Situation Plotting Algorithm and System Implementation Based on Web

N Y Chen*, L Chen and W Xiong

College of Electronic Science, National University of Defense Technology, Changsha, China

*Email: nanyustudy@163.com

Abstract. Based on the integration of existing programming languages and framework technologies, this paper designs and implements a Web-based situational plotting system. This system combines Web-oriented and map services, it uses Leaflet framework and SVG (Scalable Vector Graphics) technology design and implementation, classify the symbols and generate corresponding situational symbol frameworks, which can be extended to different areas such as emergency rescue, logistics coordination, communication planning, and realize the situational symbol mapping and editing, map browsing, and plotting. Sample set saving and uploading functions have good real-time, cross-platform and interactive features. At the same time, aiming at the problem of generating linear plotting symbols, a plotting algorithm based on Centripetal Catmull–Rom curve is proposed, which avoids sharp angle and self-intersections in the process of plotting linear symbols. The experimental results prove that compared with the traditional curve plotting algorithm, the proposed algorithm has the characteristics of fewer control points and good fitting effects, which improves the plotting efficiency.

1. Introduction

As an important technical means to reflect the situation of emergency events, the situation plotting has been widely used in fire control, earthquake disaster rescue, communication and repair, etc, the development and construction of the situation plotting system has become a hot research topic, such as the dynamic plotting software of ArcGIS, GiSpace, its dynamic plotting API is a dynamic plotting interface based on the ArcGIS Flex API that can be used for business systems based on ArcGIS development. Aodun's smart forest fire protection system Wisdomforest stores or calls fire-fighting command situations to map topographic maps in different image formats such as jpg, gif, bmp, etc, edit, draw, modify and store points, lines, surfaces, texts, labels and other elements [1]. However, most of these situational plotting systems are based on desktop applications, which are complicated to install and operate, they require high professionalism and hardware environment, and have poor scalability, it is difficult to have the real-time and cross-platform features required for emergency situation mapping, with the continuous development of network and GIS technology, Web-based application research is also deepening, relying on the good scalability and real-time nature of Internet technology, this paper is based on Leaflet framework and Scalable Vector Graphics (SVG) technology design and implementation of a web-based situation plotting system.

In the process of plotting symbol generation, traditional linear plotting algorithms are often implemented using Bezier curves, NURBS curves or B-spline curves. However, the control points of
such curves are not strict enough to control the shape of the plotted symbols and cannot pass through all control points. Many researchers have proposed improvements, but most of them need to be implemented by adding additional control points, such as Z. Y. Duan, by increasing the control point to make the B-spline curve pass through the main control point, and retain partial modification [2]. In addition, for the smoothness problem of the curve, H. Y. Chen proposed an approximate algorithm based on quartic B-spline curves to improve the smoothness and uniformity of the curve [3]. H. Q. Chen et al. used the Catmull curve interpolation algorithm to obtain the smooth curve path of the dynamic mapping of the gully class, and achieved the smoothness of the plot by the continuous smoothing of the curve [4], but the Catmull curve interpolation algorithm is prone to the shape of the plot when the control points are dense [5]. In this paper, by analyzing the characteristics of the plot curve, the Centripetal Catmull–Rom curve algorithm is used to generate linear plot symbols, which avoids the “sharp angle” and “self-intersection” of the plot curve [6].

2. Data model

2.1. Symbol classification
According to the drawing method and characteristics of the plotting symbols, the situation plotting symbols are divided into two types: point symbols and function symbols, the specific structure is shown in Figure 1. The point symbol refers to a symbol that does not vary with the scale of the map and the shape of the symbol is composed of basic primitives, the structure is simple and fixed, and the symbol has clear positioning and directivity, generally, the point symbol has only position information and attribute information, which respectively represent its positioning meaning and attribute meaning[7], such symbols can be automatically drawn by the user only by entering a coordinate point, and the size, direction and color can be adjusted in the layer editing bar and the style panel, it can be used to indicate fixed target objects such as stations, schools, hospitals, command posts, and other entities. The function symbol has a certain geographical scope meaning and shape, and the graphic composition is relatively complicated [8], it cannot be completed by simple basic primitives, the user needs to manually input some control points, and use the related algorithm to realize symbol drawing by calling atomic function. After the drawing is completed, the control points can be added, deleted, adjusted, etc, and the color, transparency, and line type of the symbol can also be changed in the style panel, and corresponding text annotations can also be added to assist the explanation. Due to its shape variability and flexibility, function symbols can be used to indicate situations such as rescue, evacuation routes, and assembly areas.

