Reducing the Error in Patterns Fabricated on the Silicon Wafers

A PROJECT REPORT

Submitted in partial fulfillment of the
requirements for the award of the degrees
of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Amarnath Reddy
140001019
Guided by:
Dr Somanath Dey,

Assistant Professor,

Computer Science and Engineering,

IIT Indore

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2017

CANDIDATE'S DECLARATION

I hereby declare that the project entitled "**Reducing the Error in Patterns Fabricated on the Silicon Wafers**" submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Computer Science and Engineering' completed under the supervision of **Dr. Somnath Dey, Assistant Professor, Computer Science and Engineering, IIT Indore and Bhardwaj Durvasula, Mentor Graphics Pvt. Ltd.** is an authentic work. Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Amarnath Reddy

CERTIFICATE by BTP Guide

It is certified that the above statement made by the student is correct to the best of my knowledge.

Dr Somanath Dey,

Assistant Professor,

Computer Science and Engineering,

IIT Indore

Preface

This report on "**REDUCING THE ERROR IN THE PATTERNS FABRICATED ON THE SILICON WAFERS** " is prepared under the guidance of Dr. Somnath Dey, Assistant Professor, Computer Science and Engineering, IIT Indore and Bhardwaj Durvasula, Mentor Graphics Pvt. Ltd.

Through this report I have tried to give a detailed description of our approach to reduce the errors in the contours after proper simulation. I have also tried to explore the possibility of incorporating evolutionary principles to reach a global optimum solution of the error reduction, which otherwise is very difficult.

I have tried to the best of my abilities and knowledge to explain the content in a lucid manner. I have also added figures to make it more illustrative and enable the readers to understand the solution easily.

Amarnath Reddy

B.Tech. IV Year

Computer Science and Engineering

IIT Indore

Acknowledgements

I wish to thank Dr Somnath Dey and Bhardwaj Durvasula for their kind support and valuable guidance throughout the duration of the project, giving me an opportunity to work at my own pace along my own lines, while providing me with very useful directions whenever necessary.

It is their help and support, due to which I became able to complete the design and technical report.

Without their support this report would not have been possible.

Amarnath Reddy

B.Tech. IV Year

Computer Science and Engineering

IIT Indore

ABSTRACT

During the mask data preparation process, the mask pattern is first fractured into basic rectangles and then fabricated by the variable shaped beam mask writing machine. To fracture mask pattern into rectangles, the edges in the pattern have to be manhattanized. The rectangle count included in the fractured pattern is preferable to be suppressed to reduce the mask fabricating time and cost. That is why kernel size of the beam used is not kept too small. As the kernel size increases, the error increases between the pattern we get on the silicon wafer at the end and the pattern we actually wanted. So instead of decreasing the kernel size, we can change the manhattanized edge patterns in a way such that the error decreases.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background of the Problem

Fig.1.1 shows the series of nanostructuring during the ebeam lithography(EBL) procedure. The exposure of a sensitive resist with an electron beam is the core of the EBL process. The resist is often a polymer dissolved in a liquid solvent, which is coated onto a surface and baked to form an even thin layer. During electron beam exposure, the solubility of the resist is altered, causing a dissolution variation with areas that are non-exposed. Following this, the pattern is developed using a liquid developer. Lastly, the combination with processes, like dry or wet etching, lift-off of evaporated material, and electro- deposition completes the fabrication of structure.

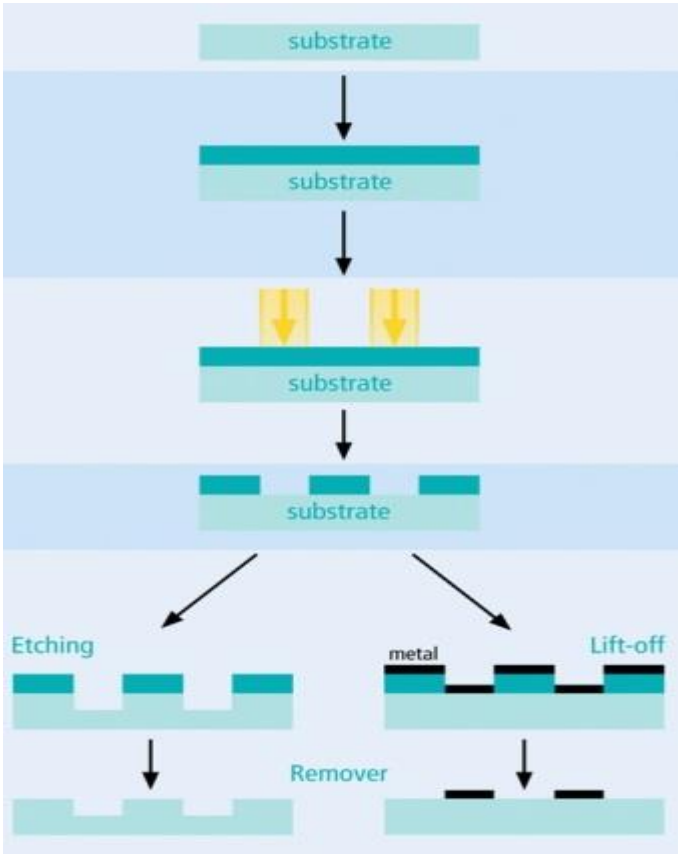

Figure 1.1 Wafer fabrication process. [4]

The proximity effect in electron beam lithography(EBL) is the phenomenon that the exposure dose distribution, and hence the developed pattern, is wider than the scanned pattern, due to the interactions of the primary beam electrons with the resist and substrate. These cause the resist outside the scanned pattern to receive a non-zero dose. Fig 1.2 describes the proximity effect.

