ベースのグラフィックスフォーマットである SVG, X3D は、次のような特徴をもっている。

- ベクターグラフィックス
- テキスト形式であるため簡単に編集でき、プログラムからも生成可能
- プラグインを用いてブラウザ上でグラフ表示可能
- JavaScript と DOM を利用してインタラクティブな機能を実装可能

SVG および X3D は Web における利用を念頭に入れ開発されたフォーマットであるため、Web ブラウザ上で表示するためのプラグインが用意され、プラグインを導入するだけで、ブラウザ上で高性能なグラフの表示が可能になる。そのようなプラグインやビューアのほとんどは、グラフの拡大・縮小や移動といった基本的な機能を標準的にもっており、グラフそのものに必ずしもプログラムを含める必要がない。操作機能を付加するためには、XML によるグラフの記述において、JavaScript などのスクリプトを記述することで、そのグラフにインタラクティブな機能を実装することもでき、さらにグラフにとどまらずアプリケーションを作成することも可能である。

XML 形式であることは、単に構造化されたテキストファイルであるという点も利点である。つまり、単純な統計グラフを XML 形式の SVG などで作成する場合、プログラミング言語を問わない。さらに、グラフが地図や CAD などのように外部のリソースに密接に関連したものである場合に、グラフを構成する要素とそれらの関連についての情報などを、グラフ自身に埋め込むことができる利点もある。加えて、XML 形式であることは、他の XML 関連の規格やツールと連携できるという点でも有利である。XML の普及に伴う更なる効果が期待でき、今後が期待される。

XML の活用の概要については立川(2004)が詳しく、また XML ベースのシステム開発のためには Fitzgerald (2004) が詳しい。統計科学における XML の利用については、山本・藤野(2005)に総合報告として詳細にまとめられている。

2.3 SVG の概要

SVG の最初のバージョンである SVG1.0 は、2001 年に W3C により勧告された。SVG がリリースされる以前には、これを実現するための仕様として、マイクロソフトによる Vector Markup Language (VML) と、Adobe による Precision Graphics Markup Language (PGML) が W3C に提案されていたが、W3C はこれらの仕様を取り込んだ形で 2001 年の 9 月に SVG をリリースし、最新版は 2003 年 1 月にリリースされた SVG1.1 (W3C, 2003b) であり、更なる機能を取り入れた SVG1.2 (W3C, 2005a) が Working Draft として提案されている。

SVG はベクターグラフィックスであるため、探索的データ解析の場面において、視覚化による分析を行ったり、GIS と連携した統計解析を行うなど、拡大・縮小を行って、データの全体や細部を切り替えてみる必要がある際に有効である。また、SVG では DOM を利用したインタラクティブな機能を有したグラフが作成可能であり、探索的データ解析や、GIS アプリケーションにおけるグラフ表示に重要な機能である。

SVG はテキストファイルにより構成され、表示するためにはビューアが必要である。現在、最も利用されているのは Adobe の SVG Viewer であり、Windows98-XP と MacOS8.6-9.1 における Internet Explorer 上で利用可能であるが、MacOS に関しては静止 SVG のみのサポートとなっている。本論文では、表示環境として Windows 向けの SVG Viewer ver.3.0 を用いている。この他の SVG ビューアとしては、Apache の XML Project による Batik SVG Toolkit (<http://xmlgraphics.apache.org/batik/>) に含まれる Java アプリケーションの Batik-Squiggle や Corel の SVG Viewer がある。また、オープンソースの Web ブラウザである Firefox (バージョン 1.5 以降) などでは、Web ブラウザ内でのネイティブな SVG の実装を実現している。

SVG の言語についての詳細は、Eisenberg (2002) などが詳しいが、ここでは後述する SVG の統計グラフの理解のために必要な最低限の概要を説明する。SVG の基本構造を紹介するために、SVG ドキュメントの簡単な例を以下に与える。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

```

<svg width="300" height="180">
  <title>Basic structure</title>
  <desc>Example of basic structure</desc>
  <!-- 図形などの SVG 要素を以下に記述する -->
</svg>

```

1 行目は XML 宣言, 2 行目は文書型定義である。SVG のルート要素は, `<svg>` 要素であり, SVG の描画サイズは `width` および `height` の属性値で指定する。 `<title>` 要素には, SVG 文書のタイトルを記述し, この内容は Web ブラウザや SVG ビューアのタイトルバーに反映される。 `<desc>` 要素には, その SVG 文書に関する説明を記述する。

SVG における座標系は, 2 次元の直交座標系であり, y 軸の正の方向が原点から下向きに設定されており, 一般的な 2 次元の直交座標系に比べると上下が反転した形となっている。 SVG ビューア上で実際に画像として表示される矩形領域は, ビューポートと呼ばれる。 SVG における座標系の任意の矩形領域をビューポートに表示させたい場合には, 以下のように `viewBox` 属性を指定する (図 2)。

```
viewBox="originX originY width height"
```

図 3 に SVG の基本的な図形要素とその記述方法を示している。 図形の位置や大きさなどの属性は, 各要素の属性として指定し, 線や塗りつぶしの色, 線の幅などの表示スタイルは, `style` 属性で Cascading Style Sheet (CSS) の形式で記述する。

SVG におけるテキストは, `<text>` 要素で記述し, タグに囲まれた文字列が, 属性 x と y に指定した位置に表示される。 `style` 属性でフォントの大きさや種類, 太さ, 色などを指定する。 文字の縁と内部はそれぞれ `stroke` と `fill` により別々に色を指定できる。 また, `text-anchor` プロパティで, x と y 属性でテキストの位置を指定できる。

また, 複数の図形要素をまとめて扱えるグループ要素 `<g>` で囲むことで, グループ全体のスタイルを指定したり, 位置を指定したりすることができる。 `<defs>` 要素の中で, `id` 属性を使って図形要素に名前をつけておくことにより, `<use>` 要素を用いて複製が利用でき, `<g>` 要素を用いて複数の図形要素をまとめて複製を作成することもできる。

SVG でグラフを作成するためには, これらの図形およびテキストを利用すればよい。 図 4 は SVG で記述された散布図行列を, Adobe SVG Viewer がインストール済みの Internet Explorer

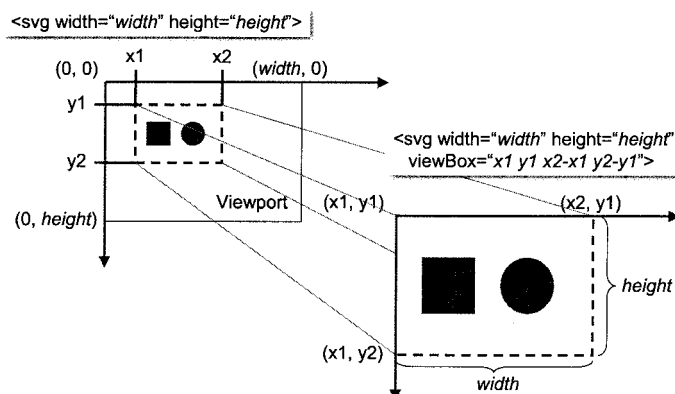


図 2. ビューポートとビューボックス。

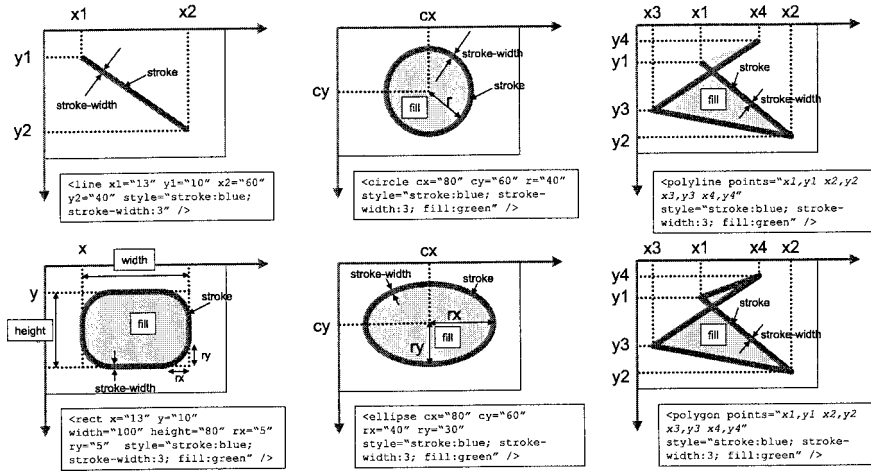


図 3. 基本図形.

