

SVG 地図を対象とした表現要素の分類とそのスキーマ抽出

藤森 史生[†] 有川 正俊[†] 森田 喬[‡]

[†] 東京大学空間情報科学研究センター 〒153-8904 東京都目黒区駒場 4-6-1

[‡] 法政大学工学部土木工学科 〒184-8584 小金井市梶野町 3-7-2

E-mail: [†] {fujimori, arikawa}@csis.u-tokyo.ac.jp, [‡] morita@k.hosei.ac.jp

あらまし 空間的な位置関係や広がりやを人に伝えるためには、視覚的な表現が効果的である。視覚的な表現は、その伝え方や組み立て方により、分かりやすさに違いがある。一般に、視覚的な表現で人に伝えたい要素は、(a)類似性・相違性、(b)順序、(c)割合の3つに集約されると言われている。ラスターデータに比べて、ベクトルデータを用いた表現は、視覚的な伝達性（視覚伝達性）に関係する要素を比較的容易に定量化することが可能である。

これらを背景に、地図を表現する SVG データを対象に、類似性・相違性、順序、割合に関係する属性を読みとり、定量化することでグラフやテーブルとして表示させ、より分かりやすい視覚的な表現へと改善するツール“Map Checker”の開発を行っている。このツールは視覚的な表現の要素分類ともいえ、地図に関しては凡例の自動抽出へと応用できる。本論文では、地図表現を中心とした SVG を対象に、表現要素である *path element* の属性を分類し、地図の表現要素の定量化と視覚化の枠組みを提案し、プロトタイプシステムの紹介を行う。

キーワード データの可視化, XML, ユーザインタフェース, 半構造データ, スキーマ抽出, 地図学

Classification based on Representational Elements on SVG Maps and Extraction of their Schema

Fumio FUJIMORI[†] Masatoshi ARIKAWA[†] and Takashi MORITA[‡]

[†] Center for Spatial Information Science, University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8904 Japan

[‡] Department of Civil Engineering, Hosei University 3-7-2 Kajino-cho, Koganei-shi, Tokyo 184-8584, Japan

E-mail: [†] {fujimori, arikawa}@csis.u-tokyo.ac.jp, [‡] morita@k.hosei.ac.jp

Abstract Visual representations are effective for people to communicate spatial location and spread. It detects the effect of visual representations that how to represent and compose their intentional messages. Generally speaking, the semantic relations in visual communication are Nominal, Ordinal and Quantitative.

Based on this background, we are developing the visual tool, called "Map Checker" for improving SVG formatted maps. This tool provides the classification based on representational elements and the extraction of their database. This paper introduces some basic principles of the classification based on representational elements and our proposed visual tool.

Keyword Visualization of Data, XML, User Interface, Semi-Structural Data, Extraction of Schema, Cartography

1. はじめに

空間的な位置関係や広がりやを人に伝えるためには視覚的な表現が効果的である。図（グラフィックス）を描く目的は、視覚的な表現を通して、相手に視覚記号間の関係を伝えることにある。しかしながら、視覚記号間の関係の組み立て方や伝え方によって伝わりやすさ、分かりやすさ（視覚伝達性）に違いがある^[4]。コンピュータが普及する以前は、技術と環境を備えた専門家が図を作成していた。そのため、視覚伝達性が十分に考慮されたものであったが、SVG(Scalable Vector Graphics)^[1]をはじめとしてコンピュータによる図の作成環境が整うにつれて、専門家ばかりでなく一般の利用者も図を作成するようになり、不適切な表

現（関係性が十分に表現されていない）を含む視覚伝達性の低い図が流通しつつある。このような現状に対して、我々は視覚伝達性に関して十分考慮できる環境を提供する必要性を感じている。

Web 上での図の標準記述言語として GIF や JPEG に代わって SVG が普及し始めている^[1]。SVG は、XML に準拠しておりテキストで構造記述されているため、コンピュータは、ベクトルを基本とする図の内部構造を比較的容易に解釈することができる。既存の環境では、コンピュータが、SVG データを表示のために解釈するのに対して、人は視覚記号間の関係を理解するために図を読む。すなわち、コンピュータを人の理解に近づけるためには、SVG データから視覚伝達性に関係

