

グリフ作成を支援するドローツールの開発 Development of drawing tool which supports making of a Glyph

石川 凜太郎[†] 三末 和男[‡] 田中 二郎[‡]
Rintaro Ishikawa Kazuo Misue Jiro Tanaka

概要

絵を使って多変量データを可視化する「グリフ」と呼ばれる手法がある。グリフは色や面積といった属性にデータの変量が割り当てられた絵を組み合わせ、多変量データを表現する。これによりデータの特徴を直感的に読み取ることが可能になる。しかし、与えたデータに対して作成者の考える絵のルールが常に満たされるようなグリフを設計し作成する作業は手間がかかる。本研究の目的はこのグリフの作成、及び利用を容易にすることである。そのためにドローツールのような感覚で絵を描いていく過程で絵のルールを与えることができ、さらにそうして作成したグリフのコード生成を行うツールの開発を行った。

1. はじめに

グリフは複数の変量からなるデータ、多変量データの可視化手法の1種である。グリフは、その色やグリフを構成する線の長さがパラメータ化された絵である。パラメータに変量を割り当てることでデータに対応した1つのグリフが生成される。これを並べ、見比べることでデータの特徴を直感的に読み取ることが可能になる。

グリフ作成において問題となるのが、その作成にかかる手間である。グリフはドローツールまたはプログラムによって作成される。ドローツールを用いて作成する場合データ毎に絵を編集する必要があるため、大量のデータを可視化するにはデータに対応した絵を生成するプログラムコードを書くのが一般的である。コードを書く際、ユーザがグリフにする絵が常に満たして欲しいと考えるルールの定式化が必要である。例えば、2つの部品からなる絵の片方の面積をデータに対応させたいとき、データの値によっては2つの絵が重なる可能性がある。ドローツールでグリフを作成する場合は絵を描き直せば良いがコードを書く場合、このルールを式で記述する必要がある。データの変量を絵のどの部分の何の属性に割り当てるのか、変量によって属性値がどのように決定されるのか、データと絵の関係を1つ1つ定義しコードに反映させなければならない。

本研究ではグリフ作成にかかる手間の削減を目的とする。そのためのアプローチとして、グリフ作成のためのドローツール、「GlyphDrawer」を開発した。ツールは描いた絵のプログラムコードを生成する機能をもっている。これにより大量のデータが与えられても、データ毎に絵を修正する必要がなくなる。また、描いた絵にルールを与える手段としてガイド機能を実装した。ガイドを用いて絵を描くことで、ユーザは絵にいくつかのルールを設定することができる。設定したルールはツールが生成するプログラムコードにも反映され、ユーザは絵のルールを定式化することなくグリフを作成することができる。

2. 関連研究

既存のグリフの中でも有名なものが1973年にH.Chernoffが発表した"Chernoff face"である[1]。"Chernoff face"は簡略化された人間の顔の絵を用いた可視化手法で、人間の顔の目の大きさや眉毛の角度にデータの変量が割り当てられている。並んだ顔を見比べた時に、多変量データの特徴を顔の表情として掴むことができるようになっている。

Ribarskyらが発表した「Glyphmaker [2]」はプログラミングの知識のない人のグリフ作成とその利用支援を目的としたツールである。ユーザはツールの中で絵を作成し、データを読み込み、グリフを生成する。グリフの作成はツール上で与えられた数種類の部品を3次元空間上で組み合わせることによって行い、複数の部品をグルーピングし、グループに対していくつか条件を設定することでグリフにある程度のルールを与えることができる。

3. グリフ作成ドローツール

GlyphDrawerを用いて描くことのできる絵は2種類ある。1つは点を任意の座標に打っていきそれを直線または円弧で繋いでいくことで作成される絵。もう1つが正円である。ユーザはこの2種類の絵の組み合わせからなるグリフを作成することができる。描いた絵をグリフとして用いるためにGlyphDrawerがもつ重要な機能が変量設定機能、プログラムコード生成機能、そしてガイド機能である。ユーザはツール上でまず変量の数と名前を決め絵を描く。絵ができた各部品に割り当てる変量、そして変量を部品のどの属性に割り当てるのかを設定する。最後にコード出力ボタンを押すことでコードが自動生成される。

3.1 変量の設定

描いた絵の任意の属性へデータの変量を割り当てることではじめてその絵をグリフとして利用できるようになる。変量を割り当てる属性は無数に存在するが、本研究では、属性を大まかに絵の形状と色の2つに分け、それぞれの中で最も汎用性のある属性と考えられる「length(長さ)」と「brightness(輝度)」への変量の割り当てを可能にした。

3.2 プログラムコード生成機能

GlyphDrawerは描いた絵のコードを出力する機能を実装している。ツール上で設定した絵のルールや変量の種類は全てプログラムコードに反映される。生成されるプログラムコードは設定した変量を引数にもつJavaScriptの関数の形になっている。言語にJavaScriptを選んだのは作成したグリフのWeb上で容易に活用できるようにするためである。

3.3 ガイド

ガイドは描いた絵をグリフにする上で重要な役割を担う。ガイドは描画領域上の任意の位置に設置することができ、設置したガイドはチェックボックスを用いて使用、不使用

[†]筑波大学大学院システム情報工学研究科

[‡]筑波大学システム情報系

の切り替えを行う。GlyphDrawer には直線ガイドと円形ガイドの 2 種類のガイドが実装されている。加えて、円描画機能で描いた円もガイドとして用いることができる。