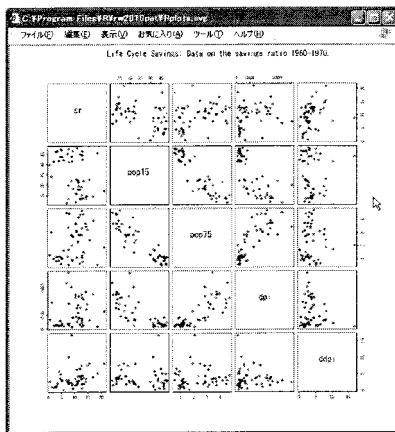


図 4. SVG で実現された散布図行列.

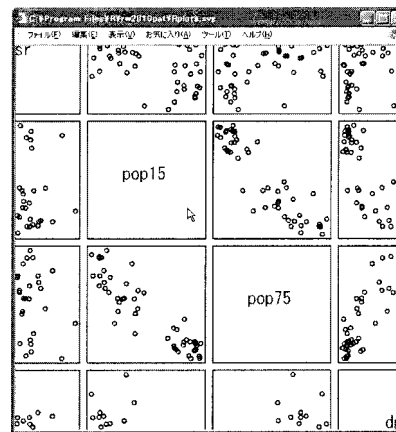


図 5. 拡大表示された散布図行列.

で表示した様子を示している. SVG Viewer の機能により, グラフ上の強調したい部分で右クリックし表示されるメニューで「ズームイン」を選択すると拡大表示される(図 5).

2.4 X3D とは

X3D は, 3次元グラフィックスをインターネットにおいて利用可能にするために, 転送するデータ量が少なくすむように考案されたモデリング言語である VRML (Virtual Reality Modeling Language) の後継の言語であり, Web3D (Web3D Consortium) が規格を制定している.

X3D の前身である VRML は 1994 年に VRML1.0 が提案され, VRML97 として知られる VRML2.0 は ISO 規格として 1996 年に提案されたものである. X3D は 2004 年に Web3D コンソーシアムにより ISO 規格として, XML フォーマットも含めた形で提案され (Web3D, 2004), 2005 年に XML および VRML による X3D エンコーディングを含む 6 つの ISO 標準が提案されている. 最新の仕様については Web3D の Web サイト (<http://www.web3d.org/>) で確認できる.

SVG 同様に X3D も X3D プラグインが導入されている Web ブラウザもしくは X3D ビューアを利用して 3 次元仮想空間を表示できる。Windows 環境で利用できる Octaga Player (<http://www.octaga.com/>) は個人利用や非商用には無料の X3D ビューアであり、Internet Explorer, Netscape, Opera, Mozilla などのブラウザのプラグインとしても対応している。本論文における X3D の表示は、この Octaga Player を利用している。その他にも、Flux Player をはじめ商用からフリーウェアに至るまで多くのビューアが提供されている。最新のビューアについての情報は (http://www.web3d.org/tools/viewers_and_browsers/) で得ることができる。

X3D は、その利用目的に応じて、地理空間情報を利用する GeoSpatial, ヒューマノイドアニメーションのための H-Anim, CAD などのいくつかのワーキンググループにより、開発・標準化作業が行われている。X3D の記述により実現される仮想空間は X3D ワールドと呼ばれる。X3D は、ワールドの実現規模により、Interchange, Interactive, Immersive, Full の 4 つのプロファイルを指定でき、プロファイルの指定により利用できる機能が制限される。単なる 3D モデリングの場合には Interchange を、センサーの利用によるインタラクティブ性を利用する場合には Interactive を、GeoSpatial などのコンポーネントを利用する場合には Full プロファイルを利用する。