する表現要素を抽出し、コンピュータが視覚記号間の関係を解釈するためのデータに加工する必要がある。

このような背景をもとに、図の視覚伝達性の向上を目的として、SVG データから視覚伝達性に関する表現要素を抽出・分類し、利用者が視覚伝達性を意識して改善するための利用環境の枠組みとその実現例として新しいツールを提案する。具体的には、地図を表現する SVG データを対象に、コンピュータが視覚伝達性に関する属性を読みとり、視覚伝達性の観点から集計あるいは換算し、グラフやテーブルとして視覚化することで、人が図を直接見るだけでは分かりにくい視覚伝達性に関する情報を容易に理解し、改善することに利用できるツール“Map Checker”を開発している。このツールが提供する最大の機能は、SVG データで表現される図から視覚的表現要素を分類できる点である。この機能は、SVG データで使用される様々な視覚記号の種類を特定し、さらにその視覚記号の特性を定量化することにより、SVG データから視覚記号に対応した意味のある属性とその値、つまりデータベースを抽出する。この機能の重要な一例として、SVG で記述された地図 (SVG 地図) からの凡例の自動抽出がある。

2. 図の記号学

この節では、視覚伝達性を評価するために必要な論理として、フランスのビジュアルコミュニケーションの専門家であるジャック・ベルタンの図の記号学^[4]から、グラフィック関係と視覚変数について紹介する。

2.1. グラフィック関係

ジャック・ベルタンによれば、視覚伝達性の観点からすると、図の目的は、論理の可視化となる。すなわち図で人に伝えたい意味内容は、視覚記号間の以下の3つの関係となる^[4]。

- (a) 類似性・相違性
- (b) 順序
- (c) 割合

それぞれの関係は、視覚記号間の違い (クラス) を表している。具体的には、(a)類似性・相違性は、単にクラスが分離していることを、(b)順序は、相対的なクラス間の違いを、(c)割合は量的にクラス間の違いを意味づけている。

2.2. 視覚変数

図は、類似性・相違性、順序、割合の様々な関係を表現することができるが、そのための表現手段として、視覚記号間において人が知覚できる8つの変化を利用

する^[4]。図1は、平面の構成方式 (点, 線, 面) に8つの視覚変数を適用したものである。

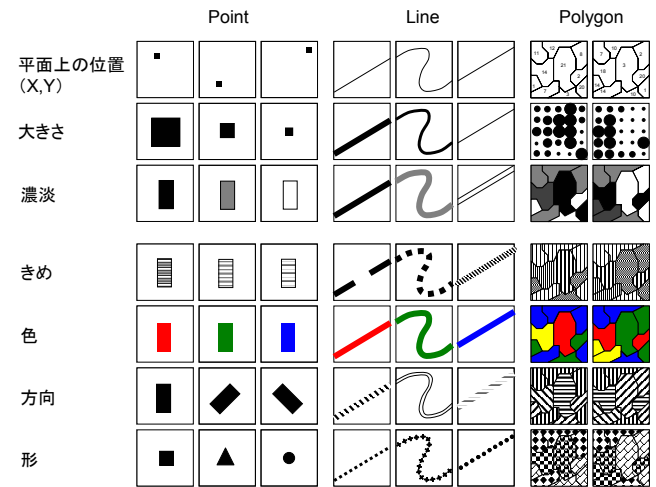


図1. 視覚変数

視覚変数である平面上の位置 (x と y), 大きさ, および濃淡は, (b)順序 または (c)割合を表すことができ、並び替えを行うことができる。この4つの変数をまとめて図像の変数とする。図像とは、人が一瞬にして視覚記号間の関係を知覚できる意味のある形である。視覚変数であるきめ, 色, 方向および形は, (a)類似性・相違性のみを表すことができ、ある視覚記号を他の視覚記号と分離させるだけである。これら4つの視覚記号を分離の変数とする。