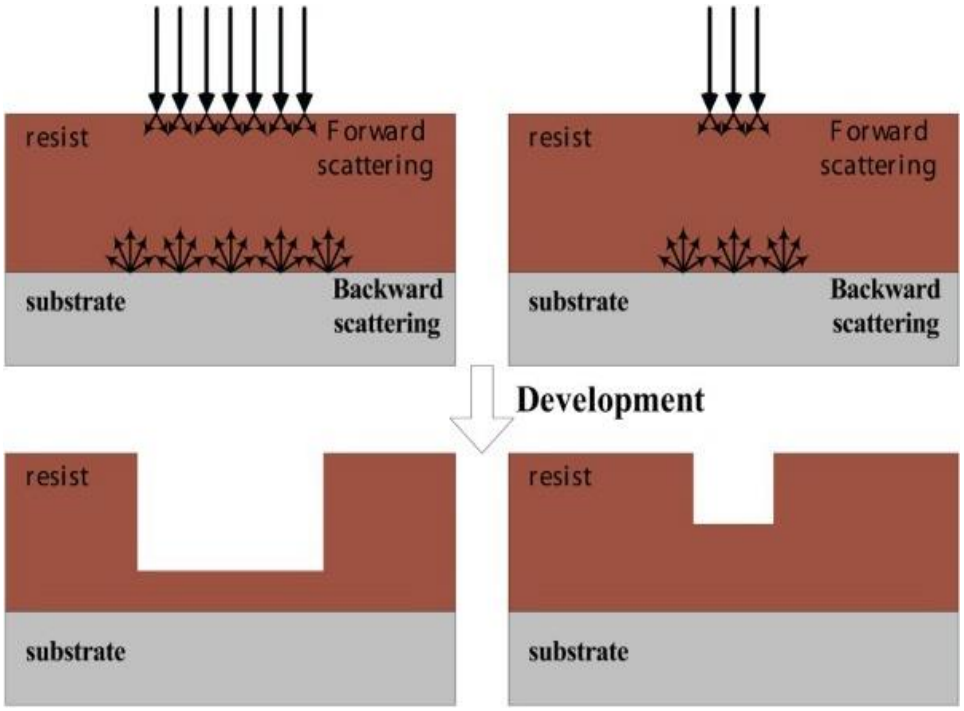

Figure 1.2 Proximity effect. [5]

Important contributions to weak-resist polymer chain scission (for positive resists) or crosslinking (for negative resists) come from electron forward scattering and backscattering. The forward scattering process is due to electron-electron interactions which deflect the primary electrons by a typically small angle, thus statistically broadening the beam in the resist (and further in the substrate). The majority of the electrons do not stop in the resist but penetrate the substrate. These electrons can still contribute to resist exposure by scattering back into the resist and causing subsequent inelastic or exposing processes. This backscattering process originates e.g. from a collision with a heavy particle (i.e. substrate nucleus) and leads to wide-angle scattering of the light electron from a range of depths (micrometres) in the substrate. The Rutherford backscattering probability increases quickly with substrate nuclear charge. That is why, we don't get the patterns which we actually wanted to get on the silicon wafer.

1.2 Motivations

Chips are made of large number of circuits. As there are errors between the patterns on the wafers and the circuits we design initially, there has to be some gap between the circuits. Otherwise circuits are connected badly and we will end up in getting wrong results. So if we reduce the error, we can reduce the gap between the circuits. As a result, we can accommodate more number of circuits on a chip. More number of functionalities can be added with which we can increase the processing speed of a chip. At present, there are 14nm node technology chips available in the market. So if we can reduce the error we can desire for smaller number node technology.

1.3 Objectives

Reducing the error between the target contour and the contour we get finally on the silicon wafer. Here contour refers to the circuit patterns.

Chapter 2

Literature Review

In this chapter, we will go through the process of generating data masks and how the patterns are fractured into rectangles, reasons for generating rectangles and then the process of projecting the data on the masks on to the silicon wafers. In section 2.1, we will discuss various mask generating techniques and their advantages over one another. In section 2.2, we will discuss the ebeam process.