X3D の基本構造を紹介するために X3D ドキュメントの簡単な例を以下に与える。

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE X3D PUBLIC "ISO//Web3D//DTD X3D 3.0//EN"
"http://www.web3d.org/specifications/x3d-3.0.dtd">
<X3D version='3.0' profile='Interchange'>
  <Scene>
    <Transform translation='3 0 1'>
      <Shape>
        <Sphere radius='4' />
        <Appearance>
          <Material diffuseColor='1 0 0' />
        </Appearance>
      </Shape>
    </Transform>
  </Scene>
</X3D>
```

1 行目は XML 宣言、2 行目は DTD 宣言である。X3D のルート要素は <X3D> タグであり、version 属性によって X3D のバージョンを、profile 属性により利用プロファイルを指定する。X3D では通常、要素をノードと呼び、属性をフィールドと呼ぶため、本論文でも以降の説明ではそれに従う。X3D ワールドは、<Scene> ノード内に様々なオブジェクトを配置することにより構築される。例では、赤色 ((1,0,0)、色は RGB を 0 から 1 の値で指定する) で半径 4 の球を (3, 0, 1) 座標に配置している。

X3D の座標系は右利きのデカルト 3D 座標系を使用している。初期設定では、オブジェクトは正の x 軸を右方向、正の y 軸を上方向とし、正の z 軸方向は手前に向けられている。視点は Viewpoint で指定できる。

X3D ワールドを構築するオブジェクトは <Shape> ノード内に記述し、基本的な図形オブジェクトとしては Box, Sphere, Cylinder, Cone がある。オブジェクトの配置位置は、<Transform> ノードの translation フィールドに 3 次元ベクトルの形で、例のように <Shape> ノードを囲む形で指定する。オブジェクトのサイズや半径などはフィールドに記述し、Box では size フィールドで箱の幅・高さ・奥行きをベクトルで、Sphere では radius フィールドで球の半径を、Cylinder では height および radius フィールドでそれぞれ円柱の高さと円の半径を、Cone では height および radius フィールドでそれぞれ円柱の高さと底面の半径を指定できる。これらの基本オ

プロジェクトを図6に与えている(左から Box, Sphere, Cylinder, Cone).

オブジェクトの材質は<Appearance>ノードにおいて<Material>ノードを用いて指定する.<Material>ノードでは diffuseColor フィールドで表面の色を, transparency フィールドで透明度を指定できる.

X3D で文字を表示するためには, <Text>ノードを用いて, string フィールドに表示する文字を指定する. 文字の種類は<fontstyle>ノードで指定できる. テキストは透明な平面として実現されているので, ワールドの回転により向きが変わってしまうが, <Billboard>ノードで囲むことにより, 常に正面を向くテキストを実現できる.

上の基本オブジェクトの組合せにより複雑な図形が表現できるが, それらを<Group>ノードで囲むことにより1つの複合オブジェクトとすることができる. オブジェクト(または複合オブジェクト)に対して DEF フィールドにより名前をつけ, USE フィールドを利用することによりオブジェクトの複製ができる. USE による複製はオブジェクトのコピーを作成するには便利であるが, 色や大きさの変更はできない. そのような目的のためには<ProtoDeclare>ノードを用いて, プロトタイプとしてオブジェクトを定義し, <ProtoInstance>ノードを用いてインスタンスとして属性を変えたオブジェクトの複製が実現できる. このプロトタイプを用いた効率的なグラフ作成については後述する.

X3D で統計グラフを作成するためには, これらのオブジェクトを利用して構成すればよく, 図7に X3D で記述した 3D 散布図を Octaga player で表示した様子を示している. メニューの下にあるアイコンで Walk, Fly, Examine のモードを指定することにより, マウスのドラッグ

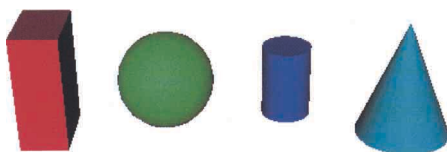


図6. X3Dの基本オブジェクト.

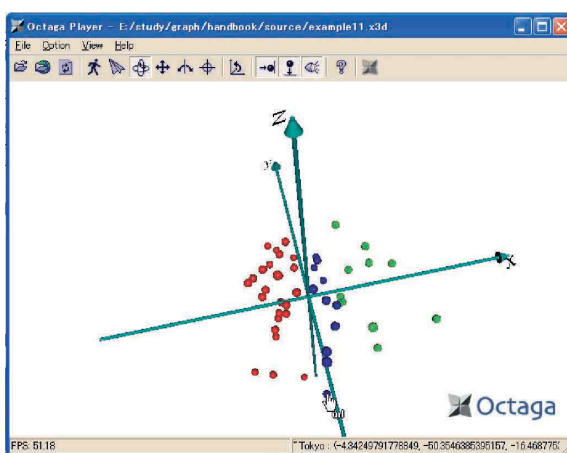


図7. 3D 散布図の例.

や矢印キーでのワールド内の移動や回転の仕方が規定される。3D 散布図を回転した場合には、Examine モードを利用し、拡大・縮小(中心に寄ったり離れたりする)には Walk モードを利用するとよい。

更に、X3D では、<IndexLineSet>ノードでワイヤフレームを用いた 3D 表示が、<IndexFace Set>ノードでワイヤフレームの表面部分を面とした表現が利用可能であり、関数により与えられる曲面の表現にはこれらのノードの利用が適している。また、地形の表現には<ElevationGrid>が適しているが Interactive 以上のプロファイルが必要であり、ワールドの実現に多くのメモリを必要とするために、通常の統計グラフを実現するためには、現時点ではあまり適していないと考えられる。

3. SVG による統計グラフ

すでに SVG についての概要を説明し、統計グラフが SVG を利用して構成できることを示したが、ここでは SVG の特徴を利用して Web 上でインタラクティブなグラフを作成する方法について説明を与える。更に、ここで説明するようなインタラクティブな統計グラフを、統計解析システム R で作成する関数について紹介する。

3.1 SVG によるインタラクティブ統計グラフ

SVG では、ベクターグラフィックスの利点の 1 つとして、レイヤーの処理(重ね描き)が可能である。統計グラフにおいては、レイヤーを用いることにより、データのプロットや解析結果を重ね合わせて表示できることが望ましい。SVG では、<g>要素によってレイヤー機能、すなわち、レイヤーの表示・非表示を切り替えるなどのインタラクティブな機能(イベント処理)が適用できる。

SVG グラフィックスに対するインタラクティブな機能は、JavaScript など(VBScript をサポートしているものもある)により DOM を介して実装することが可能である。簡単な例を以下に与える。

```
<svg width="300" height="300">
<title>Scripting Sample</title>
<script type="text/javascript"><![CDATA[
function clickHere(evt){
    alert(evt.target.nodeName);
}
]]></script>
<g id="button" onclick="clickHere(evt)"
    style="pointer-events:all;">
<circle id="c1" cx="150" cy="150" r="50"
    style="fill:pink; stroke:red;" />
<text x="150" y="150" style="text-anchor:middle;">
Click Here!</text>
</g>
</svg>
```

SVG の各要素には、キーボードやマウス操作、SVG 文書の読み込みなどのイベントを捕捉して、スクリプトを実行するためのイベントハンドラを実装できる。この例では、<g>要素に onclick イベントハンドラを実装しているため、グループ化された図形要素上でマウスボタンがクリックされると、clickHere 関数が実行される。この関数の引数 evt は、発生したイベントに関する情報を保持するオブジェクトである。target プロパティにより、イベントが発生した要素の参照が可能である。SVG においてよく用いられるイベントハンドラは、onmouse {down, move, out, over, up} のようなマウスイベントや、onload, onresize, onzoom のよう

な SVG 固有のイベントなどである。

マウスポインタの横に、マウスポインタが指している点に関する情報(散布図においては、データ点のラベルや値など)が表示される tooltip (ツールティップ)の機能を SVG において実現する方法について説明する。以下に基本的なコードを示す。

```
<svg width="500" height="500" onzoom="ZoomControl()" >
<script type="text/ecmascript"><![CDATA[
var svgdoc = svgDocument;
var svgroot = svgdoc.documentElement;
var tooltip = svgdoc.getElementById("tooltip");
function ShowTooltip(evt){
    ttrect=svgdoc.getElementById("ttrect");
    tttext=svgdoc.getElementById("tttext");
    tttext.childNodes.item(0).data=evt.target.getAttribute("id");
    ct=svgroot.currentTranslate;
    ttrect.setAttribute("x",evt.clientX-ct.x+10);
    ttrect.setAttribute("y",evt.clientY-ct.y-20);
    tttext.setAttribute("x",evt.clientX-ct.x+15);
    tttext.setAttribute("y",evt.clientY-ct.y-5);
    ttrect.setAttribute("width",tttext.getComputedTextLength()+10);

