# Object Filling Using Table Based Boundary Tracking

**Dr. Mokhtar M. Hasan**
mmwaeli@gmail.com
University of Baghdad -College of Science for Women - Computer Science Dept.

**Abstract**

The feature extraction step plays major role for proper object classification and recognition, this step depends mainly on correct object detection in the given scene, the object detection algorithms may result with some noises that affect the final object shape, a novel approach is introduced in this paper for filling the holes in that object for better object detection and for correct feature extraction, this method is based on the hole definition which is the black pixel surrounded by a connected boundary region, and hence trying to find a connected contour region that surrounds the background pixel using roadmap racing algorithm, the method shows a good results in 2D space objects.

**Keywords:** object filling, object detection, object holes, noise removal, scanning 2D objects.

## املاء الاشكال باستخدام طريقة الجدول لتتبع الاطار

**د.مختار محمد حسن**
جامعة بغداد - كلية العلوم للبنات - قسم علم الحاسوب

**الخلاصة**

من اهم خطوات التصنيف الصحيح للاشكال وتمييزها هو كيفية استخلاص الخواص المميزه لها، هذه الخطوة تعتمد بشكل اساسي على التحديد الدقيق للشكل في المنظور المعطى، خزارزميات تحديد الاشكال ممكن ان ينتج عنها بعض الضوضاء الغير مرغوب بها كمحصلة لعمل هذا الخوارزميات والتي تؤثر سلبا على الشكل النهائي المحدد، في ورقة البحث هذه، تم استحداث طريقة جديدة لاملاء الفراغات الموجودة في الاشكال المدخلة والذي يؤثر ايجابا في تحديد الشكل المطلوب بشكل ادق وبالتالي استخلاص خواص مميزه لهذا الشكل، حيث ان تعريف الفراغ المعتمد هنا هو النقاط ذات اللون الاسود المحاطة بمنطقة متصلة وهو اساس عمل الخوارزمية التي تم تصميمها في هذا البحث باستعمال تتبع خارطة الطريق، وقد تم الحصول على نتائج جيدة جدا للاشكال ذات البعد الثنائي.

**كلمات مفتاحية:** املاء الاشكال، تحديد الاشكال، فراغات الاشكال، ابعاد الضوضاء، تتبع شكل ثنائي الابعاد

## 1. Introduction

We can define the object hole as "a black pixel surrounded by a connected boundary region", this definition is mentioned for each pixel; another definition of the object holes is "background region surrounded by a connected border of foreground pixels" [1], the other definition is "an object has a hole whenever it contains a closed path which cannot be transformed into a single point by a sequence of elementary deformations inside the object" [2], one more final hole definition is "break in a surface mesh" [3].

The term "hole" has not good interpretation in case of 3D objects since in these objects the view is changed and we need to define a new term(s), 3D objects contain cavities and concavities, the cavity is the existence of a hollow in the object, or a bounded connected component of the background, while the concavity is considered to be a contour with concave shape [2]

The image transformation and segmentation techniques cause some noise to be added inside the object that they working on, this detected object in his current state can cause some problems since some parts of that objects are missing due to the noise, for the correct feature extraction phase or any future manipulation with that object; the noise should be removed [2], and we are talking about the noise within the object which is more difficult to handle than outside.

Furthermore, for object fitting and building the best model that describes the object distribution, we need to scan or formulate the model from a completed object, any missing in these data will cause a false or incorrect fitting that will affect the simulation process, however, for a correct parameter estimations and fitting process, a correct filling algorithm should be applied [4].

This step can be postponed to be a post-processing step after applying all the pre-processing steps on object and hence, the new emerged holes from pre-processing step will be processed as well, and also this will reduce the computational time since the operation will be applied close to hole area rather than spending time with un-processed object [4].

In this paper, a 2D object filling model is proposed, it’s based on locating the black pixels (background) surrounded by a connected white (foreground) region (pixels), we have applied a binary image representation for object filling operation.

## 2. General System Description

The proposed filling algorithm composes of two main stages; these stages are show in Figure (1).