![Figure 1. Plotting symbol classification](image)

2.2. Symbolic data Structure
The basic elements contained in the plot symbol are: symbol color, fill color, fill transparency, texture style, line style, line width, font, font color, font size, font weight, image, zoom, control point, center point, etc, the concrete symbol concept layer class diagram is shown in Figure 2.
The basic public interface is defined in the public parent class `BaseObject`. The symbol extension interface `getPath()` uses the SVG language to draw the situation symbols. When the user needs to add symbols, the interface can be extended, and the attributes are set in the corresponding subclasses. `Save()` saves the symbols, `Load()` loads the symbols, and there are some mouse operations. `RegularMarkObject` and `IrregularMarkObject` inherit from `BaseObject`. The `CenterLatlngs` attribute is used to locate the center point of the symbol in the rule symbol class, that is, the point symbol class, its subclasses are `VectorObject`, `GridObject`, and `TextObject`, added vector symbol style settings such as color, zoom, and rotation to `VectorObject`, the image resource and the width and height methods are defined in the `GridObject`. In the `TextObject`, the text paragraph style setting and the outer frame style setting are defined for the text symbol, and the class is also inherited by `IrregularMarkObject` enables irregular notes to obtain text annotation. The `ControlLatlngs` property and `ElevationLatlngs` property are set in the `IrregularMarkObject` class. Among them, the `ElevationLatlngs` property can be read in the 3D graphics, so that the 3D plot symbol can obtain the elevation value, as well as defining the line type, line width, fill color, fill transparency, texture style and other methods.

**Figure 2.** Plot symbol concept layer class diagram

**Figure 3.** Plot symbol application extension
In practical applications, due to the good scalability of the symbol structure of the system, as shown in Figure 3, the plot symbols can be extended to emergency rescue, logistics coordination, communication planning and other fields according to requirements.

3. Symbol generation algorithm

3.1. Control point generation

For point symbols, the control point generation is relatively simple, only the user needs to input a single coordinate, but for irregular symbols such as arrows, routes, and assembly areas, the shape is not fixed, and changes with the actual situation, so the generation of control points becomes the key to symbol plotting. The control point generation algorithm is discussed below by taking the assembly area symbol as an example.

As shown in Figure 4, the assembly area symbol is controlled by the main control point input by the user and the slave control point generated by the main control point according to the corresponding algorithm. After obtaining the master-slave control point, the corresponding feature polygon can be generated, and then the Bezier curve is fitted to form the assembly region symbol. The specific algorithm is as follows:

1. Connect \( p_1 \) and \( p_3 \) and take the midpoint \( p_m \).
2. Connect \( p_1p_2 \), \( p_2p_3 \), make a vertical line of the angle bisector of the \( \angle p_1p_2p_3 \) by the point \( p_2 \). On the vertical line, take two points \( p_{21} \) and \( p_{22} \) on both sides of \( p_2 \) according to the fixed ratio \( t \), so that \( p_1p_2: p_{22}p_{21} = p_3p_2; p_{22}p_{21} = t \).
3. According to step 2, the slave control points \( p_{31} \) and \( p_{32} \), \( p_{m1} \) and \( p_{m2} \), \( p_{11} \) and \( p_{12} \) are obtained by \( p_3p_2 \) and \( p_2p_m \), \( p_m p_1 \) and \( p_1p_2 \) in order, thereby generating feature polygons.
4. A Bezier curve is generated by fitting \( p_1, p_{12}, p_{21}, p_2 \), which is tangent to the starting point \( p_1 \) and the end point \( p_2 \).
5. According to step 4, three Bezier curves are sequentially generated by \( p_2, p_{22}, p_{31}, p_3, p_{32}, p_{m1}, p_m, p_m, p_{m2}, p_{11}, p_1 \), and a total of four curves are used to complete the symbol drawing.

Figure 4. Assembly areas symbol implementation method

Figure 5. Comparison of three of Catmull-Rom curves

After the drawing is completed, the main control point can be adjusted in the editing panel until the expected requirements are met. The specific program flow is as follows:

Input: control point coordinates
Output: curve node coordinates
Definition:
- getThirdPoint function (parameter 1, parameter 2):
Connect the two-point coordinates of parameter 1 and parameter 2, calculate the third point coordinate by the default scale k on the vertical bisector of the line and return the coordinate value.

