

カタマリを見つけて迎れる ヒストリデータブラウザ(4)

松原 伸人、土屋 正人

matubara@sra.co.jp, m-tsuchi@sra.co.jp

大量のデータをインタラクティブに操作する、Webブラウザ上で動作するアプリケーションを開発しています。

「カタマリを見つけて迎れるヒストリデータブラウザ」¹は、歴史データや開発ログといった大量のイベントを、時間軸上に並べて表示することで、イベントが連続的に起きている時間帯や切れ目を見つけて迎れるようにするWebアプリケーションのプロタイプです。

このプロタイプでは、どのような実装をすれば大量のデータをインタラクティブに扱えるか？ 反対に、実現したいインタラクションを実装してみた結果として、インタラクティブ性能がどうなるか？ ということを試行錯誤して探っています。

大量のデータの中から特定のイベントや似たようなイベントを見つけるには、1つ1つイベントを見ていて、タグをつけて仕分けていきます。

Vol.93 では、京都の歴史データ約 12,000 の出来事のクラスタリング結果を可視化した例を紹介しましたが、今回はその実装を中心に紹介します。

◆ インタラクティブの実装

イベント数が 12,000 のクラスタリングした結果、クラスタに含まれるイベント数は 97,932 でした。

6 種類のクラスタ結果を表示して比較するような場合、約 100,000 のイベントを画面上に表示し、操作に

応じて一部のイベントの色や位置を変えて表示する処理を、30fps くらいの表示スピードで行えなければなりません。

このプロタイプでは、大量のイベントデータを表示した時に、イベントの見た目や配置を変えるとどのように見えるか、実装方法によって操作性がどう変わるかを試せるようにしています(図1~5)。

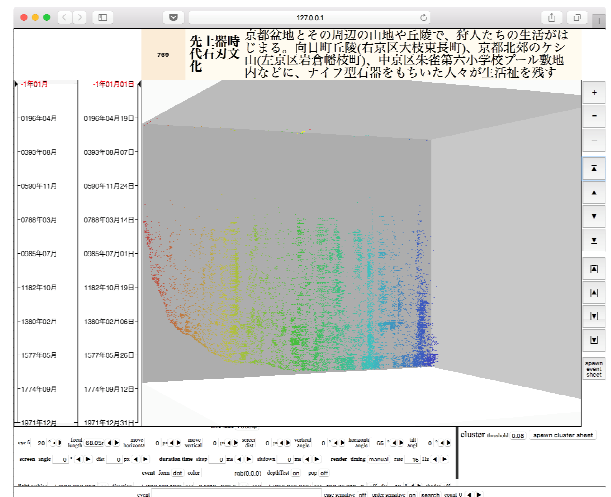


図1 点で表示 (水平方向の位置が内容の文字数に比例)

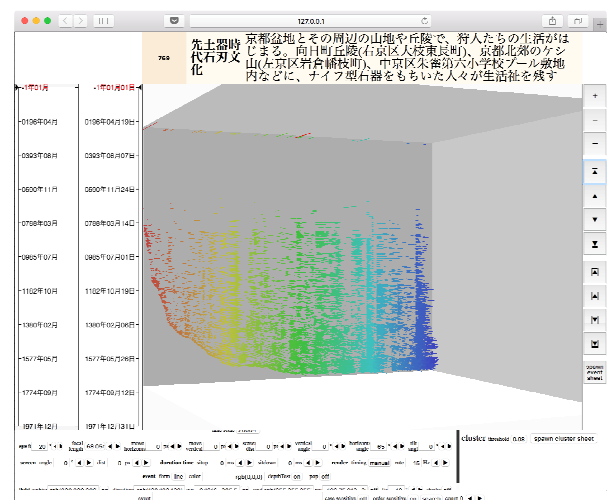


図2 線で表示 (線の長さが内容の文字数に比例)