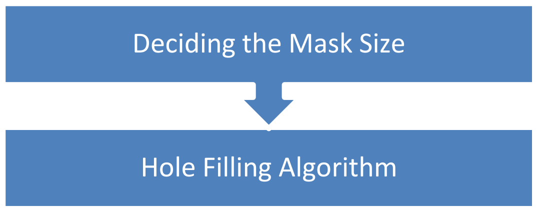


**Figure (1): Two Stages of our Suggested System.**

The second stage with more detailed is shown in Figure (2), we can note that this algorithm can handle one object in one time, we can repeat this algorithm for handling more than one object.

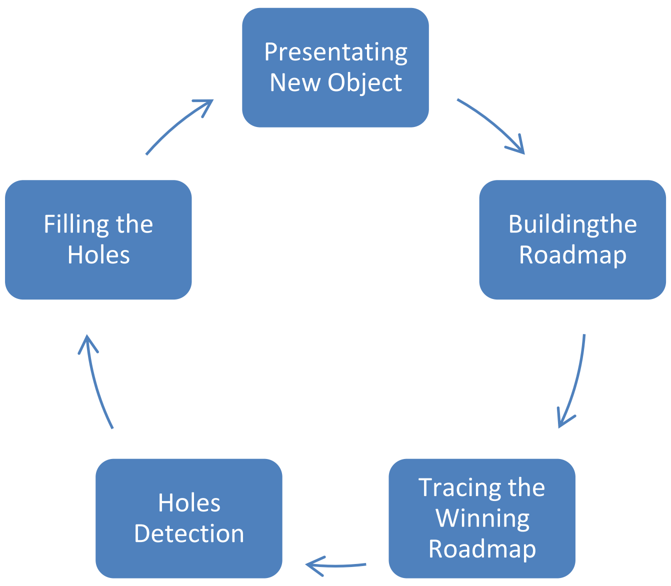


**Figure (2): Hole Filling Algorithm.**

### 2.1 Determining Mask Size

The mask is very important to locate the pixel hole, we have applied a diamond mask shape with certain diameter which is the distance between two opposite vertex of the diamond, Figure (3) shows some masks with different diameters.

$$\begin{matrix} 0&0&1&0&0\\ 0&1&1&1&0\\ 1&1&0&1&1\\ 0&1&1&1&0\\ 0&0&1&0&0 \end{matrix}$$

a

$$\begin{matrix} 0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&1&1&1&1&1&0&0&0\\ 0&0&1&1&1&1&1&1&1&0&0\\ 0&1&1&1&1&1&1&1&1&1&0\\ 1&1&1&1&1&0&1&1&1&1&1\\ 0&1&1&1&1&1&1&1&1&1&0\\ 0&0&1&1&1&1&1&1&1&0&0\\ 0&0&0&1&1&1&1&1&0&0&0\\ 0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0 \end{matrix}$$

b

a) 5x 5 mask sizes. b) 11x11 mask size.
Figure (3): Different Mask Sizes Employed by our System.

As shown in figure (3), the centre pixel of the mask is zero which means a black pixel which means a hole pixel, whereas the surrounded pixels are foreground pixels.
However, the mask size is important in this process since it is the tool for finding the pixel hole, we can estimate the mask size so that it covers the maximum hole region, the mask selection is controlled by the following assumption:
Let the Hij represents the hole j of object i, which means the input image may has a collection of objects each with many holes, let d represent the diameter of the mask, then

$$d > \text{length}\,(H_{ij}) \qquad (1)$$

where length(w) is the maximum distance between each opposite hole region, in other words, the mask should cover the hole completely.
After deciding the mask size, the processing speed will be enhanced as the mask size goes down, we can make the mask size big enough to cover whole the input object but this will slow the algorithm down, so, a careful estimation of the mask size will enhance and speed up the processing time.

## 3. System Implementation

The following sections describe the detailed implementation of the proposed system, according to the cycle of Figure (2).

### 3.1 Presenting New Input

In this stage a new input is presented to the system, the input take the form of an image with single object or multiple objects, however, the system performs pixel wide operations and the number of objects in the input image has no impact, this input is converted to binary image for discriminating between the foreground pixels and background pixels.