2.1 Mask Generation

There are two readily available techniques for generating masks: Optical and Electron-beam (named for the form of energy used to expose the plates). Both of these methods lead to a set of master plates, which may be used to pattern the wafers directly. More commonly, a secondary set of plates called working plates, is used in the actual photolithography process. Working plates are printed directly from the masters, thus allowing one set of masters to be used to produce many wafers.

Of optical and E-beam mask generation, optical mask generation is the older of the two processes, offering low cost and wide availability. The pattern generation process is slow, however, and its speed is directly related to the complexity of the design. Because of the difficulty in controlling alignment during the step and repeat process and the danger of reticle defects, it is costly to include more than two different chip types on the same set of working plates. Electron-beam mask generation is free of these shortcomings, but is not yet widely available. The flexibility and fast turnaround afforded by E-beam closely matches the requirements of many research institutions.

2.2 Ebeam Process

As in the optical process, electron-beam mask generation equipment can be used to create patterns that are stepped and repeated on a master plate. More commonly, however, an entire master plate is written in one step. E-beam masks offer several advantages to researchers interested in fast turnaround: one-step mask generation (if the masters are used to directly expose the wafers), speed, flexibility, and reduced defects from

certain causes. For instance, a defect on a pattern means that each and every chip will have the same defect, in addition, the step and repeat process is a potential source of defects (for example, alignment problems, defects from dust specks). Both of these problem areas are eliminated with e-beam masters.

Unlike optical pattern generation equipment, electron-beam exposure systems are raster oriented. The mask can be visualized as a piece of graph paper, where the squares are the same size as the e-beam diameter (typically 0.25A or 0.5p). All geometric data is ultimately converted into a bitmap (a rectangular array of l's and 0's), which is placed on top of the graph paper — the squares containing l's are exposed, those containing 0's, not. Conceptually, the exposures are made by sweeping the electron beam in a repeating "S" pattern from the lower left-hand corner of the mask, blanking and unblanking the beam according to the input stream of bits. To a first approximation, the beam visits each point on the mask regardless of whether the point is exposed, and so the writing time is independent of the design complexity. In practice, this is not entirely true. Some machines are programmed to skip large blank areas, and so take less time to write sparse designs.

For practical reasons, the writing sequence is not quite that straightforward. Assume that we wish to write and array of 8 identical chips. The chip is divided into horizontal strips of fixed height and the geometric shapes within each strip are fractured into rectangles and trapezoids. Software is available to convert conventional PG formats to this e-beam format, or the designer can generate the trapezoids and rectangles directly. The location of each strip of the chip, in this case there are four strips, along with other information is used to create a command sequence for writing the array.

The first step in writing is converting the trapezoids and rectangles for a given strip into a bitmap, this process, called corefill (because the bitmap is loaded into core) is relatively time consuming. For this reason, it is only done once for a given strip. The machine then writes every area on the mask that is covered by that strip, before it converts another. Mechanically, the mask is moved in the x direction, while the electron beam scans in the y direction along short scan lines.

Chapter 3

Proposed Methodology

3.1 Pre-processing

Input is given in the form of edge collection. Each edge in the collection is given a polygon number (which polygon it belongs to) and the direction of the edge. We need to assemble all those edges in the collection according to the polygon number assigned. So from here on, we will take each polygon and process the remaining steps.

3.2 Evaluation Points

Pick the first vertex as first evaluation point on the target polygon. Mark it as the previous evaluation point. Search for the next evaluation point. Keep measuring the distance from previous evaluation point to the present vertex we are checking. If the distance exceeds the given distance, we will take it as an evaluation point. Then mark it as the previous evaluation point. Search for all the evaluation points in the same manner. We will store all the evaluation points of the target polygon in a vector.

For example in Fig. 3.1, A,B,C,D,E,F are all the evaluation points collected out of all the vertices in the polygon. The distance between A and B, B and C, C and D, D and E, E and F, F and A is less than 100nm. (This can be taken on our choice depending on the output performance and time limitations.)