¹ GSLetterNeo Vol.91、92、93 を合わせてご一読頂けると幸いです。

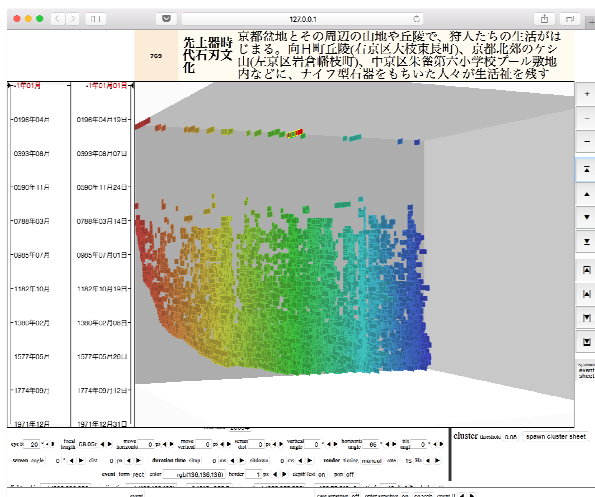


図 3 四角形で表示 (四角の横幅が内容の文字数に比例)

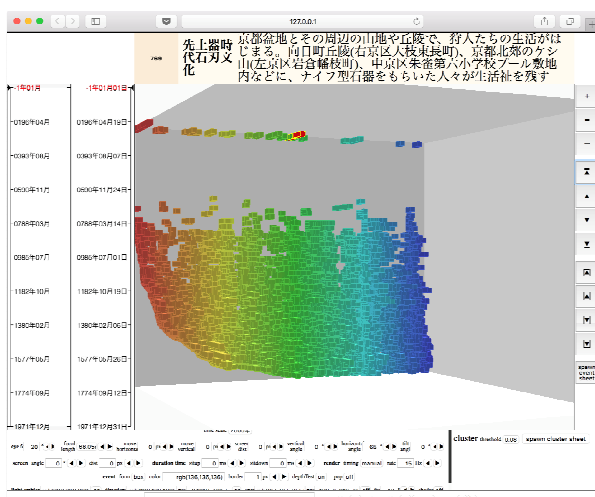


図 4 直方体で表示 (横の辺の長さが内容の文字数に比例)

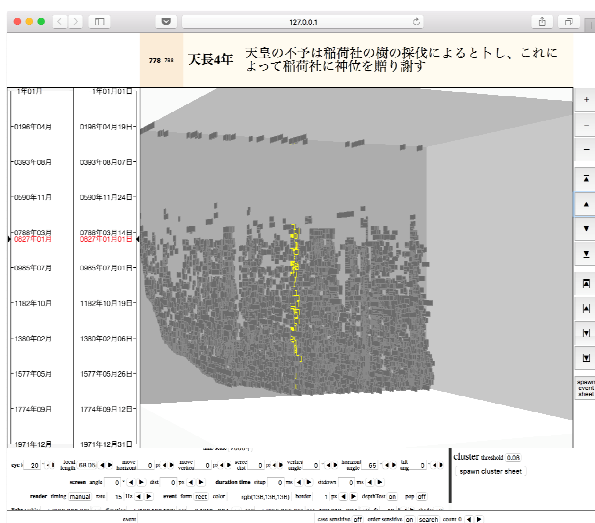


図 5 黒一色で表示。選択したイベントやクラスターのハイライトが目立つ

イベントの分類データをインタラクティブに扱うことを目的とするものとしては、Canvas を用いて 2 次元表現する方法、CSS を用いて 2 次元表示する方法、CSS を用いて 3 次元表示する方法を経て、現在は WebGL を用いた 3 次元表示で実装を進めています。

WebGL による実装では threejs を用いています。threejs は、三次元形状作成に geometry、material、mesh といった 3D-Object や座標変換に perspective-camera といったオブジェクトを用い WebGL の複雑な記述を行わずに記述できる、JavaScript 用の 3 次元プログラミングライブラリです。

