GSLetterNeo vol.141

2020年4月

Webブラウザにおける2Dコンテンツ上での透視投影変換を用いた3D表現（3）

松原 伸人 matubara@sra.co.jp

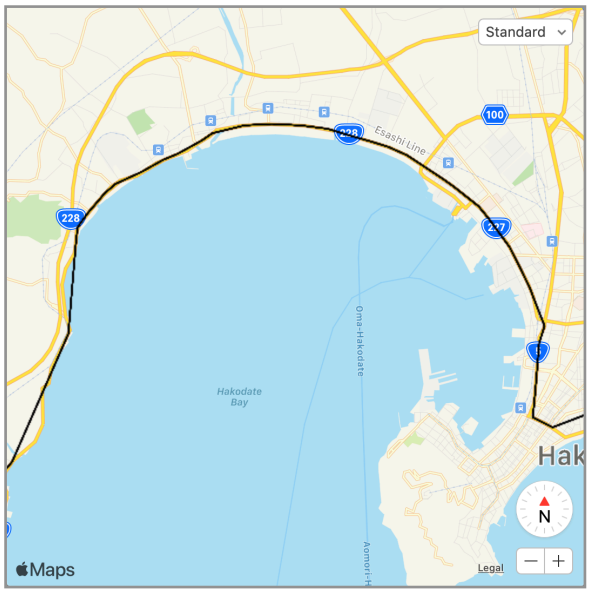
はじめに

今回はVol.139に続き、Webブラウザの透視投影変換を適用することで2Dコンテンツに情報を重ねて3D的に表示する方法を紹介します。Vol.139では 開発中の運行情報の可視化プログラムの BEAm で用いている GTFS について説明し、運行経路の表示方法を紹介しました。このプログラムは **[test-draw_routes]** を Webブラウザで開いて試せます。GTFS データがあるフォルダを指定すると、データを読み込んで地図上に全運行経路を表示します。運行経路は黒色の線で描いています。

プログラム
[test-draw_routes]
test-draw_routes.html

経路の2次元平面上での描画

BEAm ではおもに、GTFS の routes.txt と trips.txt と stop_times.txt と stops.txt の4つのデータを用い、地図上に各系統の運行経路の駅の位置を直線でつないで表示し、各旅程の始発駅から終着駅までの各駅の発着時刻を直線でつないで表示します。Vol.139に書いたように、各系統の旅程の運行経路は stop_times.txt および stops.txt に書かれている駅の経緯度を停車順に直線で繋いだ線で canvas 平面上に描けます。次の画像は、読み込んだ GTFS の中から1本目の系統の経路を描画した様子です。**[test-draw_route]**



GTFSの中から1本目の系統を描画した様子

リファレンス | GTFS
リファレンス | Static Transit | Google Developers

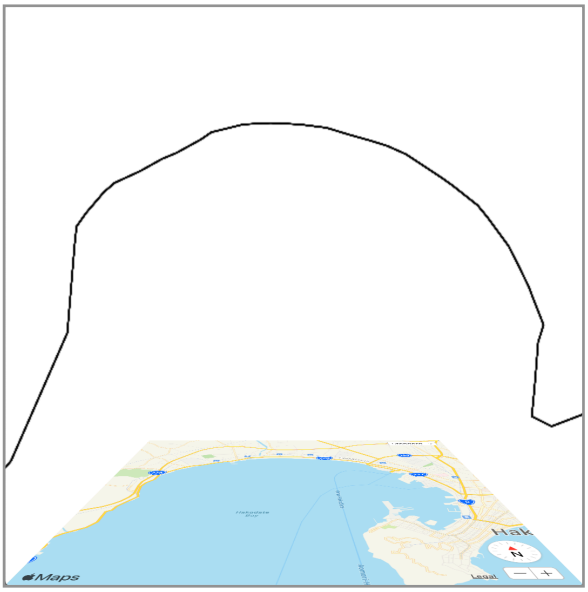
プログラム
[test-draw_route]
test-draw_route.html

地図を傾ける

Vol.137では、HTML の iframe を CSS の perspective プロパティを利用して表示エリア奥に傾けて立体的に表示する方法を紹介しました。地図も同様に傾けて表示できます。例えば、地図の下端を軸にして表示エリア奥に90度傾けると次の画像のように表示されます。**[test-rotate_map]** canvas は表示エリア奥に傾けずにそのままにしています。そのかわりに、これまで地図上の位置は二次元座標で扱ってきたのを変更し、表示エリアの奥行き方向を加えた三次元座標で扱うようにします。

プログラム
[test-rotate_map]
test-draw_route_rotate_map.html

本紙面に掲載している画像は、共同研究を行っている公立はこだて未来大学の協力により、函館バスから提供いただいた GTFS を使用し生成しています。



地図を表示エリア奥に90度傾けた様子

2次元から3次元への変換

表示エリアに水平な方向を X軸、表示エリアに垂直な方向を Y軸、表示エリア奥方向を Z軸 とする三次元空間とします。90度傾けた地図はちょうど X-Z平面上 に移ったことになります。X-Y平面上 にある経路も90度傾ければ Y-Z平面上 に移ります。この2次元平面から3次元空間への座標の変換は次のように書けます。2次元平面では原点が表示エリアの左上隅で水平右方向が正の x座標 で垂直下方向が正の y座標 です。用いる3次元空間では原点が、表示エリアの中央で水平右方向が正の x座標 で垂直下方向が正の y座標 で奥方向が正の z座標 としました。CSSで地図の傾きを変数 --rotationX に設定しています。2次元-3次元座標変換ではこの --rotationX 変数の数値を用いて回転移動変換しています。