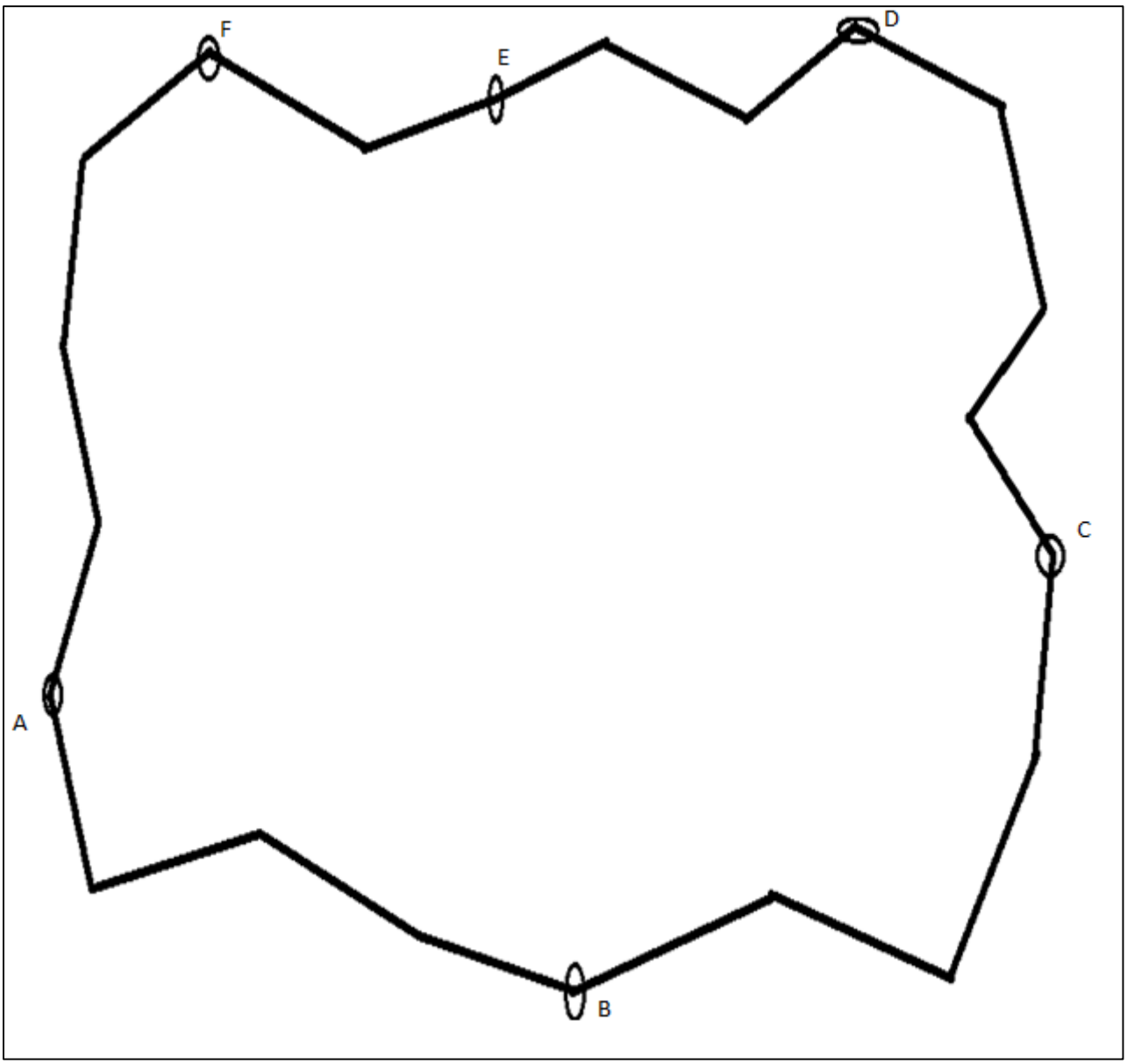

Figure 3.1 Placing evaluation points.

3.3 Assigning Edges

For each evaluation point, find the nearest point on the manhattan polygon from it. Take it as the first vertex. For the next evaluation point, again find the nearest point on the manhattan polygon from it. Take it as the second vertex. Now, for all the edges in between the first vertex and the second vertex on the manhattan polygon, assign first half of the edges to the previous evaluation point and next half to the present evaluation point. Now, consider present evaluation point as the previous one and take the next one as the present. We will do the same steps for all the evaluation points.