    tooltip.style.setProperty("visibility","visible");
}
function HideTooltip(){
    tooltip.style.setProperty("visibility","hidden");
}
function ZoomControl(){
    tooltip.setAttribute("transform","scale("+1/svgroot.currentScale+)");
}
]]></script>
<g id="data" style="pointer-events:all; stroke-width:1; stroke:black; fill:white"
    onmousemove="ShowTooltip(evt)" onmouseout="HideTooltip(evt)">
<circle cx="431.25" cy="97.1" r="3" id="x=685 y=533"/>
<circle cx="425" cy="142.6" r="3" id="x=680 y=498"/>
</g>
<g id="tooltip" style="visibility:hidden">
<rect x="0" y="0" width="70.5" height="20" id="ttrect"
    style="fill:#CCC; stroke:#000; opacity:0.85; stroke-width:0.5px;" />
<text x="5" y="20" id="tttext" style="fill:#000; font-size:14px;">dummy</text>
</g>
</svg>
```

データ点に関しては一部のみ示しており、軸やラベルについては省略している。スクリプトとして3つの関数、ShowTooltipとHideTooltip、ZoomControlを定義している。ShowTooltip関数は、id属性がdataとなっている<g>要素におけるonmousemoveイベントハンドラから呼び出され、マウスカーソルがデータ点上にある際に実行される。この関数は、まずtooltip用に予め用意された<text>要素と<rect>要素がそれぞれ、tttextとttrectとして取得される。tttextの子ノードに、テキストノードとして、イベントが発生した<circle>要素のid属性の内容、つまりイベントが発生したデータ点の座標がセットされる。次に、イベント発生時のマウスポインタの座標が、evt.clientXとevt.clientYで参照され、tooltipの構成要素の位置がこれに基づいて設定される。最後に、tttextの文字列の長さに基づいて、ttrectの幅を変更してから、tooltip全体を非表示から表示に変更(visibility属性をvisibleに設定)している。HideTooltip関数は、onmouseoutイベントハンドラから(つまり、マウスポインタがデータ点

から外に出たときに)呼び出され、tooltip 全体を非表示にする。ZoomControl 関数は、<svg>要素に記述されている onzoom イベントハンドラから呼び出され、SVG ビューアで拡大・縮小が実行された際に tooltip の大きさを一定に保つように、tooltip の transform 属性の値を現在の拡大率(svgroot.currentScale)に基づいて調整する。

統計グラフや地図上に関連情報を表示することは、視覚的な分析を行う上で重要な役割を果たす。しかし、同時に多くの情報を一度に表示するのは逆効果である。それゆえ、それらの情報を必要に応じてインタラクティブに表示したり、非表示にしたりできる機能が望まれる。SVG でこのような機能を実装するには、まず、表示・非表示を切り替える情報の種類ごとにレイヤーを作成(<g>要素でグループ分け)し、それぞれの<g>要素の style 属性に visibility プロパティを設定する。次に、JavaScript により、このプロパティを visible (表示)や hidden (非表示)に変更できる関数を記述する。そして、この切り替えを実現するためのユーザーインターフェースの作成や、イベントハンドラの記述を行う。このユーザーインターフェースは、SVG により作成する方法と HTML のフォームを利用する方法が考えられるが、HTML のフォームを利用する方が SVG のグラフを汎用的に用いることができる。ここでは、HTML のフォームを利用した例について紹介する。

ここで用いる例は、R の cluster パッケージの votes.repub データをユークリッド距離に基づく非類似度によりクラスタリング(*k*-means 法)を用いて 2つのクラスターを構成した。clusplot 関数で作成したこのグラフ(図 8)は、データに対する第 1、第 2 主成分得点を 2次元の座標平面上にプロットし、2つのクラスターをそれぞれ異なるシンボル(○, △)で表している。各点のケースラベルが tooltip として表示される機能および、全データ点のデータラベル、クラスターを示す楕円領域の表示・非表示などの処理を、JavaScript により実装することで、Web ページで実現できるインタラクティブ統計グラフとなっている。

このグラフは、レイヤーの切り替えを行うインターフェースとして HTML のフォーム(control.html)を採用しており、フレームによってグラフを表示する SVG の部分(cluster.svg)と切り離されている。control.html には、ラベルおよびクラスター領域の表示・非表示を切り替えるためのチェックボックスを用意している。

```
<form name="main">
<input type="checkbox" name="label" onclick="parent.parent.setVisibility('label')">
  Label on/off</input> <br/>
<input type="checkbox" name="clustArea"
  onclick="parent.parent.setVisibility('clustArea')">Cluster Area on/off</input>
</form>
```

cluster.svg においては、楕円領域の<ellipse>要素が、id 属性が clustArea である<g>要素の子要素として、また、データラベルの<text>要素が id 属性が label である<g>要素の子要素として記述されている。表示・非表示を切り替えるためのスクリプトは次の通りである。

```
function init(){
  parent.parent.setVisibility = setVisibility;
}
function setVisibility(id){
  sty = svgDocument.getElementById(id).style;
  curvb = sty.getPropertyValue('visibility');
  if(curbv=='hidden') sty.setProperty('visibility','visible');
  else sty.setProperty('visibility','hidden');
}
```

cluster.svg がロードされると、最初に init 関数が実行され、その中で setVisibility 関数が、index.html の文書に関連付けられる。これは、SVG と HTML フォームを組み合わせ

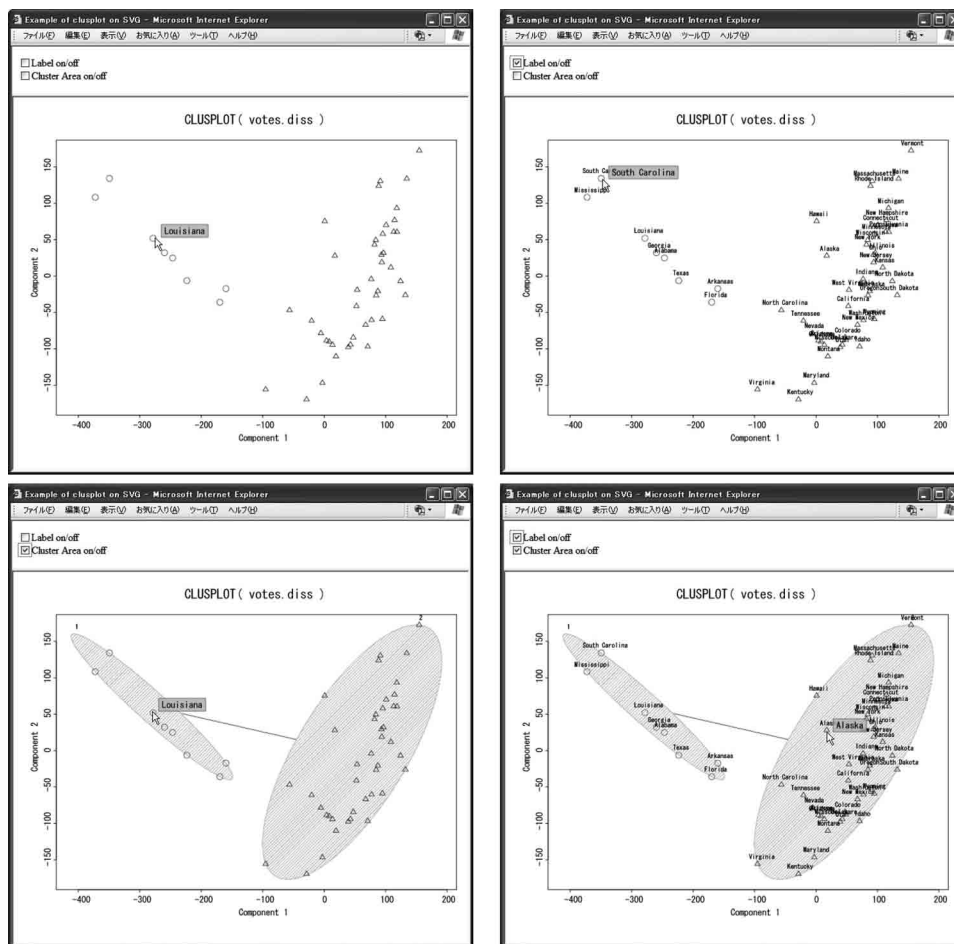


図 8. Web ページでのインタラクティブな SVG グラフ.

利用する際に必要な処理であり、これによって、control.html から、この関数を呼び出すことができるようになる。setVisibility 関数は、control.html でクリックされたチェックボックスの id 属性の内容を受け取り、これに対応する<g>要素の visibility プロパティを hidden から visible、もしくはその逆に変更する。図 8 は各レイヤーが切り替わる様子を示している。

3.2 SVG による統計グラフの作成ツール

SVG によりインタラクティブ統計グラフが実現できるとしても、統計グラフを SVG の記述方法に従って全て記述するのは困難であり、また面倒でもある。SVG はテキストエディタで作成しなくても、Adobe Illustrator や CorelDraw などの画像作成ソフトウェアで画像を SVG 形式で出力でき、フリーのオフィスアプリケーションである OpenOffice、グラフ作成ソフト gnuplot をはじめ、商用のグラフ作成ソフトウェアも SVG 形式の出力をサポートしているものがある。

統計解析ソフトウェア R は、様々な統計グラフを作成できるが、T Jake Luciani によりリリースされたライブラリ RSvgDevice (<http://www.darkridge.com/~jake/RSvg/>) を用いること

により、PostScript や PDF などのようなデバイスと同様に、SVG をグラフィックデバイスとして利用でき、R で作成可能なグラフを SVG 形式で出力することが可能である。SVG による対散布図をテキストエディタで全て記述することは現実的ではないが、RSvgDevice を利用すれば、次のような数行のコマンドで実現できる。

