 Vertex Chain Code for Rectangular and Triangular Cell

Lili Ayu Wulandhari
Department of Modeling and Industrial Computing
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor Bahru, Malaysia
Phone: +60143862335
lili.wulandhari@gmail.com

Habibollah Haron
Department of Modeling and Industrial Computing
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor Bahru, Malaysia
Phone: +607-5532012
habib@utm.my

ABSTRACT
Image representation always becomes an important and interesting topic in image processing and pattern recognition. Since introduced by Freeman in 1961, that is known as Freeman Chain Code (FCC) the evolution and improvement of chain code representation scheme has been widely used as a topic of research. In 1999, Bribiesca introduced a new two-dimensional chain code scheme that is called Vertex Chain Code (VCC). An element of this chain indicates the number of cell vertices, which are in touch with the bounding contour of the shape in that element position. The VCC is composed of three regular cells, namely rectangular, triangular, and hexagonal. This paper covers two of three cells; rectangular and triangular one. This research purpose the cell-representing and transcribing algorithm, which visualizes the thinned binary image into the VCC-cells and then transcribes them into vertex chain code.

Keywords
Vertex Chain Code, Rectangular Cell, Triangular cell, Cell-representation, Transcribing algorithm

1. INTRODUCTION
Nowadays computer utilization in image analysis is regarded important. Accordingly, representation an image in analysis also becomes an interested case to be discussed. One of the representation ways is chain code. Chain code is widely used because it preserves information and allows considerable data reduction. Since introduced by Freeman [1], chain code development increases rapidly. One of the chain code beside Freeman chain code is a chain code that introduced by Bribiesca [2], known as Vertex Chain Code (VCC). Vertex chain code is a result of work that is motivated by the idea to obtain various shape features computed directly without going to coordinate representation.

Vertex chain code is one of the chain codes that preserve 3 objectives of coding scheme for line structure that was stated by Freeman [3]. The three objectives are: (1) it must faithfully preserve the information of interest, (2) it must permit compact storage and convenient for display, (3) it must facilitate any required processing. The three objectives are somewhat in conflict with each other, and any code necessarily involves a compromise among them.

Vertex chain code has some important characteristic [2]; (1) The VCC is invariant under translation and rotation and optionally may be invariant under starting point and mirroring transformation, (2) Using the VCC it is possible to represent shapes composed of triangular, rectangular and hexagonal cells, (3) The chain elements represent real value not symbols such as other chain code, are part of the shape indicate the number of cells vertices of the contour nodes, may be operated for extracting interesting shape properties (Figure 1), (4) Using VCC it is possible to obtain relations between the bounding contour and the interior shape. Vertex chain code has some special characteristic that interested to be observed.

Figure 1. The Example of Rectangular-VCC

This paper proposes the algorithm for getting vertex chain code from rectangular and triangular cell-representation from thinned binary image. This paper is consist of methodology of the research, development of the algorithm, experimental results, conclusion and future work that is presented in section 2, 3, 4, and 5 respectively.
2. METHODOLOGY

This chapter describes about methodology of this research. First, it explains about problem identification, problem identification is known by studying some literature about the previous research. Second it is about data source or data definition. Next section describes the development of the algorithm and the last explain about how this algorithm is implemented. Figure 2 shows the chart of the research methodology.

Problem identification is the part for identifying what are the questions that provide the basis for algorithm building. This identification is started from literature review about chain code. From this literature review, it is known the development of chain code, especially vertex chain code. So that it is found some gaps Based on the literature review that has done, it is got that development of vertex chain code increases rapidly. There are many new chain codes created. Most of them are created based on freeman chain code or rectangular vertex chain code. It is rare to use the others cells of vertex chain code, i.e. triangular and hexagonal cells. This research will cover all of vertex chain code cells to represent an image.

In this research it will represent a thinned binary image into vertex chain code cells and then transcribe them into vertex chain code, which both of these processes are called the mapping process. The mapping algorithm that consists of cell-representation and transcribing algorithm will be created. The mapping algorithm has input thinned binary image. This input is used in validating the algorithm. The data of input that is presented in the algorithm will be explained in the next section. This algorithm will be implemented in MATLAB programming, so this algorithm can use easily and automatically.

For validation, it is also involved the data set. Data that is used in this research are the data in thinned binary image. These data are also used in previous work about vertex chain code. These previous data is useful to validate the validity of the exits algorithm. Without these data it is difficult to know validation of the algorithm.