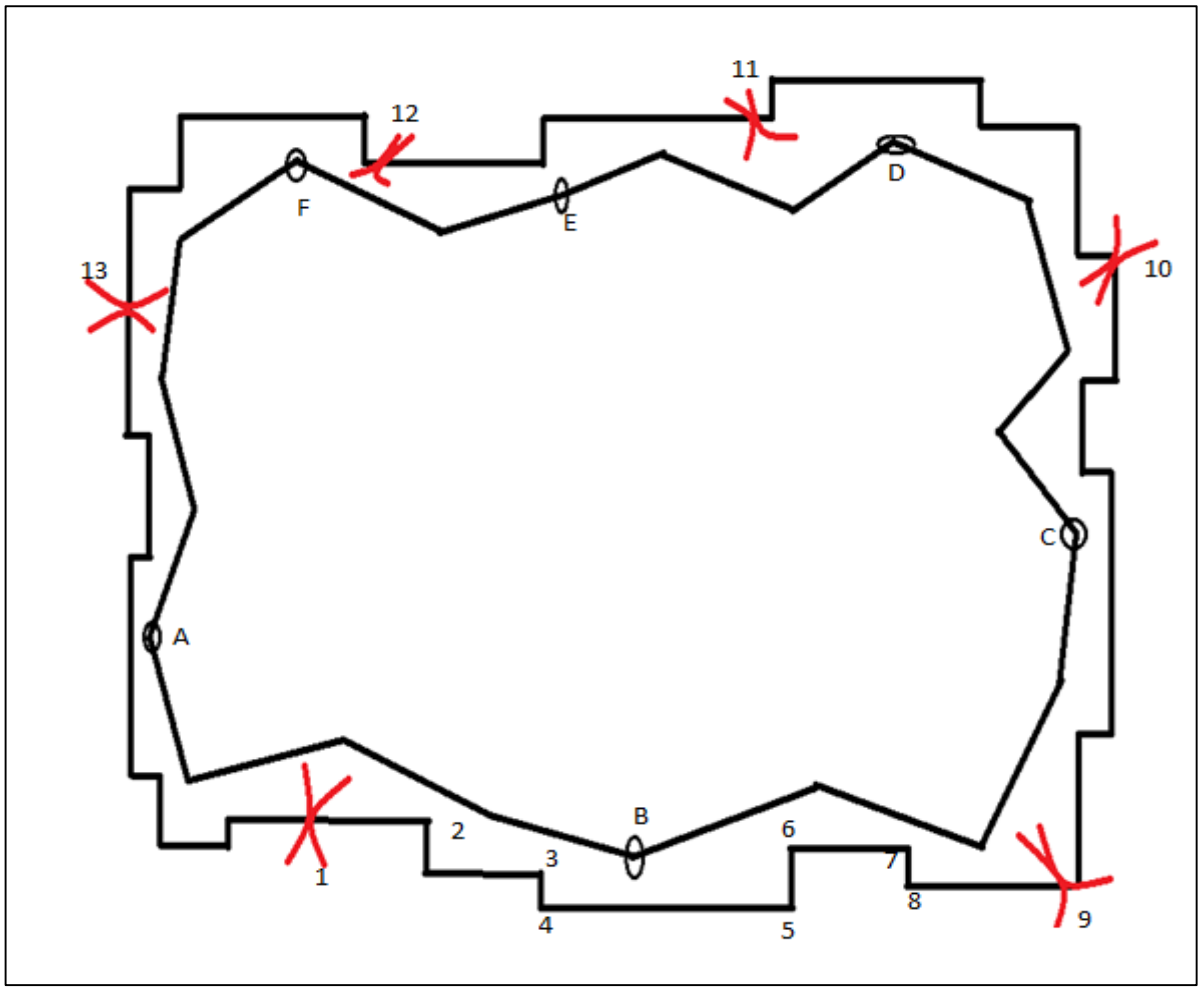

Figure 3.2 Assigning edges.

For example, in Fig. 3.2, edges 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9 are assigned to the B evaluation point. Because the nearest vertex to B is 3 and half number of edges are shared between the adjacent

evaluation points. Likewise, edges between 9 and 10 are assigned to C, edges between 10 and 11 are assigned to D, between 11 and 12 are assigned to E, edges between 12 and 13 are assigned to F, edges between 13 and 1 are assigned to A.

3.4 Finding Normal

First we will find the normal at the point. We consider two vertices beside this evaluation point. We connect the two points and calculate the perpendicular pointing outside the polygon.

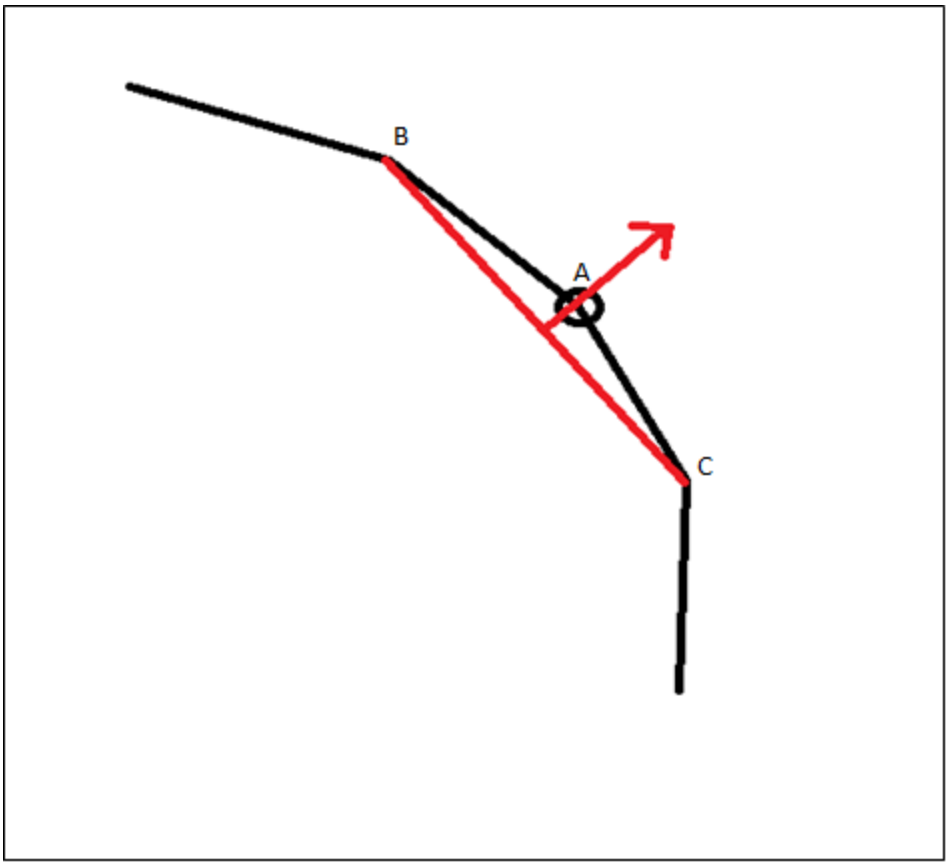

Figure 3.3 Normal at a point.

