Geovisualisation Of Animated Tides In Coastal Area

Architecture and applications with 2 Open Source and OpenGL platform

Antoine Masse, Sidonie Christophe
National French Mapping Agency (IGN), COGIT laboratory

Code Loves Maps Workshop, AGILE 2016, Helsinki
2016/06/14
Visualisation Of Coastal Area: What Are The Challenges?

**Challenge**: Convey information through geovisualisation

- with manipulation of various geographical objects
- with objects that may vary in time or/and in space
- with user-interaction: control of the stylisation, etc.

**Question**: Does this already exist?
Visualisation Of Coastal Area: What Already Exists?

Interactive, data quality

Not animated, not continuous, fixed stylisation

https://coast.noaa.gov/llv/#/lake/michigan
Visualisation Of Coastal Area: What Already Exists?

Interactive

Not animated, not continuous, fixed stylisation

http://flood.firetree.net/
Visualisation Of Coastal Area: What Already Exists?

Interactive, data quality

Not animated, fixed stylisation

http://ss2.climatecentral.org/
Visualisation Of Coastal Area: What Already Exists?

Animated fixed stylisation

https://www.windyty.com
Visualisation Of Coastal Area: What Already Exists?

Animated, fluid

Fixed stylisation, not interactive (video)

http://co2.digitalcartography.org/
Visualisation Of Coastal Area: What Already Exists?

Animated, interactive, multiple stylisation

Not continuous, not fluid

http://data.shom.fr/#donnees/oceanographie/animation
Visualisation Of Coastal Area: What Already Exists ?
And What Do We Need ?

Stylisation
- Standard OGC SLD/SE
- Photorealistic and non-photorealistic tools

Interactivity
- Multiple level of user interaction from Expert to Novice

Animation
- 4D degree of liberty
- Real-time information

Accuracy (data & rendering)
- High quantity of heterogeneous data
- Smooth rendering
Our Proposition

• Architecture
  OpenGL rendering engine
  Open Source software (available on github)

• Implementation
  • Stylisation: OGC standard SLD/SE and more (Bertrand’s presentation)
  • Interactivity: user can directly modify style/data/animation
  • Animation: based on real phenomenon (tides: observation and prediction)
  • Quality: based on high quality data with WMS/WMTS services
Our Implementation Proposal

Geographical data
(cartographic elements, ortho-images, dtm, etc.)

Other data
(tides, etc.)

Data selection for visualization

Data processing (combination, etc.)

Inspiration source

Graphical parameters
(colors, etc.)

Category
(abstraction, photorealism)

Rendering engine

Style

User

Final rendering

perception

Data selection for visualization

Data processing (combination, etc.)

Inspiration source

Graphical parameters
(colors, etc.)

Category
(abstraction, photorealism)

Rendering engine

Style

User

Final rendering

perception
OPENGL rendering engine: how shaders work?

Rendering Engine

- Data
- Vertex Shader
  - Triangle assembly
  - Rasterization
- Fragment Shader
  - Blending
- Stylisation
- Screen
- Framebuffer

AGILE '16 - CODE LOVES MAPS WORKSHOP
Opengl: How Shaders Works?
Example of a colormap from raster image

Vertex Shader

```glsl
uniform float m00 = 1.; // X homothetic value in 3x3 matrix
uniform float m02 = 0.; // X translation value in 3x3 matrix
uniform float m11 = 1.; // Y homothetic value in 3x3 matrix
uniform float m12 = 0.; // Y translation value in 3x3 matrix
uniform float screenWidth;
uniform float screenHeight;

layout (location = 0) in vec3 vertexPosition;
layout (location = 1) in vec2 vertexTextureCoord;
layout (location = 2) in vec4 vertexColor;

out VertexData {
    vec4 color;
    vec2 textureUV;
} vertexOut;

void main(void) {
    gl_Position = vec4(-1.0 + 2.0 * (vertexPosition.x * m00 + m02) / (screenWidth + 1.0),
                      1.0 - 2.0 * (vertexPosition.y * m11 + m12) / (screenHeight + 1.0), 0.0, 1.0);
    vertexOut.color = vertexColor;
    vertexOut.textureUV = vertexTextureCoord;
}
```
Opengl: How Shaders Works?
Ex: colormap from raster DTM image

```glsl
uniform sampler2D bufferImage;
uniform float globalOpacity = 1.0;
uniform float objectOpacity = 1.0;
uniform int typeColormap = 0;

// Vertex data in
in VertexData {
    vec4 color;
    vec2 textureUV;
} fragmentIn;

// Fragment out
out vec4 outColor;

void main(void) {
    // Image coordinates (screen)
    vec2 P = fragmentIn.textureUV;
    // The raster goes in the rectangle
    vec4 pixel = texture(bufferImage, P);
    // We apply the colormap
    outColor = interpolateColor(pixel);
    // Opacity (multiplication)
    outColor.a = outColor.a * globalOpacity * objectOpacity;
}
```