ガイドの最も大切な役割は描く絵に対するルール設定を可能にすることである。ガイドを使用して絵を描くことでガイドと絵の位置関係の固定や、絵の「length」に対して変量を割り当てた際の絵と変量の対応付けを行うことができる。設定したルールは生成するプログラムコードに反映される。ガイドの使用時、ガイドとマウスマウスカーソルのなす角や距離が画面上に重畳表示される。これは精密な絵の作成を支援するものだが、それに加えて、絵をグリフにした際の「仮の座標」を決める手がかりとなる。絵の「length」に変量を割り当てたとき、絵の各座標は変量によってその値が決まる値が未確定の座標となる。しかし、絵をグリフ化する際には基準となる座標を決めてあげる必要がある。これが「仮の座標」である。ガイドを使用して絵を描くことでガイドのある特定の点を基準とした相対座標でこの仮の座標を定義することができる。

円形ガイドまたは描いた円をガイドとして用いて絵を描いたとき、仮の座標は円の中心からの相対座標で表され、絵と円の中心点とのなす角が固定される。また、円形ガイドを用いて絵を描くと各制御点とガイドとの距離が変量によって値が決まるパラメータとなる。円をガイドとして用いた場合には円と各制御点との距離は固定される(図 2)。

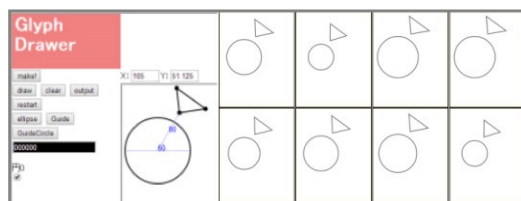


図 2 円をガイドとして用いたグリフ作成

直線ガイドを用いて絵を描いたとき、仮の座標は直線の始点からの相対座標で表され、絵の制御点から直線ガイドへ正射影した点とガイドの始点との距離が固定される。直線ガイドを用いて絵を描くと各制御点とガイドとの距離が変量によって値が決まるパラメータとなる(図 3)。

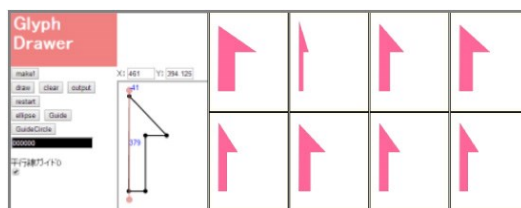


図 3 直線ガイドを用いたグリフ作成

ガイドを実装した目的はユーザにルールを意識しながら絵を描かせることにある。ツールにはいくつかのルールが定式化されて組み込まれており、使うガイドや描く順番に対応したルールがコードに反映される。ユーザが絵を描く手順を読み取り自動的にルールを設定することでプログラムによるグリフ作成に必要なルールの定式化にかかる手間の削減を図った。

4. ユースケース

作成したツールを使用してグリフの作成を行った。作成したのはある 1 地点への人の移動量を表現したグリフである(図 4)。読み込んだデータは日付とある地点に留まった人

の数、入ってきた人の数、出て行った人の数、の 4 変量をもつ JSON 形式のデータである。日付の区切りは 1 日単位でそれが 1 ヶ月分あり、1 ヶ月分の絵をカレンダーの上に並べてグリフにした。また、絵には「円と 2 つの矢印が常に接する」、「日付を除く 3 つの変量をそれぞれ円の半径、円に向かう矢印の太さ、円から出て行く矢印の太さに対応させる」というルールを設定した。

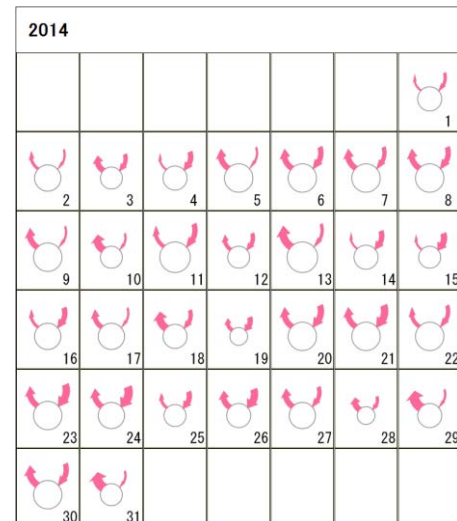


図 4 ユースケースとして作成したグリフ

このグリフを作成するに辺って矢印の太さを変量に対応させるため円の上に円形ガイドを設置し、それに沿って矢印を描いた。また、円と矢印が常に接するというルールを満たすために最初に描いた円をガイドとして用いた。これにより、コード上で矢印の円と接する点を円の中心からの相対座標で表されるように設定した。カレンダーは HTML5 を用いて作成し、絵の JavaScript 関数をカレンダーを作成するコードの中で呼び出して図 3 のグリフを作成した。ツールを用いたことにより 3 変量に対応し、さらに位置の制約を持つ絵をコードを書く手間なしに描くことができた。絵は JavaScript のコードになっているため、データが 10 年分になったとしても関数を呼び出すだけで大量の絵を生成することができる。

5. まとめ

本研究ではグリフ作成にかかる手間の削減を目的としたツールの開発を行った。グリフ作成において、最も時間がかかると思われるのが絵に対しルールを設定する作業である。開発したツールでは、絵を描いていく過程でこのルールの設定を可能にするガイド機能を実装し作業の時間短縮を図った。さらに、既存研究との差分として作成したグリフのプログラムコードを生成する機能を実装した。作成したグリフのコードを別のコードに埋め込める形にすることで、Web ページなどでの外部利用を容易にした。

参考文献

- [1] Herman Chernoff, "The Use of Faces to Represent Points in K-Dimensional Space Graphically", Journal of the American Statistical Association, Vol. 68, No. 342 (1973), pp.361-368
- [2] William Ribarsky, Eric Ayers, John Eble and Sougata Mukherjea, "Glyphmaker: Creating Customized Visualizations of Complex Data", Computer, Vol. 27, Issue. 7 (1994), pp.57-64