- getBisectorNormals function (parameter 1, parameter 2, parameter 3):
  According to the steps (2) and (3) described above, the corresponding slave control points are generated by the parameters 1, parameter 2, and parameter 3 coordinates, and the coordinate values are returned.

- getBisectorNormals function (parameter 1, parameter 2, parameter 3, parameter 4):
  According to the Bezier curve algorithm, a cubic Bezier curve is generated from the coordinates of four control points and the coordinate values of each node of the curve are returned.

1. User input control point.
2. When the number of control points input by the user is less than 2, the user is prompted to continue inputting.
3. When the number of control points input by the user is equal to 2, the getThirdPoint function is called, and \( p_0 \) and \( p_1 \) are input to obtain the coordinates of the third point \( p_2 \).
4. When the number of control points input by the user is greater than 2, set the number of input control points to \( n \), connect \( p_0 \) and \( p_2 \), take the midpoint \( p_m \), call the getBisectorNormals function, and input \( p_0, p_1, p_2, \ldots, p_n, p_0, p_1 \) for each of the three points, generate slave control points.
5. Call the getBisectorNormals function, input an array of master-slave control points, generate a Bezier curve for every four points, and finish drawing.

3.2. Linear plotting algorithm based on Centripetal Catmull–Rom curve
The Catmull curve algorithm is easy to appear sharp angle or self-intersection when plotting, as shown in Figure 5, green, red, and blue represent the uniform-CR, centripetal-CR, and chordal-CR curves, respectively, where uniform-CR and Chordal-CR has a certain degree of sharp angle or self-intersection. In order to avoid such defects, this paper uses Centripetal Catmull–Rom curve interpolation algorithm[5] to generate linear plot symbols. The algorithm principle is as follows:

Let \( p = [x \ y]^T \), For the curve segment \( Q_i \) defined by the control points \( p_{t-1}, p_t, p_{t+1}, p_{t+2} \) and the node sequences \( t_{i-1}, t_i, t_{i+1}, t_{i+2} \), can be obtained by equation(1):

\[
Q_i = \frac{t_{i+1} - t_i}{t_{i+1} - t_i} L_{012} + \frac{t_{t_i}}{t_{i+1} - t_i} L_{123}
\]

Among them,

\[
L_{012} = \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-1}} L_{01} + \frac{t_{t_i}}{t_{i+1} - t_i} L_{12}
\]

\[
L_{123} = \frac{t_{i+2} - t_i}{t_{i+2} - t_i} L_{12} + \frac{t_{t_i}}{t_{i+2} - t_i} L_{23}
\]

\[
L_{01} = \frac{t_{t_i}}{t_{t_i} - t_{i-1}} p_{t-1} + \frac{t_{t_i}}{t_{t_i} - t_i} p_i
\]

\[
L_{12} = \frac{t_{i+1} - t_i}{t_{i+1} - t_i} p_i + \frac{t_{t_i}}{t_{t_i} - t_i} p_{t+1}
\]

\[
L_{23} = \frac{t_{i+2} - t_i}{t_{i+2} - t_i} p_{t+1} + \frac{t_{t_i}}{t_{t_i} - t_i} p_{t+2}
\]

Moreover,

\[
t_{i+1} = \sqrt{|p_{t+1} - p_i| + t_i}
\]

And \( t_0=0, t \) is an interpolation variable, its value range is \( t \in [0,1] \), when \( t \) is between 0 and 1, the interpolation point coordinates of the Centripetal Catmull–Rom curve can be calculated.

In order to adapt to the plotting operation, solve the interpolation problem of the first end of the Catmull curve. When the input control points are \( p_1, p_2, p_3, \ldots, p_{n-1}, p_n \), use \( p_1, p_2 \) and \( p_{n-1}, p_n \) to generate control points \( p_0 \) and \( p_{n+1} \) at the beginning and end respectively:

\[
p_0 = p_1 - \eta(p_2 - p_1)
\]

\[
p_{n+1} = p_n - \eta(p_n - p_{n-1})
\]

