Vector Circle Clipping Algorithm Based on Polygon Window of Hexagonal Grid System

Minghua Cao¹, Heng Zhang², Changjie Zhou³, Yi Sun⁴ and Hui Yu⁵

¹,²,³,⁴,⁵ Department of Computer Science and Technology, Beijing University of Posts and Telecommunications, Faculty of Computer Science, NO. 10, Xitucheng Road, Haidian District, Beijing, China.
⁵Author to whom any correspondence should be addressed, China waterborne transport research institute, NO. 8, Xitucheng Road, Haidian District, Beijing, China.
Email: yuhui@wti.ac.cn, minghuacao@bupt.edu.cn

Abstract. This paper proposes an algorithm for cropping vector circles in polygon window under hexagonal grid system for the problem of long time consuming and low efficiency of polygon window clipping algorithm in existing Non-Cartesian coordinate system. First, the algorithm converts Cartesian coordinates into hexagonal grid coordinates, then performs intersection detection by vector operation, and uses geometric relations to judge the virtual real intersection points, thus obtaining the coordinates of the real intersection points, and thus cutting the arc results. Finally, the final cropping result is obtained. The experimental results show that the algorithm is an effective vector circle clipping algorithm based on the hexagonal grid system polygon window, and the algorithm has high efficiency and stability.

1. Introduction
The circular cropping algorithm is an important algorithm in the fields of computer graphics [1], two-dimensional computer animation and robot kinematics. Currently, the existing circular cropping algorithms are mostly used for cropping in a rectangular coordinate system using a rectangular window [2]. However, the use of polygons for cropping [3] is rare, and the algorithm using hexagonal grid coordinates [4] is even rarer. And processing methods in the past have a great number of shortcomings, especially the use of polygon windows for circles. In the case of cropping, this has led to deviations in some areas of application. To this end, this paper proposes an effective solution for cropping vector circles in a polygonal window under a hexagonal grid system. Through the coordinate transformation, the coordinates of the hexagonal grid are determined. In order to improve the efficiency of the algorithm, the method of converting each line segment of the circle and the polygon into a vector is used [5], and then the vector operation is performed to obtain the result of the intersection detection, thereby obtaining the virtual intersection and real intersection. And then find the coordinates of the real intersection point, by cutting the result of arc, the algorithm is obtained, and the purpose of the vector circle is achieved.

2. Algorithm in This Paper

2.1. Analysis of the Relationship Between the Position of a Polygon Window and a Circle
Considering the polygon window as multiple line segments, you can define the positional relationship between the polygon and the circle as follows:
Definition 1. If any line segment of the polygon window intersects the circle, the circle and the polygon are intersecting.

Definition 2. If the polygon window does not intersect the circle, and any line segment is tangent to the circle, and the circle is not inside the polygon, the circle and the polygon are circumscribed.

Definition 3. If the polygon window does not intersect the circle, and any line segment is tangent to the circle, and the circle is inside the polygon, the circle and the polygon are inscribed.

Definition 4. If any line segment of a polygon does not intersect the circle or the circle and the circle is not inside the polygon, the circle, the polygon and the polygon are separated.

Definition 5. If any line segment of a polygon does not intersect the circle or is tangent to the circle and the circle is inside the polygon, the circle and the polygon are separated.

If the polygon and the circle are separated or circumscribed, no result can be cropped; if the polygon and the circle are separated or inscribed, the result of the clipping is the entire circle [6]. Therefore, the algorithm of this paper will analyze and verify the intersection state as the object.

2.2 Coordinate System Conversion
Conversion using the coordinate transformation method proposed in [7]:

\[
\begin{align*}
X &= x + \frac{y'}{2} \\
Y &= \frac{\sqrt{3}}{2} \cdot y
\end{align*}
\] (1)

Set the coordinates of the vertex of the polygon: \(P = (P_0, P_1, P_2, \ldots, P_{n-1}, P_n)\) among them \(P_i = (x_i, y_i)\) \(i = 0, 1, 2, \ldots, n\).

