# MODIFIED SEQUENTIAL ALGORITHM USING EUCLIDEAN DISTANCE FUNCTION FOR SEED FILLING

[1]SAADIA SADDIQUE, [2]MALIK SIKANDAR HAYAT KHIYAL, [3]AIHAB KHAN, [4]MEMOONA KHANUM, .

[1]Graduate, Department of Computer Science, Fatima Jinnah Women University, Rawalpindi, Pakistan

[2]Chairperson, Department of Computer Science and Software Engineering, Fatima Jinnah Women University, Rawalpindi, Pakistan

[3]Assistant Professor, Department of Software Engineering, Fatima Jinnah Women University, Rawalpindi, Pakistan

[4]Lecturer, Department of Computer Science, Fatima Jinnah Women University, Rawalpindi, Pakistan

**ABSTRACT**

Euclidean Distance Transform has been widely studied in computational geometry, image processing, computer graphics and pattern recognition. Euclidean distance has been computed through different algorithms like parallel, linear time algorithms etc. On the basis of efficiency, accuracy and numerical computations, existing and proposed techniques has been compared. This study proposed a new technique of finding Euclidian distance using sequential algorithm. An experimental evaluation has shown that proposed technique has reduced the drawbacks of existing techniques. And the use of sequential algorithm scans has reduced the computational cost.

***Keywords:*** *Euclidean Distance Transformation, Image Processing, Image Transformation, Sequential Algorithm.*

## 1. INTRODUCTION

Distances play a central role in comparison of binary images, especially for images resulting from local features such as edge or corner detection and used to obtain the medial axes of digital shapes. So many researches focus on finding distance through different algorithms. In computer vision and in many morphological image application distances plays a very important role. Euclidean distance is most common among all types of distance. Since Euclidean distance transform has many applications so many algorithms are proposed to find that distance [6]. One method for distance calculation is to use the multiple morphological erosion operators with a suitable structuring element until all foreground pixels of the image have been eroded away. The shape of structuring element depends upon the type of the distance metric normally square shape element used for chess board; cross shape for city block and disc shaped element for Euclidean distance is used. But it is the extremely inefficient way. Although these approaches shows the linear time complexity of these according to the number of pixels of the image but it does not compute the exact Euclidean distance, which is required for some applications.

In parallel computers these algorithms are difficult to parallelize during the computation. Euclidean distance calculation is although a simple but time-consuming operation. It takes an excessive amount of time. The drawbacks of those techniques for calculating distance are very expensive in terms of computing time and resources.This paper focuses on finding exact Euclidean distance by using Sequential algorithm in order to reduce computational cost. Sequential Algorithm is modified for Euclidean distance calculation. The division of this paper is as follows, In section 2 related work is given. In section 3 Preliminaries have been explained. In section 4 framework of the proposed system has been discussed. Section 5 gives proposed technique. In section 6 results are presented and in section 7 conclusion and future work of this research work is presented.

## 2. Related Work

Moore [5] discusses about distance, its ifferent types and algorithms with their advantages and disadvantages for calculation of different distance transform function. For example the brute force technique, although calculate the

exact Euclidean distance but its needs more resources and computation time is high. Another approach is incremental which uses the concept of neighborhood. Peuquet states region expansion technique using real numbers [7]. Cuisenaire [1] states difference between chamfering and vector distance transforms methods. The drawback of chamfer method or Sequential Weighted Distance Transform (SWDT) is that it gives no orientation information and is rotation sensitive. To overcome these limitations Danielsson[6] present vector method. The disadvantages of Danielsson [6] algorithms are that the distance is calculated only from the distances associated to the pixels in neighbors. So this is not sufficient to produce exact results in all cases and the processing and storage cost to be paid.
Noticing the errors in Danielsson method, a region growing algorithm is presented by Cuisenaire [1] which uses the concept of bucket sorting by using ordered contoured propagation by Voronoi diagrams. This technique leads to linear time complexity of Euclidean distance. Porikli and Kocakin [8] present the two fast algorithms to calculate the approximate distance transformation of 2 Dimensional binary images.
The Chen and Chuang [3] algorithm is same as of using dimension reduction or independent scanning. Hirata [2] states unified algorithm which calculates the Euclidean distance with time complexity $O(n^2)$. Vincent [4] proposed a new algorithm named chain propagations for calculating Euclidean distance. This algorithm is well suited for computing Euclidean distance in conventional computers as compare to all early proposed algorithms. It requires a random access to the pixels so not only suited to general purpose computers but also for some architecture it is extremely fast on parallel computers.

**3. Preliminaries**

Euclidean distance is most common in most of applications. There are various ways of applying the distance transform depending on which distance metric is being used and how the local distance information is propagated. e.g. hamming distance, Manhattan distance Chessboard, and Euclidean distance etc. But Euclidean is being the most common.