Thinned binary image is obtained from binary image. Binary image is the image that is built by 1 and 0. Usually a value of 1 is for white and 0 for black. In the simple case a binary image is displayed as an object and background. Value of 1 is assigned for the object and 0 for the background. In thinned binary image the boundary of an image usually is signed by 1, and the others including inner and outer of the boundary is signed by 0. Thinned binary image is got by using thinning algorithm. One of thinning algorithm was proposed by Haron [4]. Figure 3 shows the binary image and its thinned one.

The Development of the algorithm and the experimental result will be explained completely in section 3 and 4.
3. THE ALGORITHM DEVELOPMENT

This paper is covered two from three cell of vertex chain code that Bribiesca [2] propose, namely rectangular and triangular cell. This section will describe the algorithm for representing the thinned binary image into rectangular and triangular cells, and then transcribing it into vertex chain code.

3.1 Rectangular-VCC Algorithm

The first cell for the algorithm is rectangular cell. Before the explanation of the algorithm, it is important to know about the rectangular characteristics. The rectangular cell is the common cell that is always used. It is simple and easy to work with. They are easy to locate in Cartesian coordinates (x, y), and the axes are orthogonal. Each rectangular vertex is represented either point x or y in coordinates Cartesian. It simplifies to represent a thinned binary image as an input into the cell, because there are not changes and translation of the coordinates from thinned binary image into the cell-representation.

![Figure 4. Rectangular in Coordinate System](image)

Each faces of rectangular has four edges, thus we expect 4 times as many as faces as amount of edges. However each edge is shared by two faces, so the amount of edge is formulated below

\[
\text{edge} = 4n - m
\]

Where \(n\) = number of rectangular, \(m\) = number of shared edge.

Each face also has 4 vertices. Every vertex can be shared by maximum 4 faces. The shared vertex by two faces is given variable 1, continuously variable 2 and 3. The formulation is shown below

\[
\text{vertex} = 4n - (3v_4 + 2v_3 + v_2)
\]

The \(v_4, v_3, v_2\) are the shared vertex by four, three, and two faces respectively. The relationship between face, vertex, and edge will be important in designing the coordinates system for the cell.

3.1.1 The Rectangular Cell-Representation Algorithm

The cell-representation algorithm of Rectangular-VCC is an algorithm that represents a thinned binary image as rectangular cells. The algorithm has thinned binary images as input. Each code 1 in the thinned binary image represents each face of the rectangular cell. The direction of code 1 adjacent to another code 1 leads to the formation of the next rectangle. The relation among the code is configured based on eight directions connectivity (Table 1). This eight direction connectivity is opposite from the coordinate system, thus the cell-representation of the image is also opposite from the original one.

![Table 1. Eight Direction Connectivity](image)

![Table 2. Representation of Rectangular Cells](image)

Representation of the thinned binary image into the rectangular cells is distinguished by the position of the code in direction connectivity. Here the image is divided into 2 part of area, namely left or right. Especially for position by condition \((row, column) = (row+1, column+1)\) and \((row, column) = (row+1, column-1)\), there is different representation if it lies on either right or left. But for the other position in connectivity there are not any differences. Table 2 shows the representation of Rectangular-VCC formatted by the direction of code 1 adjacent to another code 1. When each code in the binary image is represent to the cell, a line drawing consisting of rectangle cells will be created.

![Table 2. Representation of Rectangular Cells](image)

Based on the representation of the rectangular cells from thinned binary image, the cell-representation algorithm is created. In this algorithm a thinned binary image is represented as an array which has \(n\) rows and \(m\) columns.

Rectangular of Cell Representation Algorithm:
1. Input : Thinned Binary Image (TBI)
2. Check number of rows and columns
3. Divide the image area becomes 2 parts; right and left
4. Scan the thinned binary image from the top to the bottom, left to the right.
If it finds code 1 and satisfy:
- TBI (row, column) = TBI (row, column+1) = 1, or
- TBI (row, column) = TBI (row+1, column) = 1, or
- TBI (row, column) = TBI (row +1, column–1) = 1, or
- TBI (row, column) = TBI (row+1, column+1) = 1
Draw the segment of the cells according to Table 4.1, start from vertex of TBI (column, row).
5. Do Step 4 until all of code 1 from the thinned binary image is represented on rectangular cells.

The results from this represented will be used to extract the vertex chain code from thinned binary image that is described in the next section.