### 3.2 Building the Roadmap

The system iterates for each pixels in the input object and seeking for a hole pixel which is the background pixel, while foreground pixels are ignored.
In order to understand the roadmap, we will consider the following objects in Figure (4).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

a

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

b

a) closed object, b) opened object

Figure (4): two objects will be considered as an example.

Both of the latter objects are convolved with the mask of size 11x11 that mentioned in Figure (3b), the resulting will be in Figure (5) which is called the masked window.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

a

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

b

a) convolution of figure(4a) and figure (3b)
b) convolution of figure(4b) and figure (3b)

Figure (5): convolution operation applied between object window and 11x11 mask.

We can notice that the resulting convolved window is 11x11 since the mask size is. After that convolution operation, we have to look to see if there is any way out starting from the middle of the masked window to the border, there are many possibilities and combinations of such path, in either meaning, we try to find a border that surrounding the centre background pixel, the pixel to be considered as a hole pixel if such border exists. Figure (6) shows different samples of sub-masked windows which is very difficult to trace.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | **1** | **1** | **1** | **1** | **1** | 0 |
| 0 | 0 | 0 | **1** | x | 0 | 0 | **1** | 0 |
| 0 | **1** | **1** | **1** | 1 | 1 | 0 | **1** | 0 |
| 0 | **1** | 0 | 1 | 0 | 0 | 0 | **1** | 1 |
| 0 | **1** | 0 | 0 | 0 | 1 | 0 | 0 | **1** |
| 0 | 0 | **1** | **1** | **1** | **1** | **1** | **1** | **1** |

a

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | **0** | **0** | **0** | **0** | **0** | **0** | 1 |
| 0 | 1 | **0** | 1 | 1 | 1 | 1 | **0** | 1 |
| 0 | 1 | **0** | 1 | 1 | 0 | 1 | **0** | 1 |
| 0 | 1 | **0** | 1 | **x** | 0 | 1 | **0** | **0** |
| 0 | 1 | **0** | **0** | **0** | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

b

a) hole pixel, b) non hole pixel

Figure 6: two different sub-masked window.

As shown in figure(6), two sub-masked windows are presented, Figure (6a) shows that this background pixel is a hole pixel whereas Figure (6b) shows not.

Now, to find such connected border around the centre window background pixel, we expanded the above masked window into an equal length of tracing paths, each path represents a full diamond, so, we will have r tracing paths where r represents the vertical radius of the diamond (vertical and horizontal are the same).

There is a problem facing this expansion which is; the number of pixels in each tracing path is different since it produced from different size diamond, equation (2) shows the size for each diamond in the mask, considering the diamond diameter is d and radius is r.

$$NOP=DN * 4 \qquad (2)$$

Where NOP =number of pixels, and DN=diamond number, the diamonds are start numbering from the centre of the mask, the following example shows the 5x5 mask with their diamonds and the tracing paths.

Consider we have 9x9 mask which is shown in figure (7) :

$$\begin{matrix}
0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 5 & L & 5 & 0 & 0 & 0 \\
0 & 0 & 5 & K & g & M & 5 & 0 & 0 \\
0 & 5 & H & f & 4 & h & N & 5 & 0 \\
5 & G & e & 3 & x & 1 & a & A & 5 \\
0 & 5 & F & d & 2 & b & B & 5 & 0 \\
0 & 0 & 5 & E & c & C & 5 & 0 & 0 \\
0 & 0 & 0 & 5 & D & 5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0
\end{matrix}$$

**Figure 7: 9x9 coded mask.**

We have coded the diamond with letters so the expansion will be more digested. In order to make the length of the tracing path for each one equal, we have to repeat some points, these repeated points is the vertices of the diamond. This repetition is controlled by the following equation:

$$RP=r_{global} - r_{local} \qquad (3)$$

Where RP is the number of repeated points, $r_{global}$ is the radius of the bigger diamond (mask), and $r_{local}$ is the radius of that diamond, in this equation we have controlled the number of repeated pixels in order to formulate a unique length tracing paths, Table (1) shows the tracing paths of figure (7).