In mathematics, Euclidean distance can be defined as the straight line connecting two points distance and can be measured with a ruler. The Euclidean distance [6] between points $P = (p_1, p_2, ..., p_n)$ and

$Q = (q_1, q_2, ..., q_n)$ In Euclidean *n*-space, is defined as

$$\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+...+(p_n-q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}$$

**4. Framework**

The proposed system calculates the Euclidean distance by using sequential algorithm. Once the image is acquired then after the preprocessing step, the Euclidean distance is calculated from the image. The steps involve in the implementation phase are as shown in fig 1. The logical design of the system is shown in Fig 2.

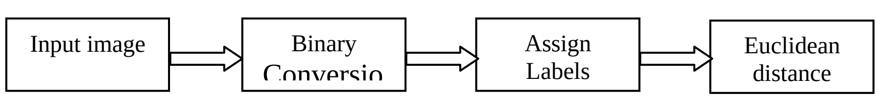


**Fig 1:** Process of Euclidean Distance Calculation

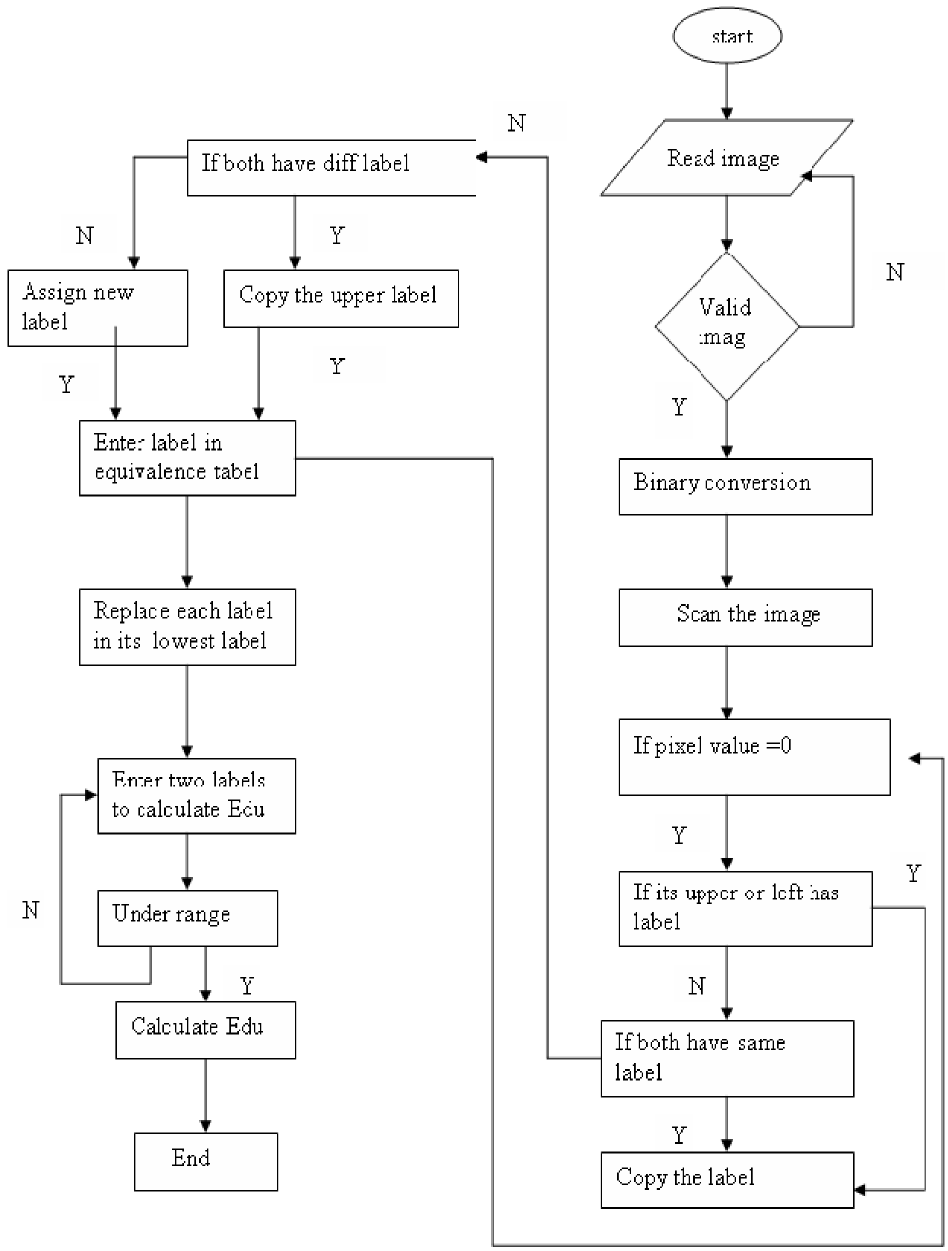


**Fig 2:** Flow Chart of the Proposed System

## 5. Proposed Technique

### 5.1: Model of Proposed Technique

The proposed research based on the calculation of exact Euclidean Distance by using Modified Sequential Algorithm. Sequential or recursive algorithms are proposed for the purpose of reducing the number of scanning required for the computation of an image transform.