プログラム

[test-perspective]
test-draw_route_perspective.html

```
function pointToPoint3d (point2d) {
    let rect = canvasRect(),
        point3d = {
            x: (point2d.x / rect.width - 0.5) * rect.width,
            y: (point2d.y / rect.height - 0.5) * rect.height,
            z: 0
        }
    return rotatePoint3dToMap(point3d)
}
```

[test-perspective]

透視投影変換

90度表示エリア奥に傾けた地図は、表示エリア中心に向かって台形に歪んでいます。この歪み方により地図に奥行きがあるように見えます。この歪め方が透視投影変換で、CSS の perspective と perspective-origin プロパティによる効果です。ここでは DeepPoint.perspectiveTransform で定義している透視投影変換を 用いています。上記で求めた3次元座標の透視投影変換での表示エリア canvas 2次元平面上への座標変換は次のように書けます。サンプルプログラム **[test-perspective]** では、GTFS を読み込んでルートを1つ、90度表示エリア奥に傾けた地図上に描きます。

Perspective projection
Transformation matrix - Wikipedia

```
function perspectiveTransform (point3d) {
    let rect = canvasRect(),
        vpoint = DeepPoint.perspectiveTransform(point3d, {width: rect.width / 2, height:
rect.height / 2}, rect.height * 1.5, rect.height * 0.75)
    return {
        x: vpoint.x * rect.width + rect.width / 2,
        y: vpoint.y * rect.height + rect.height / 2
    }
}
```

[test-perspective]

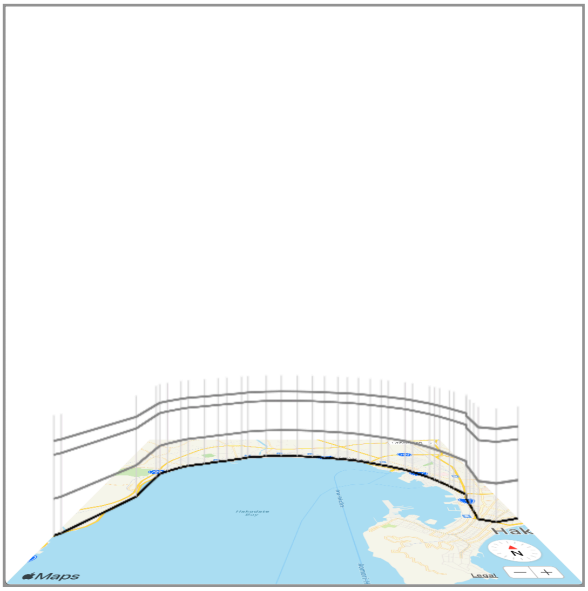
旅程の描画

最後に GTFS の旅程を地図上に表示してみます。GTFS の旅程は系統ごとに始発駅から終着駅にいたる路線で停留する各駅での発着時刻を記述したデータです。各系統は、9時発の旅程とか10時発の旅程とか複数の旅程が記述されています。運行本数が5分に1本くらい多く組まれている系統もあれば、1日に1本のような系統もあります。サンプルプログラム **[test-draw_time_routes]** は、地図の各駅（停留所）の位置に垂直に1日を表す上端が0時で下端が24時の時間軸を描きます。そして旅程ごとに、始発駅から終着駅まで停留する駅の時間軸の発時刻と着時刻の位置を直線でつないで順に描きます。本数の多い系統は横に走る線がたくさん描かれます。朝方に本数が多いと上の方に線が描かれ、夕方に本数が多いとしたの方に線が描かれ

プログラム

[test-draw_time_routes]
test-draw_time_routes.html

90度奥に傾けた地図上に描いた様子



GTFSの旅程を地図上に表示

ます。webインスペクタを開いてselectRouteAtIndex(1) を実行すると2本目の系統を描画します。系統を1本ずつ選んで表示できます。mapRotation(30) を実行すると地図の傾きを30度に変更して描画します。このプログラム例では、駅の上の時間軸が常に垂直になるように描画しています。地図の傾きが変わっても旅程が見やすい状態を維持することを意図しています。時間軸の座標の計算は **[test-draw_time_routes]** の165行目あたりにあるtimelineOfStop関数に書いてあります。時間軸の高さは canvas の高さの4分の1に設定しています。このあたりを書き換えれば、地図の傾きに応じて時間軸の高さを変化させたり、真上から見下ろした時は時間軸を表示しないようにしたりするなど、色々な表現ができると思います。

```
function timelineOfStop (stop) {
    let line = timelineByStop_id[stop.stop_id],
        timelineHeight = canvasRect().height / 4
    if (!line) {
        let from3d = point3dOfStop(stop),
            to3d = {
                x: from3d.x, y: from3d.y - timelineHeight, z: from3d.z
            }
        line = {
            from: perspectiveTransform(from3d),
            to: perspectiveTransform(to3d)
        }
        timelineByStop_id[stop.stop_id] = line
    }
    return line
}
```

[test-draw_time_routes]

GSLetterNeo vol.141
発行日 2020年4月20日
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 https://www.sra.co.jp/gsletter/

お問い合わせ
gsneo@sra.co.jp
〒171-8513 東京都豊島区南池袋2-32-8
株式会社SRA