```
> devSVG(width=8,height=8)
> plot(LifeCycleSavings,oma=c(10,10,10,10))
> title("Life Cycle Savings: Data on the savings ratio 1960-1970.")
> dev.off()
```

このコマンドにより得られた SVG の出力が図 4 である。他のデバイスに出力するコマンドを入力する労力とほとんど変わらない上、図 5 に示すような拡大・縮小および移動が SVG ビューアの機能により利用できる。

SVG のエキスパートにとっては、RSvgDevice によりグラフを SVG として作成したものを、前述の手順によりグループ化し、スクリプトとしてインタラクティブな機能を加えたりするなど、更なる SVG のメリットを享受できる。実際、図 8 のインタラクティブグラフは、cluster パッケージの `clusplot` 関数による作図を、RSvgDevice により SVG として保存し、そのファイルに対しインタラクティブグラフとしての機能を追加することで実現している。

RSvgDevice は描画情報を単に SVG 形式で出力するだけなので、インタラクティブな統計グラフとするには、スクリプトの追加や、要素のグループ化や id 付けなど、多くの作業が必要となる。このような作業は、JavaScript や DOM の知識がない者にとっては、敷居が高いと感じるであろう。そのような問題を解決するために、我々は、R においてインタラクティブ機能を持つ SVG 形式の基本的な統計グラフを出力するためのコマンドを提供する RInG (R Interactive Graphics) ライブラリを開発した。現在、ソースコードと Windows 版のバイナリパッケージが Web サイト <http://www.fwu.ac.jp/fujino/Xg4stat/SVG/> で入手できる。R にパッケージをインストールし、パッケージをロードすることで利用できる。

RInG ライブラリでは、現在、3 種類の基本的なインタラクティブな統計グラフ(散布図、ヒストグラム、ボックスプロット)を出力するための関数を提供している。`iplot` は 2 次元散布図を出力するための関数である。この関数で提供される散布図では、データ点上にマウスカーソルを合わせるとデータ点の値を tooltip 上に表示するとともに、データ点から x 軸上と y 軸上への補助線を表示する。`iplot` による作図は以下のように行う。上のコマンドは散布図に目盛りの補助線をグリッドとして示すグラフ(図 9)を作成し、下のコマンドにより垂直線プロットが作成される。

```
> iplot(cars$speed, cars$dist, col="blue", grid=T,
main="Speed and Stopping Distances of Cars")
> iplot(BJsales,type="h",col="red", main="Sales Data with Leading Indicator")
```

`ihist` はヒストグラムのバーの上にマウスカーソルを合わせたときに、そのバーの階級に関する上限値、下限値、階級値、個体数を tooltip に表示するようなヒストグラムを作成する関数である。

```
> ihist(rnorm(1000),breaks="FD")
```

`iboxplot` は、ボックスプロットの箱の上にマウスカーソルを合わせたときに、そのグループに対する 5 数要約値が tooltip に表示され、外れ値のデータ点の上にマウスカーソルを合わせると、データ番号が tooltip に表示されるようなボックスプロットを作成する関数である。以下のコマンドによって作成されたボックスプロットが図 10 である。

```
> attach(InsectSprays)
```

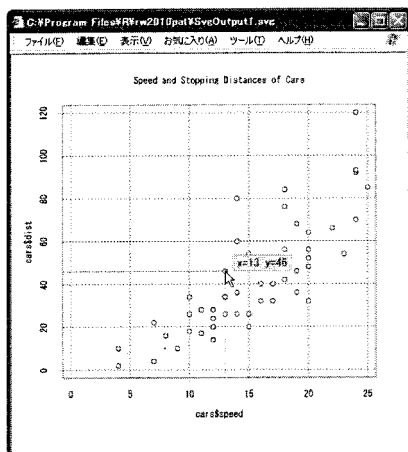


図 9. iplot 関数による散布図.

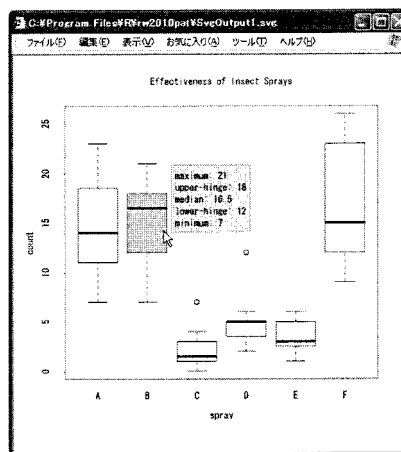


図 10. iboxplot 関数によるボックスプロット.

```
> iboxplot(split(count,spray),xlab="spray",ylab="count",
main="Effectiveness of Insect Sprays")
```

4. X3D による統計グラフ

すでに X3D についての概要について説明し、3D 表示の統計グラフを X3D を利用して作成できることを示した。X3D で実現できる統計グラフについては、山本 他 (2001) で X3D の前身である VRML を用いたグラフ表現について紹介しており、その方法は X3D についても同様に実現できる。ここでは、いくつかの X3D によるグラフ表現について紹介し、X3D による 3D 散布図を R で作成する関数について紹介する。

4.1 X3D による 3D 統計グラフ

3D モデリング言語を利用して 3D 統計グラフを作成する利点は、データを 2次元表現するためのプログラミングの必要がないことである。3次元の座標をそのまま利用できることは、グラフ作成の負担を軽減し、統計解析システムにおける 3次元データの視覚化が容易に実現できることを意味する。

図 11 は、3D 散布図で用いる軸であり、円錐と円柱で矢印を作っている。3つの軸を配置するため、<Group>ノードの DEF フィールドに myAxis と名前をつけ、2つ目と3つ目の軸については、<Shape>ノードで USE フィールドに myAxis を指定することで複製が利用できる。

```
<Group DEF='myAxis'>
  <Shape>
    <Appearance DEF='ARROW_APPEARANCE'>
      <Material diffuseColor='0 .7 .7' transparency='0' />
    </Appearance>
    <Cylinder radius='0.04' height='10.2' />
  </Shape>
  <Transform translation='0 5 0'>
    <Shape>
      <Cone bottomRadius='.15' height='.3' />
      <Appearance USE='ARROW_APPEARANCE' />
    </Shape>
  </Transform>
```

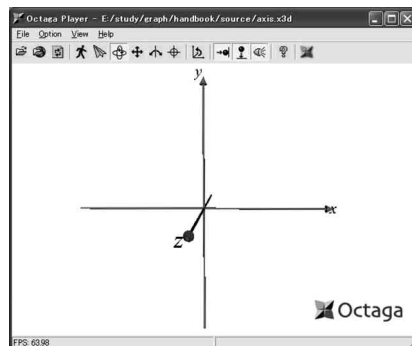


図 11. X3D による 3D 散布図用の座標軸.