For example in Fig. 3.3, B and C are the vertices beside A. So, form a line segment connecting B and C. Get the slope of the line segment. With this, we can know the slope of the perpendicular line to the formed line segment. This will be the normal for evaluation point A.

3.5 Checking Error

Check the difference between intensity at the evaluation point and the threshold intensity. If the calculated difference is negative change the normal direction sign. The negative value indicates that the point is outside the simulated contour. So we will change the direction. In Fig. 3.4, the target contour (green color) is inside. So, the normal direction will change at the evaluation point.

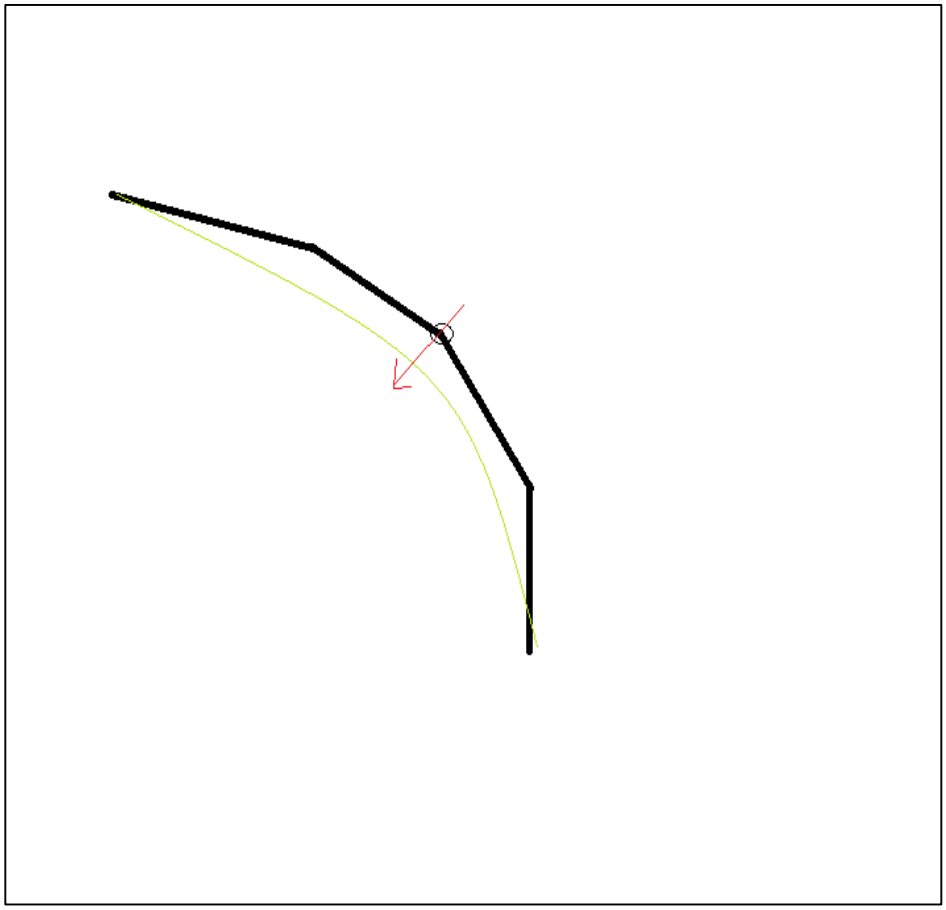

Figure 3.4 Direction change.

If the direction is positive, it indicates that we are moving in the proper direction. If the difference value is equal to the zero we don't need to do anything. Now take some step size. Calculate the intensity at

the step size away from the evaluation point. Keep increasing the step size until it the intensity matches the threshold intensity. If the intensity difference sign changes, it means that we have crossed the threshold in between. For example in Fig. 3.5, after moving step size s in normal direction, we will reach point A1, we calculate intensity difference at A1. Likewise, we calculate intensity difference at A2, A3 and so on. At A4, intensity difference is negative whereas it was positive at A3. It is clear that the target contour lies between A3 and A4. So, we will search for the exact point in between A3 and A4 using binary search technique.