Assumed directed edges: \(\overrightarrow{e_i} = \overrightarrow{P_iP_{i+1}}\), \(i = 0, 1, 2, \ldots, n-1\).

Assume that the coordinates of the center of the Cartesian coordinate system are: \((x_c, y_c)\), radius is \(R\).

Let any two points under the hexagonal grid system be \(P_i(x_i, y_i)\) and \(P_j(x_j, y_j)\)

Then the coordinate formula of any point under the hexagonal grid system is:

\[
Q_i = Q(x_i + \frac{y_i}{2}, \sqrt{3} \cdot \frac{y_i}{2})
\] (2)

Extend the generalized distance formula [8] under the hexagonal grid system, Obtained by the cosine formula:

\[
(x_2 - x_1)^2 + (y_2 - y_1)^2 - 2(x_2 - x_1)(y_2 - y_1) \cos \frac{2\pi}{3}
\]

Simplify the generalized distance formula:

\[
d_{ij}^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (x_2 - x_1)(y_2 - y_1)
\] (3)

A circle is defined as a collection of points in the plane that are at the same distance from the fixed point. Let \(d=R\), Get the equation of the circle:

\[
(x-x_c)^2 + (y-y_c)^2 + (x-x_c)(y-y_c) = R^2
\]

2.3 Length Formula
Let \(L_i = \overline{PP_{i+1}}\). Known by (2) and (3):

2
\[ L_i = \left( (x_{i+1} - x_i) + \frac{1}{2} (y_{i+1} - y_i) \right)^2 + \left( \frac{\sqrt{3}}{2} (y_{i+1} - y_i) \right)^2 + \left( (x_{i+1} - x_i) + \frac{1}{2} (y_{i+1} - y_i) \right)^2 \]

Rewrite the ray function [5]:
\[ \overrightarrow{OP_i}(t) = \overrightarrow{OP_i} + t\overrightarrow{d} \]  

among them ‘t’ is truncation factor, \( t \in [0, L_i] \),
\( \overrightarrow{d} \) is the unit vector of \( \overrightarrow{P_iP_{i+1}} \),
\[ \overrightarrow{d} = \left( \frac{x_{i+1} + \frac{y_{i+1}}{2} - x_i - \frac{y_i}{2}}{L_i}, \frac{\sqrt{3}}{2} (y_{i+1} - y_i) \right) \]

2.4. Intersection Detection, Virtual and Real Intersection Judgment and Real Intersection Solution

Let \( \overrightarrow{c} = \overrightarrow{PO_i} \), then the projection of \( \overrightarrow{e} \) in the \( \overrightarrow{P_iP_{i+1}} \) direction is \( \overrightarrow{a} \), which is \( \overrightarrow{a} = \overrightarrow{e} \cdot \overrightarrow{d} \), \( \overrightarrow{d} \) is unit vector.

Let the intersection of \( \overrightarrow{P_iP_{i+1}} \) and the circle is \( T_k \), the projection end point of \( \overrightarrow{e} \) in the \( \overrightarrow{P_iP_{i+1}} \) direction is \( \overrightarrow{A} \), let \( M = (r^2 + a^2 - |\overrightarrow{e}|^2) \), \( b = |\overrightarrow{OA}| \), then separately use Cosine theorem in \( \Delta AOT_k \) and \( \Delta AOP_i \):

\[
\begin{align*}
&\begin{cases}
  f^2 + b^2 - 2bf \cos(\pi - \frac{\pi}{3}) = R^2 \\
  b^2 + a^2 - 2ab \cos(\pi - \frac{\pi}{3}) = |\overrightarrow{e}|^2
\end{cases}
\end{align*}
\]

Obtain:
\[ b = \frac{R^2 + a^2 - f^2 - |\overrightarrow{e}|^2}{f - a} \]

Eliminate \( b \) into a standard equation:
\[ f^4 - af^3 + (|\overrightarrow{e}|^2 - 2r^2) f^2 + a(|\overrightarrow{e}|^2 + r^2 - a^2) f + M - a^2r^2 = 0 \]  

As can be seen from Figure 1, when \( b < R \), \( \overrightarrow{P_iP_{i+1}} \) intersects the circle:
\[ \frac{R^2 + a^2 - f^2 - |\overrightarrow{e}|^2}{f - a} < R \]

Obtain:
\[ f_1 < f < f_2 \]  

If the condition (6) is satisfied, there is an intersection and the intersection detection is completed.