Where \( \eta \in (0,0.5) \), from which a complete curve through \( p_1 \) to \( p_n \) can be drawn, as shown in Figure 6, \( p_0 \) and \( p_7 \) are generated control points.
The specific program flow of the algorithm is as follows:
Input: control point coordinates
Output: curve node coordinates
Definition:
- `centripetalCatmullRom` function (parameter 1, parameter 2, parameter 3, parameter 4):
  Calculate the coordinates of each node of the Centripetal Catmull–Rom curve according to the algorithm described above, and return the coordinate values.
  1. The user inputs a control point.
  2. When the number of control points input by the user is less than 2, the user is prompted to continue inputting.
  3. When the number of control points input by the user is equal to 2, the first point $p_0$ is calculated using the input $p_1$, $p_2$ points, and the straight line segment $p_1p_2$ is simultaneously generated.
  4. After the user inputs $p_3$, use the input $p_2$, $p_3$ points to calculate $p_4$, call the `centripetalCatmullRom` function, input $p_0$, $p_1$, $p_2$, $p_3$ and $p_1$, $p_2$, $p_3$, $p_4$, respectively, generate a Centripetal Catmull–Rom curve between $p_1p_2$ and $p_2p_3$.
  5. If the control point $p_n$ is continuously input, the curve between $p_{n-2}p_{n-1}$ is updated by $p_n$ input by the user, then use $p_{n-1}$, $p_n$ to calculate the end point $p_{n+1}$, and call the `centripetalCatmullRom` function, input $p_{n-2}$, $p_{n-1}$, $p_n$, $p_{n+1}$ to obtain a curve between $p_{n-1}p_n$.
  6. Repeat step (5) until the user stops input.

4. System implementation

4.1. System architecture
The system is built using the lightweight open source framework Leaflet, which uses SVG technology to draw landscape plotting symbols. Leaflet is a lightweight open source Javascript library that supports HTML5 for developing interactive online map WebGIS applications [9]. The architecture is similar to the open source software OpenLayers, the core framework is the Map class [10], the Map class provides a map container in which to place map controls, add layers, add annotations, add symbols, bind events, etc [11]. Scalable Vector Graphics (SVG) is a markup language based on Extensible Markup Language (XML) for describing two-dimensional vector graphics and vector point matrix blending graphics [12]. SVG images will not be lost in size when zoomed in or resized, and can be created or edited with any text editor [13]. SVG has built-in support for JavaScript [14], which helps solve the problem of interactive operation of WebGIS client maps and improves the interoperability of GIS users [15].
The overall architecture of the system is shown in Figure 7.

![System Architecture Diagram](image)

**Figure 7. System architecture**

The data management part is mainly responsible for loading and displaying the base map data, reading and uploading the sample set of the already drawn situation map, and realizing the specific situation symbol plotting operation, and carries out the size, direction, color, and attribute editing styles such as fill mode and transparency, and adding, deleting, displaying, and hiding the plot layer, the GIS service part provides the basic name address space query function and the printout function of the situation map.

4.2. Experimental evaluation

In order to test the plotting performance and effect of the Centripetal Catmull–Rom curve, this paper compares it with Uniform-CR curve, Chordal-CR curve, Bezier curve, B-spline curve (interpolation, fitting) to generate linear path symbols. Since the path symbol indicates the travel route, in order to reduce the unnecessary consumption and loss in the actual application, the curve generated by the fit is better as the closer to the fold line generated by the control point in the case of ensuring smoothness and continuity. That is, the average residual ω of the fitting accuracy is preferably the smallest, and the calculation formula is equation (10):

\[
\omega = \frac{\sum[(x_i-x_l)^2+(y_i-y_l)^2]}{n}
\]  

(10)

Where \((x_i, y_i)\) is the point coordinate generated by the curve fitting, \((x_l, y_l)\) is the coordinate of the line point generated by the control point connection, \(n\) is the number of feature points generated by the fitting, and the plot of each type of curve is as follows Figure 8 shows:

![Curve Comparison Chart](image)

**Figure 8. Curve comparison chart**

It can be seen from the figure that at the control point 3 and the control point 4, the Uniform CR curve has a sharp angle, at the control point 5, the interpolated B-spline curve appears self-intersection,
and at the control points 6, the Uniform CR curve appears self-intersection. Observe the residual condition of each curve, as shown in Figure 9, take 1000 nodes and calculate the residual of the point on the fold line and the corresponding curve to generate the point. The average residual result is shown in Figure 10:

![Deviation Comparison Chart](image1)

**Figure 9.** Deviation comparison chart

![Average Residual Comparison Chart](image2)

**Figure 10.** Average residual comparison chart

From the comparison of experimental results, the average residual of the linear symbols generated by the Centripetal Catmull–Rom curve algorithm is the smallest, $\omega = 0.0011$, the results show that the curve fits best with its characteristic polygons, in addition, it is noted that the maximum local curvature of the Centripetal Catmull–Rom curve is concentrated at the control points in the experiment, which avoids the occurrence of sharp angle and self-intersection. These features are very suitable for application in situation plotting, and can obtain smooth and stable path curves, its actual plotting effect is shown in Figure 11.