### 5.2: Working of Proposed System

For implementing the proposed technique, the following steps and functions are involved.

**Step 1: Read an image**

The input image will be read out.

**Step 2: Binary Conversion**

The image will be converted into binary form by using *"Dither"* function.

**Step 3: Assigning Labels**

The next step is to assign labels to binary image. The assignment of labels is based upon following criteria's[1]:

**Case I: First row and first column:**

i. If first row and column of the image is scanned and the pixel value is zero then assign a new label to binary image.
ii. Else IF first row but not first column of the image is scanned and the pixel value is zero and also left neighbor pixel of new image is not equal to zero then assign the value of left neighbor pixel as a label otherwise assign a new label.
iii. Else IF first column but not first row of the image is scanned and the pixel value is zero and also upper neighbor pixel of new image is not equal to zero then assign the value of upper neighbor pixel as a label.

**Case II: Not a first row and first column:**

i. If pixel value is zero and one of its left or upper neighbors has a label then copy the label.
ii. Else IF both have the same labels then copy the label.
iii. Else IF both have the different labels then copy the label.

**Step 4: Euclidean Distance Calculation:** So after assigning labels the proposed system calculates the Euclidean distance by entering the labels under the range of input image. First it checks the coordinates of both labels and then Euclidean distance is calculated by putting the coordinates of input labels in the following formula. edu_dis=sqrt(((row1-row2).^2) + ((col1-col2). ^2)).

**5.3: Algorithm**

The pseudo code for all steps of the proposed system is as follow:

**//Read an Image**

I=imread(filename.fmt).
I=imread("tire.tif");

**//Binary Conversion**

im = dither(I);

**//Assigning Labels**

[row col]=size(im); //Save the image size in the form of rows and columns

im1=zeros(row,col); //Create a new image of zeros with the same size of the input image.

//im1 is the new image

k=1; //initializing labels

"For loop" is used for scanning the whole image i.e. left to right, top to bottom.

**//Case I:**

//If first row and column of the image is scanned and the pixel value is zero then assign a new label to im1

```
If row==1
    If col==1
If image(row,col)==0
Assign label k to im1
Increment the label k++.
```

//elseIf first row but not first column of the image is scanned and the pixel value is zero and also left neighbor pixel of new image is not equal to zero then assign the value of left neighbor pixel as a label otherwise assign a new label.

```
elseif row==1
    If column!=1
        If image(row,col)==0
          If(im1(row,col-1)!=0)
            Assign im1(row,col-1) as a label
           else
                Assign label k to im1
                 Increment the label k++.
```

//elseif first column but not first row of the image is scanned and the pixel value is zero and also upper neighbor pixel of new image is not equal to zero then assign the value of upper neighbor pixel as a label.

```
else If row!=1
    If col==1
      If image(row,col)==0
        If(im1(row-1,col)!=0)
           Assign im1(row-1,col) as a label
      else
          Assign label k to im1
          Increment the label k++
```

**Case II:**

```
If row!=1
    if col!=1
```

// (i) if pixel value is zero and one of its left or upper neighbors has a label then copy label

```
 If image(row,col)==0
If(im1(row-1,col)!=0&& im1(row,col-1)==0 )
    Assign im1(row-1,col) as a label
   else
    If(im1(row,col-1)!=0&&im1(row-1,col)==0)
     Assign im1(row,col-1)as a label
```

// (ii) elseif both have the same label then copy the label

```
If image(row,col)==0
 If(im1(row-1,col)==im1(row,col-1)&& im1(row-1,col)!=0 )
   Assign im1(row-1,col) as a label
```

/// (iii)elseif both has the different label then copy the upper label

```
If(im1(row-1,col)!=im1(row,col-1)&&im1(row -1,col)!=0&&im1(row,col-1)!=0 )
    Assign im1(row-1,col) as a label
```

otherwise assign a new label and increment the label

Assign a new label
Increment the label k++

//**Euclidean distance calculation**

//Now enters any two labels under the range and save their coordinates simultaneously.

```
a=input('Enter first Label: ');
b=input('Enter second Label: ');
[rows cols]=find(im1==a);
[rows1 cols1]=find(im1==b);

edu_dis=sqrt(((row1-row2).^2) + ((col1-col2).^2)).
```

**Output:**

The results are displayed by using the "imshow" function.

imshow(I);
display the original imag.
imshow(im);
display the binary image.
imshow(im1);
display the labeled image.