As can be seen from Figure 1, virtual and real intersection judgment [6] can be defined as: \( \overrightarrow{P_iP_{i+1}} \) and circle intersection is \( T_i \), which is the real intersection, \( \overrightarrow{P_iP_{i+1}} \) extension cord and circle intersection is \( V_i \), which is virtual intersection. Let \( t' = a^2 - f^2 \), if \( t' \) is negative, it means no real intersection here; if \( t' \) is positive, it means it is a real intersection. And whether the real intersection exists depends on the truncation factor ‘t’, if and only if \( t = a \pm f \), There is a real intersection.
Let $t = a \pm f$, among them the value of ‘f’ can be obtained by (5), further, the value of t can be obtained., finally use (4) to find the coordinates of $T_i$.

2.5. Cropping Result

Set $T_i$ are the intersection of each line segment of the polygon window and the circle and clockwise Sorting, among them $i = 0, 1, 2, \cdots, k, 0 \leq k \leq 2(n - 1)$

Definition 6. Assume that the projection point of point S to the target straight line $l$ is $S'$, then $S'$ meet the condition of $<SS', l> = \frac{\pi}{3}$, then $S'$ is called the hexagonal projection point of S. This projection method is called hexagonal projection.

As shown in Figure 2, let $D(x_d, y_d)$ is hexagonal projection in the line $T_iT_{i+1}$, the point of hexagonal projection in the arc $T_iT_{i+1}$ is $E(x_e, y_e)$, From (1), definition 6 and the midpoint coordinate formula are easy to get: O, D, E three points are on the same line.

Let A be the hexagonal projection point of D on the x-axis, and B be the hexagonal projection point of E on the x-axis, then because $\Delta OAD$ and $\Delta OBE$ are similar triangles.

If $\overrightarrow{OT_i} \times \overrightarrow{OT_{i+1}} \leq 0$, then $\overrightarrow{T_iT_{i+1}}$ is excellent arc, obtain:

$$\frac{x_e - x_d}{x_e - x_d} = \frac{y_e - y_d}{y_e - y_d} = \frac{|OD|}{|OE|}$$  \hspace{1cm} (7)

If $\overrightarrow{OT_i} \times \overrightarrow{OT_{i+1}} \leq 0$, then $\overrightarrow{T_iT_{i+1}}$ is inferior arc, obtain:

$$\frac{x_e - x_d}{x_e - x_d} = \frac{y_e - y_d}{y_e - y_d} = \frac{|OD|}{|OE|}$$  \hspace{1cm} (8)

And $|OE| = R$, solving (7) obtains:

$$\begin{align*}
x_e &= \frac{(x_e - x_d) \cdot R}{|OD|} + x_e \\
y_e &= \frac{(y_e - y_d) \cdot R}{|OD|} + y_e
\end{align*}$$

solving (8):

$$\begin{align*}
x_e &= \frac{(x_d - x_e) \cdot R}{|OD|} + x_e \\
y_e &= \frac{(y_d - y_e) \cdot R}{|OD|} + y_e
\end{align*}$$

If $E(x_e, y_e)$ is inside the polygon window, then the arc is clipped, otherwise, the arc is discarded.