**Table (1): Tracing Paths that corresponds to figure (7).**

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| diamond r=1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 1 | 1 | 1 |
| diamond r=2 | a | b | c | c | c | d | e | e | e | f | g | g | g | h | a | a |
| diamond r=3 | a | b | c | d | d | e | f | g | g | h | k | l | l | m | n | a |
| diamond r=4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

What we have done is, starting with vertex **1** of the inner diamond and places it in the path, then we are repeating the next point as equation (3), the shaded cells represent the repeated codes. As seen in latter figure, all the tracing paths are equally length which enable us to proceed to the next level for hole pixel discovering.

### 3.3 Tracing the Winning Roadmap

In this level we are trying to trace the road paths (maps) that are produced for each pixel, the key point is to find a full paths starting from the beginning and reaching to the end of path. Consider a path P={$w_0$, $w_1$, ...., $w_t$}, where t is the roadmap length, and consider $RM_{bk}$ is the roadmap for a given background pixel $bk_{x,\,y}$, then this pixel is said to be a hole pixel $h_{x,\,y}$ the following equation (5) holds:

$$P = \text{Tracing}\ (r,\ t,\ RM_{bk}) \qquad (4)$$

$$pixel\ status = \begin{cases} hole\ pixel & if\ length(P) = t \\ \\ background\ pixel & otherwise \end{cases} \qquad (5)$$

Where Tracing(r, t, P, RM) is our tracing algorithm.

For better understanding, we can put this in two cases, each road path in Table(1) represents a full diamond, furthermore, if this path are connected without any black pixel; this means the diamond is closing the centre pixel which, in turns, leads to hole pixel declaration, since it is surrounded by a connected region, this is the first case, the other and the most encountered case, like Figure (5a), there is no connected diamond but these is an connected region formed by multiple diamonds, our aim is to verify the first case which is faster, it not, we continue with second case which finds the hole pixels if such exist.

Table (2) shows the paths expansion corresponding to Figure (4a) which has a connected region around (3, 3).

**Table (2): Tracing the Paths for Figure (4a).**

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 | 0 | |
| start point | **1** | 0 | **1** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | **1** | **0** | **1** | **1** | end point |
| | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

As we have seen above, this point is considered to be a hole pixel since we traced a connected path from start point to end point, on the contrary, Figure(4b) should declare no hole point since there is a crack in the structure, Table (3) shows the trace.

**Table (3): Tracing the Paths for Figure (4b).**

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | **1** | **1** | **1** | **1** | **1** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | |
| start point | **1** | 0 | **1** | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | end point |
| | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | |
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

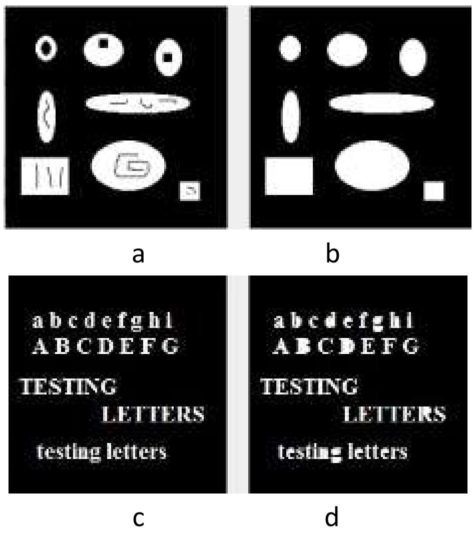
The trace stops since there is no more connection, and furthermore, you can notice that there are a gap in the middle of tracing paths, this gap of zeroes is caused since the structure is opened.