3.1.2 The Rectangular Transcribing Algorithm

The transcribing algorithm is an algorithm that transcribes the rectangular cells into vertex chain code. In fact vertex chain code shows number of rectangle that in touch with a corner of the rectangle. But for the transcribing algorithm, it proposes to see the degree of either rectangular corner or relation between two edges of the rectangular. Rectangular cell has three kinds vertex chain code; 1, 2, and 3. It represents three degree; 270°, 180°, and 90° (Figure 5).

![Figure 5. Vertex Chan Code From Rectangular Cells](image)

For transcribing algorithm it is invariant under starting point, it could start from arbitrarily point, continuously in clockwise direction.

The Transcribing Algorithm
1. Input: Boundary of thinned binary image Cell-representation
2. Determine the starting point
3. Scan the boundary
   if it is found the 90° then VCC=3
   else if it is found 180° then VCC=2
   else
      VCC=1
   end
4. Do step 3 until the starting point is reached back.

3.2 Triangular –VCC Algorithm

The second cell for vertex chain code is triangular cell. It is also one of the basic cells in digital geometry. The structure of the triangular cell is much more complicated than rectangular one [5]. The triangular elements have alternating orientation, so the arrangement of peripheral elements will vary accordingly.

It is important to establish the neighborhood connectivity for both pattern and its background. But before understanding about neighborhood connectivity, it is necessary to know that there are differences of △ and ▽ triangular representation especially in cell-representation (Figure 6).

![Figure 6. Triangular Representations](image)

The nearest neighbour of both △ and ▽ can be divided into three sets such that all the elements within each set are equidistant from the central element, distances being measured from centroid. The first set is the triangular neighbourhood that has distance 2 units away from the central element’s centroid. The second set contains the six elements which has distance $2\sqrt{3}$ unit away, and the third contains the three elements which are furthermore away at a distance 4 units (Figure 7).

![Figure 7. Three sets of the triangular nearest neighbor](image)

If the edge of the triangle = $a$, then each triangle has 1-neighbours that has distance $\frac{1}{3}a\sqrt{3}$, 2-neighbours that has distance , and 3-neighbours that has distance $\frac{2}{3}a\sqrt{3}$. It means each triangle has three 1-neighbours, nine 2-neighbours (three 1-neighbours, and six 2-neighbours), and twelve 3-neighbours (nine 2-neighbours and three 3-neighbours). Therefore the triangular is knows has 3, 9 and 12 neighborhood connectivity [5].

Triangular cell representation uses different coordinate system with usual one. Here it is used three coordinates value system [6]. It is different with three coordinate value in $Z^3$ that is independent; while three coordinate value in triangular cell is non independent. It is important to use three coordinates value for triangular cell because its symmetric properties and it is main tool to get concrete results.
Figure 8. Coordinate Values on the Triangular Cell

In this coordinate system the triangle is called point. Procedure of the triangular cell coordinate is given below [7].

Procedure: First we choose a point for the origin, which coordinate values are (0,0,0). We choose a line cross through the centre of the origin triangle, which is perpendicular to a side of the triangle. Let this line be fixed as the coordinate axis x, with the direction 0°. Let the coordinate axes y, and z also cross the centre of the origin triangle and their direction be fixed as 120°, and 240°, respectively. We assign the coordinate values to the points inductively. If the coordinate values of a triangle A are known, than the coordinates of those three triangles which have a common side with A, can be calculated as follows; Let us fix one of these three triangles, and denote it by B. The common side of A and B are perpendicular to one of the coordinate axis. According to the direction of this axis, we increase or decrease the corresponding coordinate value of B by 1 (Figure 8).

By using the procedure of the coordinate system, it is defined the parity of the points in the following way: if the sum of the coordinate values is zero, it is called the point even, otherwise it is odd. Even and odd triangles are the shape △ and ▽, respectively.

As we can see the definition of neighbors (Figure 7) fits well the coordinate procedure. This thesis is used the 12-neighbour connectivity for cell representation of thinned binary image.

3.2.1 Triangular Cell-Representation Algorithm
As the previous explanation, triangular cell is more complicated than the rectangular one. For rectangular cell it is not difficult to represent a thinned binary image to rectangular cells. But in triangular cell there are some rules that have to be noticed for thinned binary image representation. It is brought on by a condition as follows: It is represented a thinned binary image that used two values coordinate x and y. It is caused not every coordinate point is lied in the corner of the triangular, while every cell must be started from the corner, not in the middle of the triangle edge.