3. SVG の表現要素の分類

この節では、図の記号学にしたがって、視覚変数と SVG の表現要素との対応を明らかにし、システムに適用するための枠組みを紹介する。

表1は、SVG の *path element* と視覚変数との対応表である。SVG には、基本的な図形として、*line, rect, circle, polygon, polyline* 等の *element* が定められている^[1]が、*path element* を用いることでそれら全ての図形を表現することができるため *path element* に関する説明とする。視覚変数と *path element* との対応は *path element* の座標のための属性である (*d: pathdata*), 塗り操作のための属性である (*fill, stroke*) を基本的に用いている。

人は複雑な処理を行うことで、視覚変数を知覚しており、コンピュータの解釈を人の理解に近づけるためには、段階別の処理が必要となる。まず、視覚変数に関する *path element* の属性を読みとり、次に、*path element* の属性に簡易な処理を加え、視覚変数に部分的に対応づける。さらに *path element* の属性に対してより高次の処理を加えることで、人の理解に近いものとして対応づける。これらの段階別の処理をもって、視

覚変数と *path element* の表現要素を対応づけることとする。表 1 において青色の項目は、*path element* の属性を読みとることで、赤色の項目は、それらを加工することで 5 節にて紹介するプロトタイプシステムが既に対応している。

表 1. *Path Element* と視覚変数の対応表

分類項目	より人間の理解に近づけるための処理	SVGの属性を簡易処理	視覚変数に關係するSVGの属性
Imageの属性			
Geometryとしての対応			
平面上の位置	点	相対的な点の抽出	d, transform
	線	相対的な線の抽出	boundaryを算出 d, transform
	面	相対的な面の抽出	boundaryを算出 d, transform
大きさ	点	相対的な点から面積を算出	x
	線	線長を算出	stroke-width, d
	面	面積を算出	d
濃淡	点	色を明度に交換	stroke, fill
	線	色を明度に交換	stroke
	面	色を明度に交換	fill
分類の属性			
Geometryとしての対応			
きめ	点	相対的な点にパターン処理	x
	線	相対的な線にパターン処理	stroke-dasharray, stroke-opacity
	面	相対的な面にパターン処理	fill-rule, fill-opacity
色	点	色を色相へ交換	stroke, fill
	線	色を色相へ交換	stroke
	面	色を色相へ交換	fill
方向	点	相対的な点にパターン処理	x
	線	相対的な線にパターン処理	x
	面	相対的な面にパターン処理	x
形	点	相対的な点にパターン処理	座標の個数から大まかな形を抽出 d
	線	相対的な線にパターン処理	座標の個数から大まかな形を抽出 d
	面	相対的な面にパターン処理	座標の個数から大まかな形を抽出 d

視覚変数を分類している点線面は、人が知覚できる点線面であり、図素(Graphic Element)である。*path element* の属性である座標から点線面を幾何的に分類することは可能ではあるが、それらの分類は、人の理解と同等になるわけではなく、人は相対的な見えから点線面を判断している。最終的に、点線面の判別を人の理解に近づけるためには、幾何的な大きさを考慮し、しきい値をみつけることで判断することになる。座標からは、面積や長さが計算でき、それらが視覚変数：大きさに対応する。

色に関しては、コンピュータが解釈しやすい RGB から人が理解しやすい HSB に変換することが必要となる。この変換により、視覚変数：色（色相）と視覚変数：濃淡（明度）を抽出することができる。また、人は、きめ、方向、形を前提となるパターンと整合をとりながら知覚しているため、コンピュータは、パターン処理を行うことで人の理解に近づくことができる。

視覚変数と *path element* の表現要素との対応付けは、従来のラスターフォーマットに比べ、ベクトルフォーマットであれば比較的容易に実現可能である。対応付けが可能になったことにより、視覚変数をそれぞれの段階に応じて定量化・視覚化することも容易になる。