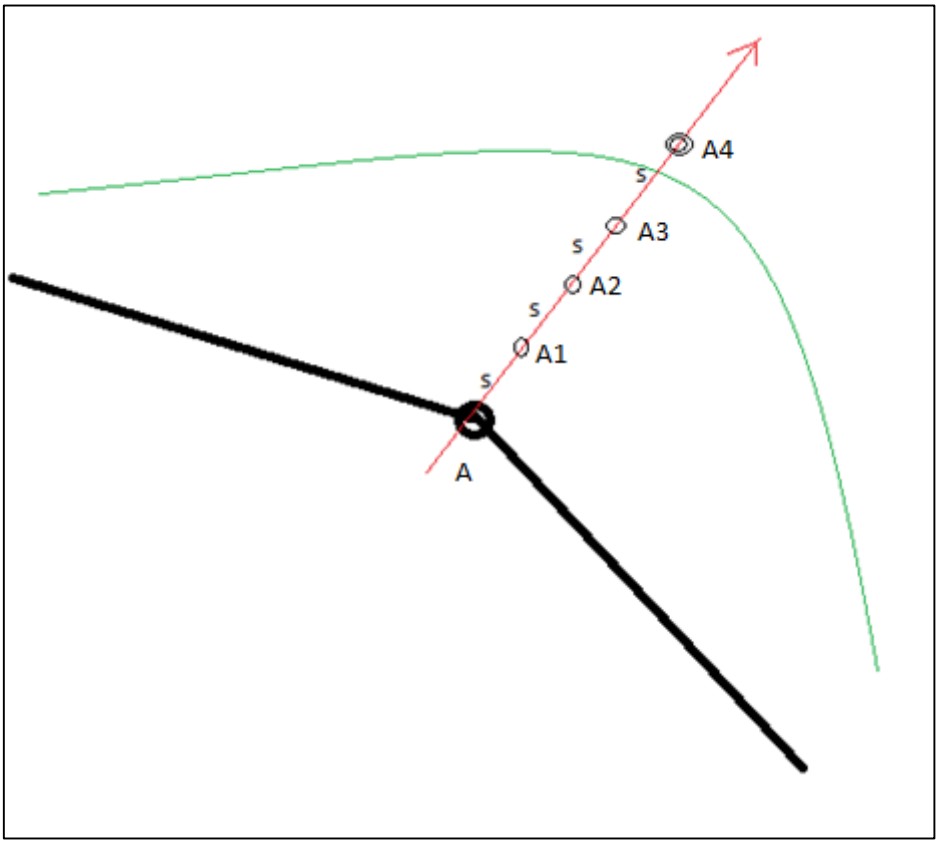

Figure 3.5 Checking error by taking step size s.

Now, we check for the exact position using binary search. Finally we get a point where intensity almost equals the threshold.

3.6 Changing the edges

The distance between this point and where intensity is matched is the error.If the error is more than tolerance, all the manhattan edges assigned for this point has to be fine-manhattanized so that the contour generated by the simulation of newly formed manhattan polygon comes closer to target contour. If the error doesn't decrease in the first iteration, replace the present edges with edges formed by lesser kernel size.

Chapter 4

Results

In this chapter, we will go through the experimental results of the problem. Fig. 4.1 shows the entire layout of the patterns. Green contour represents target contour, sky blue one represents normal simulated contour, pink one represents simulated contour (result). In section 4.1, results in which edges were fine manhattanized are shown. In section 4.2, results in which edges replaced with lesser kernel size edges are shown.