Therefore, to avoid starting point in the middle of triangle edge, it is important to commit a step as follows: In this triangular cell system, every y = odd point. It causes every x point is in the middle of triangle edge. So that every cell that is started from this (x, y) point must be moved 0.5 point to the left. The cell-representation completely is given below (Table 3). It depends on the shape of triangle; △ or ▽. And the representation is distinguished by where its position, in the right or left of the triangle.

Table 3. Triangular Cell Representation

<table>
<thead>
<tr>
<th>Right</th>
<th>Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The Triangular cell-representation is created based on the table 3. This algorithm has input a thinned binary image that is defined as an array variable in MATLAB. The cell representation is moved according to clockwise manner. The algorithm is given below.

Triangular Cell-Representation Algorithm:
1. Input: Thinned Binary Image (TBI)
2. Check number of TBI rows and columns
3. Divide the area of the image into 2 parts; left and right
4. Scan the thinned binary image from the top to the bottom, left to the right.

If it finds code 1 and satisfy:
- TBI (row, column) = TBI (row, column+1) = 1, or
- TBI (row, column) = TBI (row+1, column) = 1, or
- TBI (row, column) = TBI (row+1, column-1) = 1, or
- TBI (row, column) = TBI (row+1, column+1) = 1
Draw the segment of the cells according to Table 5.1, start from vertex of TBI (column, row). Do Step 4 until all of code 1 from the thinned binary image is represented on triangular cells.

However the triangular cell-representation algorithm is similar with the rectangular one, but triangular cell is more complicated.

It is important to be remarked that there is movement in the coordinate if the point touches the middle of triangle edge.

3.2.2 The Triangular Transcribing Algorithm

Transcribing algorithm is functioned to transcribe the triangular cell into vertex chain code. Transcribing algorithm of triangular cell is not as complicated as the cell-representation one. If the cell representation has been got, the vertex chain code of the thinned binary can be obtained. The algorithm is created based on the figure 9 that shown below.

Figure 9. Vertex Chain Code of Triangular Cell

Triangular cell has 5 different codes; 1, 2, 3, 4, and 5. It depends on the corner that is formed by each triangle in the cell. The algorithm to transcribe the triangular cell into vertex chain code is given below. The code is start from starting point that is chosen continuously in clockwise direction.

The Triangular Transcribing Algorithm

1. Input: Boundary of thinned binary image Cell-representation
2. Determine the starting point
3. Scan the boundary
   if it is found the 180° then VCC=3
   else if it is found 240° then VCC=2
   else if it is found 300° then VCC=1
   else if it is found 60° then VCC=5
   else
4. Do step 3 until the starting point is reached back.

The result of the algorithm will be shown in the next section in experimental results.

4. EXPERIMENTAL RESULTS

The experiment is done for obtaining the validation from an algorithm. For this experiment the algorithm is implemented in a system by using MATLAB language programming. Some example from thinned binary image as the input, the cell representation and the vertex chain code from the rectangular triangular cell is shown in figure 10 below.
The thinned binary image is saved as an array variable. As the input the thinned binary image, each rows and columns of the thinned binary image will be scanned for finding code 1, and represent it into the rectangular or triangular cell. The result from the cell is transcribed into vertex chain code according to the cell code. By doing this experimental it is proved that the algorithm is valid for representing the thinned binary image into the rectangular or triangular cell and transcribing it into vertex chain code.

5. CONCULSION AND FUTURE WORK
5.1 Conclusion
From some ways to represent an image digitally, vertex chain code could to be one of alternative ways. Vertex chain code could be a choice because vertex chain code was proposed to obtain various shape features computed directly without going to coordinate representation.

Vertex chain code covers three basic cells, rectangular, triangular and hexagonal. This paper presented two of them, namely rectangular and triangular cell. Rectangular cell has its own code; 1, 2, and 3, while triangular has code 1, 2, 3, 4, and 5.

Rectangular is one of the basic cells that are simple and easy to use it. For triangular cell, it is needed some movements to represent a thinned binary image into the cell, to keep the representation is started form the middle of the triangular cell edge.

The algorithm that divided into cell-representation algorithm and transcribing algorithm has been tested and validated by using tinned binary image as the input, and has been implemented in MATLAB language programming.
5.2 Future Work
The vertex chain code that was proposed by Bribiesca just limited to the shape without holes. Whereas there are many application that is needed the concept for shape with holes, such as handwriting recognition. Therefore it is important to develop the vertex chain code concept that covers the shape with holes too.

REFERENCES