```

</Group>
<Transform rotation='0 0 -1 1.57'>
  <Shape USE='myAxis' />
</Transform>

```

軸のラベルは、テキストの中身が異なるので、DEF, USE は使えないため、プロトタイプを利用する。以下に示すコードで軸ラベルを実現している。

```

<ProtoDeclare name='label'>
  <ProtoInterface>
    <field name='pTranslation' type='SFVec3f' value='0 0 0' accessType='initializeOnly' />
    <field name='pText' type='MFString' value='' accessType='initializeOnly' />
    <field name='pFSize' type='SFFloat' value='0.7' accessType='initializeOnly' />
    <field name='pFColor' type='SFColor' value='0 0 0' accessType='initializeOnly' />
    <field name='pFStyle' type='SFString' value='ITALIC' accessType='initializeOnly' />
    <field name='pFJustify' type='MFString' value='MIDDLE' accessType='initializeOnly' />
  </ProtoInterface>
  <ProtoBody>
    <Transform>
      <IS> <connect nodeField='translation' protoField='pTranslation' /> </IS>
      <Billboard>
        <Shape>
          <Appearance>
            <Material>
              <IS> <connect nodeField='diffuseColor' protoField='pFColor' /> </IS>
            </Material>
          </Appearance>
          <Text>
            <IS> <connect nodeField='string' protoField='pText' /> </IS>
            <FontStyle family='SERIF'>
              <IS>
                <connect nodeField='size' protoField='pFSize' />
                <connect nodeField='style' protoField='pFStyle' />
                <connect nodeField='justify' protoField='pFJustify' />
              </IS>
            </FontStyle>
          </Text>
        </Shape>
      </Billboard>
    </Transform>
  </ProtoBody>
</ProtoDeclare>

```

```

</Transform>
</ProtoBody>
</ProtoDeclare>
<ProtoInstance name='label'>
  <fieldValue name='pTranslation' value='5.2 -0.2 -0.2' />
  <fieldValue name='pText' value='"x" />
</ProtoInstance>

```

<ProtoDeclare>は2つのパートからなり、<ProtoInterface>ノードでは、プロトタイプとして用いたときに、指定できるフィールドについて<field>ノードにおいて名前(name)、型(type)、デフォルト値(value)を宣言している。また<ProtoBody>ノードにおいて、プロトタイプの中身を設定している。設定されたプロトタイプを利用するには、<ProtoInstance>ノードのnameフィールドにプロトタイプ名を指定し、指定したいフィールドを<fieldValue>ノードで指定する。指定しなかったフィールドについてはデフォルト値が用いられる。

この座標軸に、Sphere や Box などの X3D オブジェクトをデータ点として配置することにより、3D 散布図が実現可能である。この場合にもデータ点をプロトタイプとして作成し、色などを変更して表示するとよい。図7はそのようにして作成した3D 散布図である。

X3D ではタッチセンサーなどのセンサによりマウスのクリックやオブジェクトへの接近などに応じてイベントを生成する。このイベントを利用して、色や位置の変更などのアニメーションを設定することができる。図7の散布図では、Anchor ノードを用いて、点にマウスをあわせると、ケース名および座標が右下のステータスバーに表示される機能を実装している。

X3D の特徴を用いることにより、新しいグラフ表示の可能性が提案できる。図12は、標準化したデータの3D 散布図に、標準偏差を半径とした半透明な球を配置することにより、中心から外れたデータ点を認識しやすくした散布図である。このグラフは、ビューアの機能により回転や拡大・移動ができる。図13は、第2主成分までの主成分得点による2次元散布図上に、階層型クラスタリングのデンドログラムを配置したグラフ表示である。6クラスターを色分けすることにより、全体的なクラスター構成がわかるとともに、各クラスター内の結合具合についても考察できる。興味のあるクラスターに関して拡大することにより、クラスターの全体と詳細について確認できる。

更に、Script ノードを利用して、Java や JavaScript (ECMAScript) によりイベントをプログラミングすることもできる。本論文の SVG グラフで実現しているような、HTML のフォームを用いて Web ブラウザをユーザインタフェースとするような使い方は、X3D の DOM に相当する Scene Access Interface (SAI) を利用して JavaScript などでプログラム可能であるが、現在

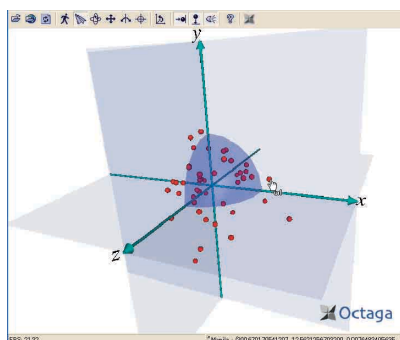


図 12. 3D 散布図への1標準偏差球の配置。

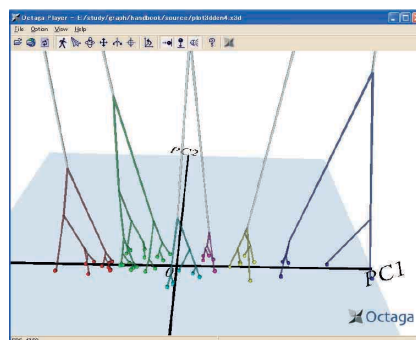


図 13. 主成分得点プロット上の3D デンドログラム。

のところ SAI は Flux Player のみが対応しているにすぎないため、まだ本格的に利用できる環境は整っていない。しかし、2006 年 8 月に Web ベースの 3D アプリケーションを X3D により実現するためのフォーラム (<http://www.ajax3d.org/>) が立ち上がり、Web における 3D アプリケーションを実現するための方法について多くの例が示されるようになってきており、今後に期待がもたれるところである。

4.2 X3D による散布図作成のための関数

SVG の統計グラフと同様に、X3D による統計グラフについても、一から作るのは大変な作業である。X3D のオーサリングツールは Flux Studio や、フリーの Seamless3d などがあるが (最新のオーサリングツールについての情報は Web3D のサイトを参照のこと)、そのほとんどはグラフを作成することを目的としているわけではなく、3D シーンを作成するためのツールである。

著者らは、R において 3D 散布図を作成する関数を作成しており、Web サイト <http://stat.sm.utokai.ac.jp/~yama/x3d/> で公開している。この関数 `x3dplot3d` は、座標の標準化を行うか否か (デフォルトは標準化を行わない)、フレーム・座標軸・ケースラベルの表示・非表示を制御する引数を指定できる。この関数で作成された 3D 散布図では、点にマウスを合わせるとケース名および座標をブラウザのステータスバーに表示する機能をもっている。

```
> source("x3dplot3d.r") # 関数の読み込み
> x3dplot3d(cityecon2, "test1.x3d", frame=T, axis=T, label="axis")
> x3dplot3d(cityecon2, "test2.x3d", scale=F, axis=T)
```

またこの関数は、`colored=T` オプションを付け、データ行列として 4 列目にグループを示す変数を与えることにより、グループ別の塗り分け散布図を作成できる。