3. Analysis of Results

3.1. Solving Equations (5)

MATLAB code as follows:

```matlab
syms a e r f;
y=f^4-a*f^3+(e^2-2*a*r^2)*f^2+(a*r^2-2*a^3+a*e^2)*f+(r^2+a^2-e^2)*2-a^2*r^2;
```
\[ f = \sqrt{z^4 - a z^3 + z^2 (|\vec{e}|^2 - 2 r^2) + z (a |\vec{e}|^2 + a r^2 - a^3) - a^2 r^2 + M} \]

The result is: among them, \( z = 1,2,3,4 \)

3.2. Solving Equations (6)
MATLAB code as follows:

```matlab
syms r a e f;
y = r^2 + a^2 - e^2 - f^2 - r^2*(f-a);
solve(y,f)
```

the result is:

\[
\begin{align*}
f_1 &= \frac{-\sqrt{(4 a^2 + 4 a R^2 - 4 |\vec{e}|^2 + R^4 + 4 R^2) - R^2}}{2} \\
f_2 &= \frac{\sqrt{(4 a^2 + 4 a R^2 - 4 |\vec{e}|^2 + R^4 + 4 R^2) - R^2}}{2}
\end{align*}
\]

If \( f \) satisfies the condition of \( f_1 < f < f_2 \), then the polygon intersects the circle, the intersection point is obtained by (4), among them: \( t = a \pm f \)

4. Conclusion
By analyzing various algorithms for cropping circles under polygon window and the characteristics of hexagonal grid systems, this paper proposes a kind of polygon window clipping vector circle, which is suitable for hexagonal grid systems. Firstly, the position relationship of the vector circle and polygon window is analyzed, then the coordinate transformation is performed, and then the distance between any two points is obtained, and then the intersection detection is performed, and the coordinates of the real intersection point are obtained by using the ray function and the geometric relationship, finally, by comparing the relationship between the superior arc and the inferior arc, The obtained arc is cut and the result is selected. It can usually be applied to the model making of water conservancy traffic.

Follow-up infrastructure dredging of the second phase of the 12.5-meter deep-water channel below the Yangtze River Nanjing and dredging period dredging maintenance dredging project (I section) navigation safety assessment project.

5. Figures

![Intersection detection](image1.png) ![Cropping result](image2.png)

**Figure 1.** Intersection detection.  
**Figure 2.** Cropping result.

6. References
[1] Jiaguang Sun. Computer Graphics [M].Beijing: Tsinghua university press, 1998.
[2] WANG Rui, YAN Xiao-min, TANG Di. A New Algorithm for Circle Clipping against Rectangular Window Based on Encoding Approach [J]. JOURNAL OF LIAONING UNIVERSITY (NATURAL SCIENCE EDITION). 2011, 38(2): 177-180.

[3] Zhang Yingnan. The Design and Implementation of the Line and Circle Clipping Algorithms against the Arbitrary Polygon Boundary [J]. Electronic Test. 2016, (5):16-18.

[4] SUN Chang-song, LI Li-jie, SONG Yang. A line-clipping algorithm for circle clip window based on hexagonal grid [J]. Technology & Economy in Areas of Communications. 2006, (1):78-80.

[5] Miao Yongchun, Tang Quanhua. Vector Circular Clipping Algorithm against Arbitrary Polygon Window [J]. Journal of Computer-Aided Design & Computer Graphics. 2016, 28(9): 1451-1458

[6] Donald Hearn, M Baker. Computer Graphics with OpenGL [M]. Prentice Hall, 2003.

[7] Liu Yongkui, Shi Jiaoying. Display of Graphics on Hexagonal Grids [J]. Journal of Computer-Aided Design & Computer Graphics. 2004, 16(3): 331-336

[8] Lu Guodong, Xing Junwei, Tan Jianrong. New Clipping Algorithm of Line against Circular Window with Multi-encoding Approach [J]. Journal of Computer-Aided Design & Computer Graphics. 2002, 14(12): 1133-1137

[9] Weiler K, Atherton P. Hidden surface removal using polygonarea sorting [J]. ACM SIGGRAPH Computer Graphics, 1977,11(2): 214-222

[10] Liang Y D, Brasky B A. A New Concept and Method for Line Capoing [J]. ACM Transactions on Graphics, 1984,3(1):1—22.

[11] Ping Zhao, Feng Chun, Bolin Li. Efficient Algorithm for Line Clipping against General Polygon [J]. JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY, 2004, 39(1):64-68.