Tracing Algorithm

```
Input: r represents mask radius, t represents total number of points in bigger diamond
        (roadmap    length)
       StatusMap is a matrix with size r x t contains the on, off pixels of the road paths
Output: background pixel or hole pixel.
Method:
Step 1: [initialization]
        initialize k[i] for all i=0, .., r-1  with true values, k represents the tracing status vector
        at each step.
        StepCounter=0;
Step 2: [repeat]
        Repeat to step (7) while StepCounter<t
Step 3: [repeat for the current step]
        For each i where i=0, 1,…, r-1  do up to step (7) the following:
        CurrentStatus=StatusMap(i, StepCounter)
Step 4: [update the upper pixels in the current step]
        If CurrentStatus is on and k[i] is on then update all k[j] for all j<i with the following:
        k[j]=k[j] ANDing StatusMap ( j, StepCounter )
Step 5: [update the lower pixel in the current step]
        Update k[j] where j=i+1 with the following:
        k[j]=k[j] ANDing StatusMap ( j, StepCounter )
Step 6: [update the current status of tracing status vector]
        k[i]=CurrentStatus ANDing k[i]
Step 7: [gap checking checking]
        If all k[j] for j=0, 1,.., r-1 are off, then declare that this is an background pixel, stop.
        Else go back to step (3).
Step 8: Declare that this is a hole pixel, stop.
```

## 4. The Experimental Results

We have tested our suggested algorithm with different test images, some of them with objects and other with text images and the results was as expected, the following Figure (8) shows some of these tested images.



(a and c) are input images, (b and d) are output images
Figure (8): Some Testing Images.

## 5. Conclusion

Hole filling algorithms are considered to be important to close the fill the noise inside the object, especially before the feature extraction phase for an object since these noises may give a wrong feature translation.

In this study we have introduced an algorithm for hole filling that looks and seeks for the a background pixel lie inside a connected region, and hence, this background pixel is announced as hole pixel.

This algorithm uses tables based road-tracking to finds out the connected region that surrounds the hole, this connected region should belong to foreground which refers to crack or hole inside the foreground, after that this hole is closed.

## 6. References

[1] Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, Pearson Prentice Hall, First Impression, 2009.

[2] Zouina Aktouf, Gilles Bertrand, and Laurent Perroton, “A three-dimensional holes closing algorithm”, ELSEVIER Pattern Recognition Letters, vol. 23 :523–531, 2002.

[3] Eric Firestone, “An Exploration Of Hole Filling Algorithms”, M.Sc. Thesis, Faculty Of California Polytechnic State University, San Luis Obispo, 2008.

[4] Elaine Cohen , Lavanya Sita Tekumalla , and Elaine Cohen,”A Hole-Filling Algorithm for Triangular Meshes”, School of Computing, University of Utah, USA, 2004.

[5] Matteo Dellepiane, Andrea Venturi, and Roberto Scopigno,” Image guided reconstruction of un-sampled data: a coherent filling for uncomplete Cultural Heritage models”, International Journal of Computer Vision, vol. 94(1), USA, 2011.

[6] Kwan-Jung Oh, Sehoon Yea,and Yo-Sung Ho,” Hole-Filling Method Using Depth Based In-Painting For View Synthesis in Free Viewpoint Television (FTV) and 3D Video”, IEEE 27th conference on Picture Coding Symposium, Chicago, IL, 2009, pp. 1-4.

[7] Alan Brunton, Stefanie Wuhrer, Chang Shu, Prosenjit Bose, and Erik Demaine, “Filling Holes in Triangular Meshes Using Digital Images by Curve Unfolding”, International Journal of Shape Modeling, vol. 16(1-2), NRC Institute for Information Technology, 2011.
[8] Mingqiang Wei, Jianhuang Wu, and Mingyong Pang, “An Integrated Approach To Filling Holes In Meshes”, IEEE International Conference on Artificial Intelligence and Computational Intelligence, 2010, pp. 306 – 310.
[9] Xiao J. Wu, Michael Y. Wang, and B. Han, “An Automatic Hole-Filling Algorithm for Polygon Meshes”, Computer-Aided Design and Applications, Vol. 5, No. 6. (2008), pp. 889-899.
[10] Chun-Yen Chen, Kuo-Young Cheng, and H. Y. Mark Liao,” ASharpness Dependent Approach to 3D Polygon Mesh Hole Filling”, Proceedings of EuroGraphics, 2005, pp. 13-16.
[11] Amitesh Kumar, Alan Shih, Yasushi Ito, Douglas Ross, and Bharat Soni,”A Hole-filling Algorithm Using Non-uniform Rational B-splines”, Department of Mechanical Engineering, University of Alabama at Birmingham, Birmingham, AL, U.S.A., 2008.