```
> x3dplot3d(cepc3, "cepc3.x3d", scale=T, axis=T, label="axis", colored=T)
```

図 7 は、R で主成分分析やクラスター分析した結果 (データフレーム `cepc3` は、最初の 3 列が第 3 主成分までの主成分得点、第 4 列はクラスター分析の結果の 3 クラスターを示す変数からなる) に対して、この関数を用いて作成したものである。

5. インタラクティブテキストのための XML

XML ベースの SVG や X3D を統計グラフとして作成するメリットは、すでに示している。ここでは、この性質を利用した XML ベースのテキスト (教材) のプロトタイプを示し、インタラクティブテキストの可能性について言及する。そのために、まず教育用に利用することが可能で、なおかつ効果的と思われるティーチウェアを SVG により作成することについて紹介し、それらを用いたインタラクティブテキストについて紹介する。

5.1 SVG によるティーチウェア

統計教育用のティーチウェアとして、インタラクティブな機能をもった Web ページやツールが数多く公開されている (Project CASE, <http://case.f7.ems.okayama-u.ac.jp/statedu/> など)。Web を利用するものの多くは現在では Flash による教材が多く、Java によるものも見られる。Flash はオーサリングツールが充実しているため、このようなアプリケーションを手軽に作成するためには最適の選択と思われるが、市販のオーサリングツールを使わずオープンな規格を利用してのツール作成を考える場合には SVG という選択肢がある。SVG はテキストファイルであることから、作成されているティーチウェアにおける表記などを他言語に翻訳して利用したい場合などは、直接エディタでファイルを編集すればよく、特別なツールを必要としない利

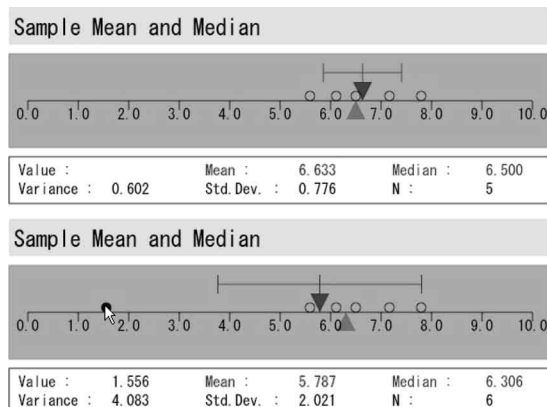


図 14. 平均と中央値に関するティーチウェア.

点がある.

ティーチウェアの実現例として, CASE プロジェクトで Flash により実現されている「平均と中央値の違いを理解する」ための SVG 版を図 14 に与えた. 座標軸をクリックすることにより, データ点が追加され, 同時に全てのデータ点の平均および中央値が表示されるというティーチウェアである. このように, SVG ではグラフばかりでなく, アプリケーションを作ることも可能である.

5.2 インタラクティブテキスト

すでに紹介しているように, SVG や X3D により XML ベースのベクターグラフィックスが作成でき, Web ページにおける表現力の向上が期待できる. 単に, 画像としてだけではなく, テキストの一部としてドキュメントと関連させることにより, インタラクティブテキストとして, より対話的な機能が可能である.

現在, PDF や HTML で, 統計に関するテキストが多く公開され, Java や Flash を用いた動的教材を含んだサイトもいくつか見受けられるが, このような Web 上で公開している教材に, インタラクティブなグラフやティーチウェアを用いることにより, Web 上でのインタラクティブなテキストが作成可能である.

Web 上で統計のテキストを提供する際の問題点の 1 つに数式の扱いがある. 現在は, 画像で数式を表示するか, または PDF ですべて一つのページを公開するという方法が多い. XML を利用した方法としては, XML のボキャブラリである MathML により数式を記述し, HTML に組み込むことにより数式を表示することが可能である. MathML にもとづく数式は, いくつかの数式エディタや Mathematica などで作成可能である.

ここで, SVG によるグラフと MathML による数式表示を含んだインタラクティブテキストの例を示す. 図 15 は, 正規母集団 $N(\mu, \sigma^2)$ の母平均 μ の信頼区間を説明するテキストと, 実際のシミュレーション結果のグラフが表示されている.

このテキストは, 文書は XHTML, 数式は MathML, 図は SVG により記述され, 全体としての XML テキストにより構成されている. この図は, 各信頼区間を示す水平の直線にマウスをあてることにより, その信頼区間の値についての情報を得ることができるインタラクティブな機能がある. 以下に, XML テキストのソースコードの一部を示す.

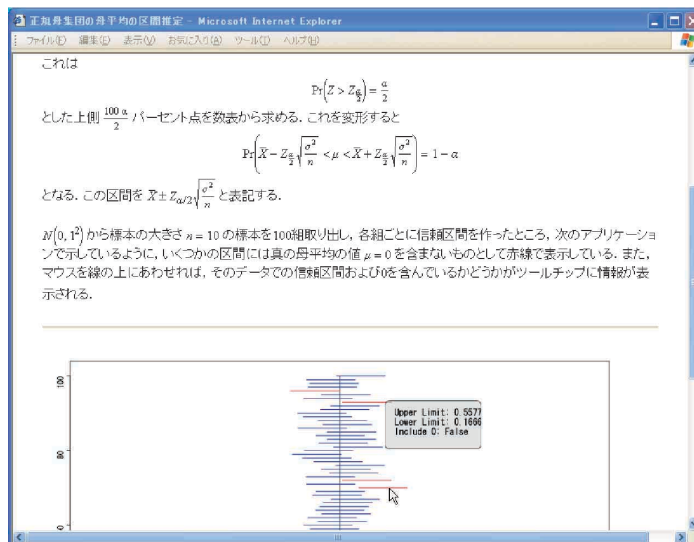


図 15. 母平均の信頼区間についてのテキスト.

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="http://www.w3.org/Math/XSL/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0 plus SVG 1.1//EN"
"http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>正規母集団の母平均の区間推定</title>
</head>
<body style="background-color:#ffffff;">
<p>母平均
<math xmlns="http://www.w3.org/1998/Math/MathML" style="vertical-align:middle;">
<mrow><mi>&mu;</mi></mrow></math>
の区間推定</p>
(略)
<embed id="conf" src="conf.svg" type="image/svg+xml" width="800" height="600" />
(略)
```

この文書は XHTML であることを, 5 行目でネームスペース (xmlns) により指定しており, 11 行目は数式を $タグ内$異なるネームスペースであることを示している. また 2 行目でブラウザで数式を表示するためのスタイル変換の指定を行っている. 図は <embed>タグにより, 別に作成されている SVG によるティーチャウェア (conf.svg) を取り込むように記述されている.

このページを表示するためには, Internet Explorer では, MathML が表示できる techexplorer などのプラグイン, SVG が表示できる Adobe SVG Viewer 3.0 などのプラグインが導入されている必要があるが, Firefox のバージョン 1.5 以降においては, MathML と SVG の表示を標準でサポートしており, Web ブラウザだけで, このような軽量かつ高機能なドキュメントを表示できる環境も整備されつつある. このようなテキストの実現方法は, 新たなオンラインテキストの候補となると考えられる.

次に, サーバアプリケーションとして作成したティーチャウェアについて紹介する. 図 16 は, 鉛筆の長さを測りその結果を度数分布表として作成するティーチャウェアである. このアプリ

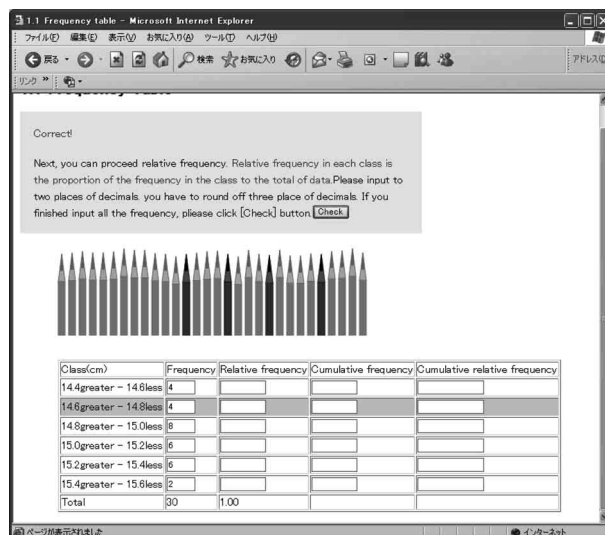


図 16. 度数分布作成のティーチウェア.