WebGL だけでなく、CSS の 3D 表現も同様の記述方法を用いて扱えるようになってきました。大量データを処理するために VBO (Vertex Buffer Object) を使用しています。

threejs で VBO を用いる場合いくつか方法があるようですが、BufferObject を使っています。BufferObject は、3 次元オブジェクト形状を頂点データの配列で表し、各頂点データは座標、色、法線ベクトル、インデックスで表しています。テクスチャを用いる場合は、テクスチャ座標を色の代わりに使用します。

各イベントを 1 つの BufferObject で作成することができますが、速く描画処理するために 1 つの BufferObject に全てのイベントを扱うように記述しています。こうすると GPU 側のメモリ上にデータを格納するため、描画処理の際に GPU へのデータ転送時間を減らすことができます。

検索結果を緑色にしたり、選んだクラスイベントを黄色にしたり、時間範囲を変えて表示するイベントを限定するといった色や形状を変更する時は、変更するデータだけ VBO を書き換えます。

書き換えるデータを指定できるよう、オブジェクトの形状データごとに対応する VBO のインデックス群を管理します。

◆ 描画オブジェクトのフィルタリング

描画したいオブジェクト (頂点) のインデックスを BufferObject にインデックスを表す整数の配列でセットします。

`BufferObject.setIndex(indices);`

検索結果だけ表示する時や、指定した時間範囲内のイベントだけ表示する時など、特定のオブジェクトだけ描画する際に用いることができます。

また、VBO は配列に設定したインデックスの順序でオブジェクトを描画します。

選択したオブジェクトを手前に表示することで強調するような時、そのオブジェクトのインデックスを配列の後ろに入れ、最後に描画されるようにします。

◆ 色の変更

色は `colors` 属性に 0~1 の RGB 値の配列で設定します。色を変えたい頂点のインデックスが `I` の場合、色が格納されている位置は次のようになります。

$R_i = I * 3$

$G_i = I * 3 + 1$

$B_i = I * 3 + 2$

`colors = BufferObject.colors`

で色の配列を取り出して、先の `Ri`, `Gi`, `Bi` の3つのインデックスで指定して配列を書き換えます。赤色に変える場合、

`colors[Ri] = 1.0;`

`colors[Gi] = 0.0;`

`colors[Bi] = 0.0;`

となります。

◆ 位置の変更

位置は `positions` 属性に 3 次元座標の配列で入っています。色の変更と同じように位置のインデックスは、

$X_i = I * 3$

$Y_i = I * 3 + 1$

$Z_i = I * 3 + 2$

と指定し、`BufferObject.positions[Xi] = 0` という風にかき換えます。

WebGL が扱える基本形状は、点、線、三角形の 3 種類のみです。複雑な形状の表現はこれらを組み合わせで行います。

◆ 使用データについて

掲載図で使用しているデータは、「ユーザーの主眼的理解醸成のためのデータ表現とインタラクティブ性のデザイン」研究グループ（主たる共同研究者：中小路久美代）²による研究の一部として、京都大学学際融合教育研究推進センターデザイン学ユニットの北 雄介先生と共同で、京都の歴史年表から収集したテキストデータです。地名や人名を抽出してイベントを分類しています。

² 本研究グループは、JST 戦略的創造推進事業 CREST「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」領域 H26 年度採択課題「データ粒子化による高速高精度な次世代マイニング技術の創出」（代表：宇野毅明（NII 教授））のメンバーです。

GSLetterNeo Vol. 94

2016 年 5 月 20 日発行

発行者 ●株式会社 SRA 先端技術研究所

編集者 ●土屋正人

バックナンバーを公開しています ●<http://www.sra.co.jp/gsletter>

ご感想・お問い合わせはこちらへお願いします ●gsneo@sra.co.jp

夢を。



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。Yawaraka Innovation
やわらかいのバージョン