4. 地図表現要素の定量化・視覚化の枠組み

この節では、地図を表現する SVG データの表現要素の定量化・視覚化の枠組みを紹介する。

従来、人の知覚に頼っていた視覚変数に対応した表現要素を定量化・視覚化することにより、作成者が視

覚伝達性を確認し評価しながら、SVG データを作成することができる。図 2 には、定量化と視覚化の枠組みをフローとしてまとめた。従来は、作成者は、SVG データを表示させたものを見ることのみで表現を修正するというプロセスを踏んでいたのに対して、この枠組みを用いることで、表現要素を定量化し、数値的に確認し修正することができる。

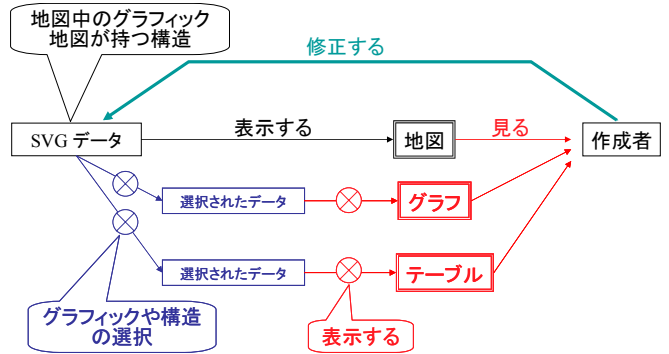


図 2. 地図表現の定量化と視覚化の枠組み

5. プロトタイプシステム

現在、地図を表現する SVG データを対象に、視覚変数に対応した表現要素の分類・定量化・視覚化を行うツール“Map Checker”を開発している。この節では、Map Checker の特徴的な部分を紹介する。

5.1. Map Checker プロトタイプ

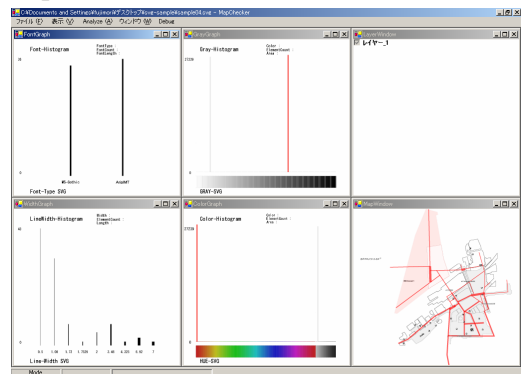


図 3. Map Checker の全体画面例

図 3 は、Map Checker の全体画面例である。利用者は、対象とする SVG ファイルを選択し、視覚変数に対応した表現要素を選択する。Map Checker は、それらの選択に基づいて、SVG データを解析することで、表現要素を定量化し、グラフとして表示する。

5.2. テーブル出力

Map Checker は、表現要素を集計し、テーブルを出力する。図 4 は、SVG データから色に関するクラス分けを行い集計したテーブル出力例である。色のテー

ブルには、行ごとにクラス分けされた色が加えられていき、各行には以下の6項目が存在する。

- (1) 色の名前 (2) 図形の個数 (3) 面積
- (4) 色相 (5) 明度 (6) 彩度

列ごとに昇順、降順といった順序付けが指定でき、それらを用いて色に関係した表現要素をチェックすることができる。

Element	Area	色相	明度	彩度
#123456	100	120	50	80
#789012	200	150	60	70
#345678	300	180	70	60
#901234	400	210	80	50
#567890	500	240	90	40
#111111	600	270	100	30
#222222	700	300	110	20
#333333	800	330	120	10
#444444	900	360	130	5
#555555	1000	390	140	0

図4. 表現要素:色に従ったテーブル出力例

5.3. DOMとViewとの更新伝搬

図5は、SVGデータのDOM構造の視覚化と各elementの表現要素をテーブルとして出力させた例である。基本的にWYSIWYG(What You See Is What You Get)で作成されたようなSVGデータの各elementにはtemporalIDが付いていない。temporalIDを付けなくても、単にSVGデータを表示することや、表現要素ごとの集計は可能である。しかし、視覚変数に対応した表現要素のインスタンス(表現要素インスタンス)と各SVGデータのelementとの連携をインタフェースとして持たせるためには、表現要素インスタンスとSVGデータの各elementのtemporalIDによる以下の2つの対応関係を持たなければならない。