ケーションでは、ユーザが、度数分布表のある階級にマウスカーソルを合わせると、その階級に対応する鉛筆が強調表示され、ユーザは鉛筆の本数を数え度数分布表に度数を入力し、度数分布表の区間に対する度数、相対度数、累積度数、累積相対度数を計算、入力した後、正解かどうか確認するという対話的な作業を行う。

鉛筆の長さは、このページが開かれた際にサーバにおいて乱数を用いて決定される。乱数の生成と度数分布表の作成(最適な階級数とそれに伴う階級の決定)は、Web サーバー上で動作している R を統計エンジンとして、サーバにおいて処理されている。R の計算結果に基づいて、SVG で描画されている鉛筆の長さおよび、HTML による度数分布表の入力フォームの階級が決定される。鉛筆の強調表示や、度数分布表のフォームに入力された値のチェックなどの動作は、SVG や HTML 内に記述された JavaScript によるものである。このような SVG を用いた Web アプリケーションとして、GIS において任意の地域や地点に関する情報を表示させる機能として実現した例が、亀川 他(2007)で詳細に説明されている。

この他、Web による XML の利用に関して、Geroimenko and Chen (2004)では、マルチメディア利用のための SMIL (Synchronized Multimedia Integration Language, W3C, 2005b)を用いて、SVG, X3D, SMIL を用いたマルチメディア対応の応用例が提案されており、その他にも SVG と X3D の利用について多くの実用例が与えられている。

6. 終わりに

本論文では、XML ベースのグラフィックスフォーマットである SVG および X3D を用いたインタラクティブグラフについて、作成方法および活用方法を紹介した。本論文で紹介したインタラクティブグラフの例や、インタラクティブグラフィックスのライブラリの他、Web 上での統計グラフ生成ツール、GIS と連携した統計データ表示環境(Fujino et al., 2004)などについては、Web ページ(<http://www.fwu.ac.jp/fujino/Xg4stat/>)に公開している。このサイトは、SVG および X3D についてのチュートリアル的な役割を与えており、統計グラフなどのテンプレ

レート(雛形)を提供しているのです, XML ベースのグラフィックスフォーマットの理解の参考となれば幸いです。

謝 辞

本論文の審査員の先生方には, 非常に丁寧な査読のうえに, 不備な点のご指摘と有益なご意見を多く頂戴しました。ここに記して謝意を表します。

参 考 文 献

- DandD Project (2004). DandD プロジェクト IV, <http://www.stat.math.keio.ac.jp/DandDIV/index.ja.html>.
- Data Mining Group (2005). Predictive Model Markup Language (PMML) 3.1, <http://www.dmg.org/>.
- Eisenberg, J. D. (2002). *SVG Essentials*, O'Reilly, Sebastopol.
- Fitzgerald, M. (2004). *XML Hacks*, O'Reilly, Sebastopol. (佐藤直生 監訳, 木下哲也, 有限会社 福龍興業 訳(2005). 『XML Hacks — エキスパートのためのデータ処理テクニック』, オライリージャパン, 東京).
- Fujino, T., Yamamoto, Y. and Tarumi, T. (2004). Possibilities and problems of the XML-based graphics in statistics, *COMPSTAT2004 Proceedings in Computational Statistics*, 1043–1052.
- Geroimenko, V. and Chen, C. (eds) (2004). *Visualizing Information Using SVG and X3D*, Springer, London.
- 亀川佳美, 藤野友和, 垂水共之(2007). 小地域データにおける地理統計解析と SVG による可視化, *統計数理*, **55**(1), 85–100.
- Meyer, D., Leisch, F., Hothorn, T. and Hornik, K. (2004). StatDataML: An XML format for statistical data, *Computational Statistics*, **19**(3), 493–509, <http://www.omegahat.org/StatDataML/>.
- Mori, Y., Fujino, T., Yamamoto, Y. and Tarumi, T. (2004). XML-based applications in statistical analysis, *Proceedings of Interface 2004: Computational Biology and Bioinformatics, 36th Symposium on the Interface*.
- 大津起夫(2004). 統計メタデータ記述の技術動向と XML 文書の利用, *大学入試センター研究紀要*, **33**, 65–88.
- Symanzik, J. (2004). Interactive and dynamic graphics, *Handbook of Computational Statistics—Concepts and Methods—* (eds. J. E. Gentle, W. Härdle and Y. Mori), Springer, Berlin, Heidelberg.
- 立川敬行(2004). 『XML 徹底入門—マルチメディア応用からセマンティック Web まで』, 電波新聞社, 東京.
- W3C (1998). Document Object Model (DOM) Level 1 Specification Version 1.0, W3C Recommendation 1 October, 1998, <http://www.w3.org/TR/REC-DOM-Level-1/>.
- W3C (2003a). Mathematical Markup Language (MathML) 2.0, W3C Recommendation 21 October 2003, <http://www.w3.org/TR/MathML2/>.
- W3C (2003b). Scalabel Vector Graphics (SVG) 1.1, W3C Recommendation 14 January 2003, <http://www.w3.org/TR/SVG11/>.
- W3C (2003c). XML 1.0 (3rd ed.), W3C Recommendation 04 February 2004, <http://www.w3.org/TR/xml/>.
- W3C (2004). XML 1.1, W3C Recommendation 04 February 2004, <http://www.w3.org/TR/xml11/>.
- W3C (2005a). Scalabel Vector Graphics (SVG) 1.2. W3C Working Draft 13 November 2003, <http://www.w3.org/TR/SVG12/>.

- W3C (2005b). Synchronized Multimedia Integration Language (SMIL) 2.1, W3C Recommendation 13 December 2005, <http://www.w3.org/TR/SMIL2/>.
- Web3D(2004). Extensible 3D(X3D). ISO/IEC 19775:2004, <http://www.web3d.org/x3d/specifications/>.
- 山本義郎, 藤野友和(2005). 統計学における XML の利用, 日本計算機統計学会, **18**(2), 165–178.
- 山本義郎, 垂水共之, 佐藤義治(2001). VRML を利用した統計グラフ表現, 統計数理, **49**(2), 293–303.
- 横内大介, 柴田里程(2001). インターデータベース—DandD インスタンスのエージェント化, 統計数理, **49**(2), 317–331.

XML-based Interactive Statistical Graphics
— Practical Use to Web-based Statistical Environments —

Yoshiro Yamamoto¹, Tomokazu Fujino² and Masaya Iizuka³

¹Department of Mathematics, Tokai University

²Department of Environmental Science, Fukuoka Women's University

³Graduate School of Environmental Science, Okayama University

We propose graphical representations on the Web using XML technology: SVG for 2D graphics and X3D for 3D graphics.

Both SVG and X3D graphics can be displayed within Web browsers such as Internet Explorer. This makes it possible to realize interactive graphics on a Web-based interactive textbook or a Web-based statistical analysis package. This paper describes the potential applications of XML, SVG and X3D and then introduces graphical representations of SVG and X3D graphics. It also introduces authoring tools for producing interactive SVG and X3D graphics. To demonstrate the advantages of the XML-based graphics format, we propose development of a Web-based textbook with interactive graphics.