https://notes.underscorediscovery.com/shaders-a-primer/
Applications

**Data**
- Elevation data
- Tide data

**Inspirations**

**Examples**
1. Water depth perception by intervals
2. Continuous perception of water depth
3. Continuous perception of water depth
4. Sea infiltration perception

**Cartographic data**

**Ortho-images**

Ortho-images
1st example
perception of water depth

Inspiration: Map abstraction (1:25,000)

Conventional colour palette (OHI)

<table>
<thead>
<tr>
<th>Depth (m)</th>
<th>0</th>
<th>-5</th>
<th>-10</th>
<th>-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Low tide
05/28/2009
14:00

High tide
05/28/2009
19:00
1st Example

Geovisualization of coastal area
Perception of water depth by intervals
1 tidal cycle (12 hours)

Time x4000

https://youtu.be/x2tFyOmTyOE
**2nd Example**

Geovisualization of coastal area

Continuous perception of water depth
1 tidal cycle (12 hours)

![Map Image](image-url)

**Time x4000**

![Map Image](image-url)

Depth (m) | 0 | -5 | -10 | -20
---|---|---|---|---
Color

[Video Link](https://youtu.be/vmKfoGPTS8E)
Inspiration: Ortho-image photo-realism

3rd example
Continuous water depth perception

Low tide
05/28/2009
14:00

High tide
05/28/2009
19:00

Natural colour palette from original ortho-photo

<table>
<thead>
<tr>
<th>Depth (m)</th>
<th>0</th>
<th>-5</th>
<th>-10</th>
<th>-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3\textsuperscript{RD} Example

Geovisualization of coastal area

Continuous perception of water depth
Ortho-photo-realism, 1 tidal cycle (12 hours)

Time x4000

https://youtu.be/Fh--szuxlzw
4th Example

Geovisualization of coastal area
Continuous perception of water depth
Ortho-photo-realism, 1 tidal cycle (12 hours)

https://youtu.be/lUZ5V9nqvVE
Next

Web-g1 Platform

Web application using the last standards for:

- Massive 3D data visualisation on desktop/tablet without any plugin
- Interactions, high data quality
- Scientific results sharing

Based on the three.js toolbox
Application

same geographical data
→ direct access with WMS/WMTS services

same statistical data (tides)
→ direct access with HTTP request

same shaders (colors, animation, etc.)
… but in 3D
1ST Example

https://youtu.be/TImlkLAYGxg
2\textsuperscript{ND} Example

https://youtu.be/T1mlkLAYGxg
3\textsuperscript{RD} Example

https://youtu.be/X4EzCt56lgk
What Are Your Thoughts About: Time Representation

What is real and what is simulated?

- Stylisation: realism for real things and abstraction for simulations?
- How do we convey time information to users?

Interfacing data and users?
What are the limits?

Imprecision of data, how do we represent that?

Stylisation
Interactivity
Animation
Quality (data & rendering)
THANKS

Any questions?
About the architecture?
About the implementation?
About the results?

Any suggestions for improvements?

http://ignf.github.io/geoxygene/
http://www.itowns-project.org/

mapstyle.ign.fr

nice software

antoinemasse.fr

nice project

nice speaker