### 6. Results

The proposed algorithm has been applied on original image. Fig 3 shows the original image.

**Fig 3:** Original Image

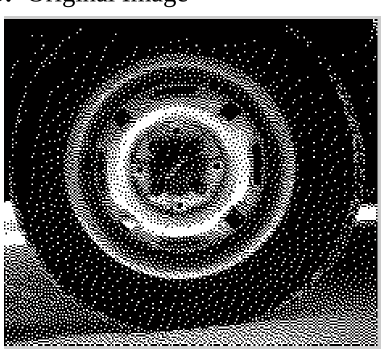

Fig. 4: Image after Binary Conversion

To reduce the deficiencies of previous algorithms, we applied the binary conversion on the original image shown in Fig 3. Fig. 4 shows the image after binary conversion.

Then the step of assigning labels has been performed to reduce the complexity of algorithm from O ($n^3$) to O ($n^2$). Fig. 5 shows image after assigning labels.

Fig. 5: Image after Assigning Labels

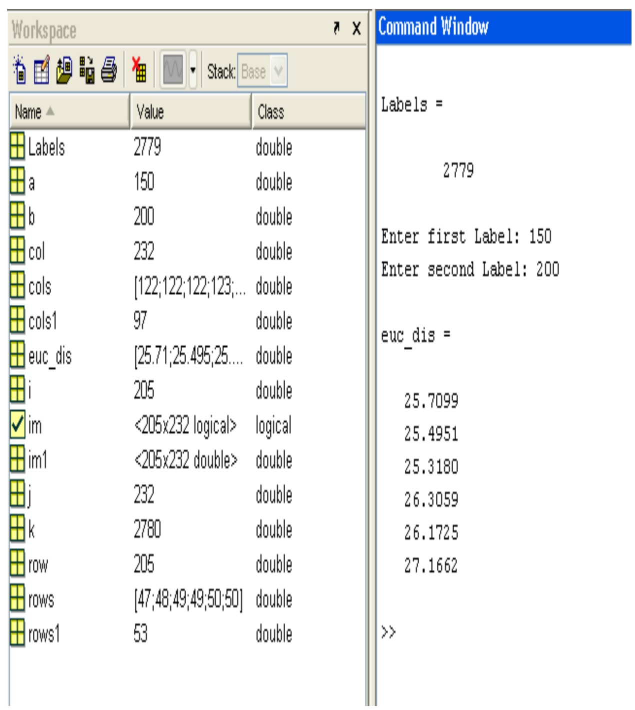


Fig 6: Euclidean Distance Calculation

Fig. 6 shows calculation of Euclidean distance using two selected labels.

### 7. Conclusions and Future Work

The proposed technique is based on the calculation of Euclidean distance using Sequential algorithm. To reduce the deficiencies of previous algorithms, this study focuses on finding Euclidean distance using sequential algorithm. This system will deal with only binary images. At a time only one image will be examined, as sequential algorithm scan only two images so the computational cost has been reduced.

**References**


[1] O. Cuisenaire. "Region Growing Euclidean Distance Transforms". In: *Proceedings of 9th International Conference on Image Analysis and Processing (ICIAP'97)*, 1997, Vol.1, pages 263-270.

[2] T. Hirata. "A unified linear-time algorithm for computing distance maps". *Information Processing Letters*, 1996, vol. 58, issue 3, pages 129-133,. ISSN: 0020-0190

[3] Ling Chen, Henry Y. H. Chuang. "An Efficient Algorithm for Complete Euclidean Distance Transform on Mesh-Connected SIMD". Parallel Computing 1995, 21(5): 841-852

[4] Luc Vincent. "New Trends in Morphological Algorithms", Harvard University, Division of Applied Sciences Pierce Hall, Cambridge MA 02138, USA Proc. SPIE/SPSE Vol. 1451, Nonlinear Image Processing II, pp. 158-170, San Jose CA, 1991.

[5] A. Moore. "The Case for Approximate Distance Transforms", University of Otago, Dunedin, Presented at SIRC 2002 – The 14th Annual Colloquium of the Spatial Information Research Centre, University of Otago, Dunedin, New Zealand.

[6] P.E.Danielsson. "Euclidean distance mapping". Journal *Computer Graphics and Image Processing*, 1980, 14:227-248,

[7] D.J. Peuquet. "An Algorithm for Calculating Minimum Euclidean Distance between two Geographic Features". *Computers and Geosciences*, 1992, 18, 8, 989-1001. ISSN:0098-3004

[8] F. Porikli and T. Kocak. "Fast Distance Transform Computation using Dual Scan Line Propagation". Mitsubishi Electric Research Laboratories, Cambridge, USA, Proc. SPIE, Vol. 6496, 649608 (2007); DOI:10.1117/12.704760.