![Actual Plotting Effect](image3)

**Figure 11.** Actual plotting effect

4.3. System implementation

The situation mapping system is widely used in the field of emergency rescue, its main function is to map the physical objects in the target area, the actual situation of the personnel of each department and the current operational situation on the map, so as to better display the situation in the area, assist the commander in decision-making deployment and assist each unit in scheduling cooperation, the overall interface of the system is shown in Figure 12.
The system mainly realizes the functions of scene symbol plotting and editing, map browsing, plotting sample collection and uploading. The situation symbol plotting and editing functions include the drawing of point symbols and function symbols, the adjustment of the size direction, the selection of style attributes, and the addition of text annotations. The map browsing includes the selection of the base map, the display and hiding of the plot layer, the zooming of the map, the roaming, and the basic place name address query function. The sample set is saved in the upload function. The drawn situation map can be uploaded to the server for saving and reading, as shown in Figure 13.

5. Conclusion
This paper builds a lightweight, efficient, scalable and cross-platform situation plotting system. The Web-based situational plotting system can be operated and used on various platforms, eliminating the need for complicated installation procedures. Vector graphics drawn with SVG technology can better display situational plotting symbols on the map at multiple scales. The generated symbol framework can be easily extended to other applications, assist the command personnel to accurately understand the situation at the scene and make scheduling decisions in a timely manner. The problem of the traditional linear plotting symbol is improved by using the Centripetal Catmull–Rom curve,
which reduces the number of control points generated, makes the curve closer to the actual path, and improves the plotting efficiency.

Acknowledgments
The work described in this paper was fully supported by a grant from the National Natural Science Foundation of China (No.41871284).

References
[1] Zhao W N 2017 Design and Implementation of a Situation Symbol Plotting System Based on OpenGL Chinese Academy of Sciences University
[2] Duan Z Y, Wang N and Yang X 2016 Research on an Improved B-Spline Curve Fitting Algorithm Mech. Desi. Manu. 23 pp 17-19
[3] Chen H Y and Jiang Y 2017 Curve approximation algorithm based on quadratic B-spline Comp. Engi. Sci. 39(8) pp 1489-1494
[4] Chen H Q, Li F X and Zhang Q Y 2012 A Progressive Rendering Method for Smooth Trajectories J. Syst. Simu. 24(2) pp 357-360
[5] Tan J H and Acharya U R 2014 Active spline model: A shape based model interactive segmentation Dig. Sign. Proc. 35 pp 64-74
[6] Yuksel C , Schaefer S and Keyser J 2011 Parameterization and applications of Catmull–Rom curves Comp. Aid. Desi. 43(7) pp 747-755.
[7] Liu H, Cao W and Zhao W J 2013 Earthquake Disaster Situation Plotting Technology Based on 3DGIS Bul. Surv. Map. 9 pp 75-79
[8] Zhu W J, Lu W J and Li Y 2018 Design and Implementation of Online Emergency Plot Symbol Based on SVG J. Geom. 43(5) 96-99
[9] Leaflet 2019 An Open-source Javascript Library for Mobile-friendly Interactive Maps http://leafletjs.com
[10] Hu T D and Hu Q W, 2015 Development of cross-platform map client based on Open Source Sci. Surv. Map. 40(7) pp 142-145
[11] Zhao H C and Qu Chao 2017 Design and Implementation of WebGIS Client Based on Open Source Leaflet Tec. Inno. Appl. 16 pp 56-57
[12] Wang W H and Wang W F 2008 Study and realization of SVG-based WebGIS Comp. Engi. Desi. 29(22) pp 5706-5708
[13] Yang J Z, Qiu C T and Chen P 2018 The graphics conversion of MapGIS format based on SVG Geop. Geoc. Expl 42(5) pp 1069-1075
[14] Han S W 2011 The realization of advanced search fuctions on SVG-Based vector WebGIS mapIndu. Inst. Auto. 4 68-72
[15] Liu D, Li G and Tian Y Z 2014 Algorithm of military arrow symbol plotting based on VML and SVG Sci. Surv. Map. 39(4) pp 115-118