- (1) SVGデータの各element⇒表現要素インスタンス
- (2) 表現要素インスタンス⇒SVGデータのelements

(1)は、SVGデータ中の1つのelementとそのelementに含まれる複数の表現要素インスタンスであり、1対多の関係である。SVGデータの1つの図形を選択に対応して複数の表現要素インスタンスを選択できるともいえる。(2)は、1つの表現要素インスタンスに対してそのインスタンスが含まれる複数のSVGデータのelementsであり、1対多の関係である。1つの表現要素インスタンスの選択に対応してSVGデータのなかの複数の図形を選択できるということもできる。

Map CheckerではSVGデータの各elementのテー

ブルとDOMに基づいた階層構造でtemporalIDを付けることでこれらの対応関係を実現した。図5では、視覚化されたDOMのなかの1つのelementを選択することで、そのelementに対応するテーブル行を選択し、下記10項目へのアクセスが可能であることを示している。

- (1) Element: element name
- (2) ID: element identifier
- (3) Unit: 点線面
- (4) StrokeColor: stroke
- (5) FillColor: fill
- (6) Width: stroke-width
- (7) Perimeter: 周長
- (8) Area: 面積
- (9) PointCount: 座標の個数
- (10) MBR: 最小四角形

Element ID	Unit	FillColor	StrokeColor	Width	Perimeter	Area	PointCount	MBR(MinX,MinY,MaxX,MaxY)
path	stroke	none	black	1	100.000	0.000	2	[0.000, 0.000] [100.000, 0.000]
path	stroke	red	black	2	141.421	50.000	4	[50.000, 50.000] [150.000, 150.000]
path	stroke	blue	black	3	173.205	100.000	6	[100.000, 100.000] [200.000, 200.000]
path	stroke	green	black	4	200.000	150.000	8	[150.000, 150.000] [250.000, 250.000]
path	stroke	yellow	black	5	223.607	200.000	10	[200.000, 200.000] [300.000, 300.000]
path	stroke	orange	black	6	244.949	250.000	12	[250.000, 250.000] [350.000, 350.000]
path	stroke	purple	black	7	263.906	300.000	14	[300.000, 300.000] [400.000, 400.000]
path	stroke	pink	black	8	280.000	350.000	16	[350.000, 350.000] [450.000, 450.000]
path	stroke	gray	black	9	294.155	400.000	18	[400.000, 400.000] [500.000, 500.000]
path	stroke	white	black	10	306.186	450.000	20	[450.000, 450.000] [550.000, 550.000]

図5. DOMの視覚化と各elementのテーブル出力例

6. むすび

本論文は、地図表現を中心としたSVGに対して、図の記号学に基づいて、視覚変数に対応した表現要素を段階的に分類する枠組み、およびそれらの表現要素を定量化・視覚化する枠組みを提案し、プロトタイプシステムの紹介を行った。今後の展開として地図作成プロセスに本システムを用い、システムの評価実験を行うことがあげられる。

文 献

- [1] Scalable Vector Graphics (SVG), W3C, <<http://www.w3c.org/Graphics/SVG/>>
- [2] 有川正俊, 藤森史生, 森田喬, “SVG地図を対象としたグラフィック論理チェックツールの提案,” 2002 地理情報システム学会講演論文集, pp.317-320, Oct.2002.
- [3] M. Arikawa, F. Fujimori, and T. Morita, “A Tool for Checking Graphic Logic in SVG Maps,” ICC2003 Internet Cartography, CD 論文集に掲載, Durban, South Africa, Aug.2003.
- [4] ジャック・ベルタン著・森田喬訳, 図の記号学- 視覚言語による情報の処理と伝達-, 地図情報センター, 1982.