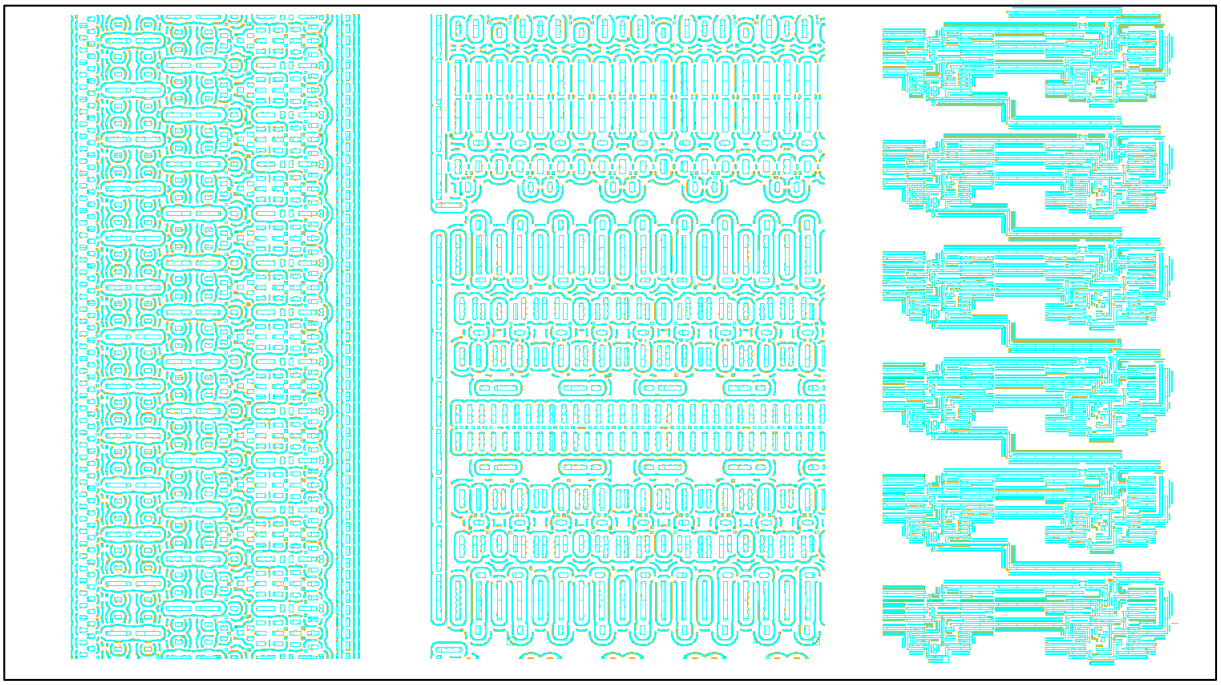

Figure 4.1 Entire Layout of the patterns.

4.1 Fine Manhattanized Edges

When we zoom in the layout shown in Fig. 4.1, we can see the polygons shown in Fig. 4.2. In Fig. 4.2, green contour represents target contour, sky blue one represents normal simulated contour, pink one represents simulated contour after fine-manhattanization, red one represents fine-manhattanized edges. At most of the places, error is been reduced.

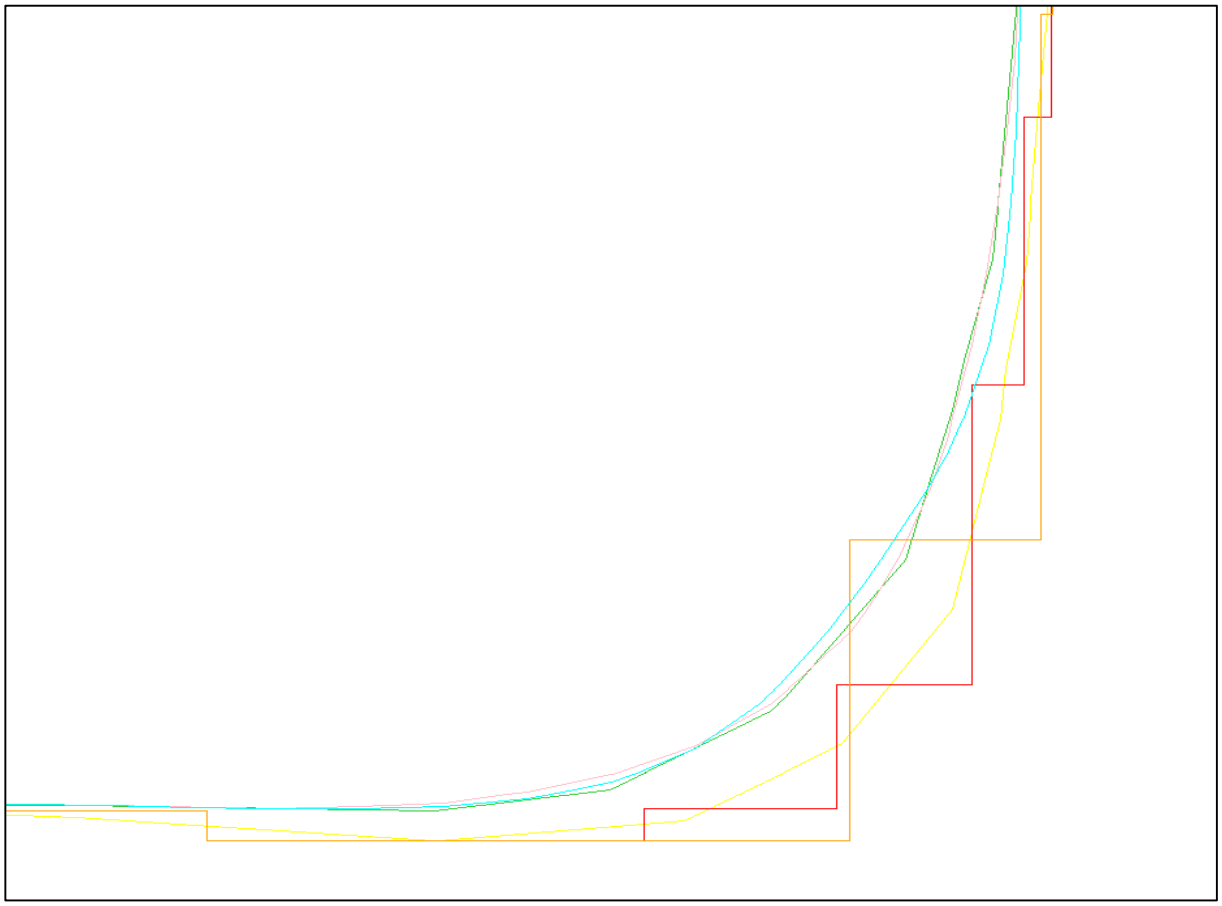

Figure 4.2 Zoom in contours of pattern1

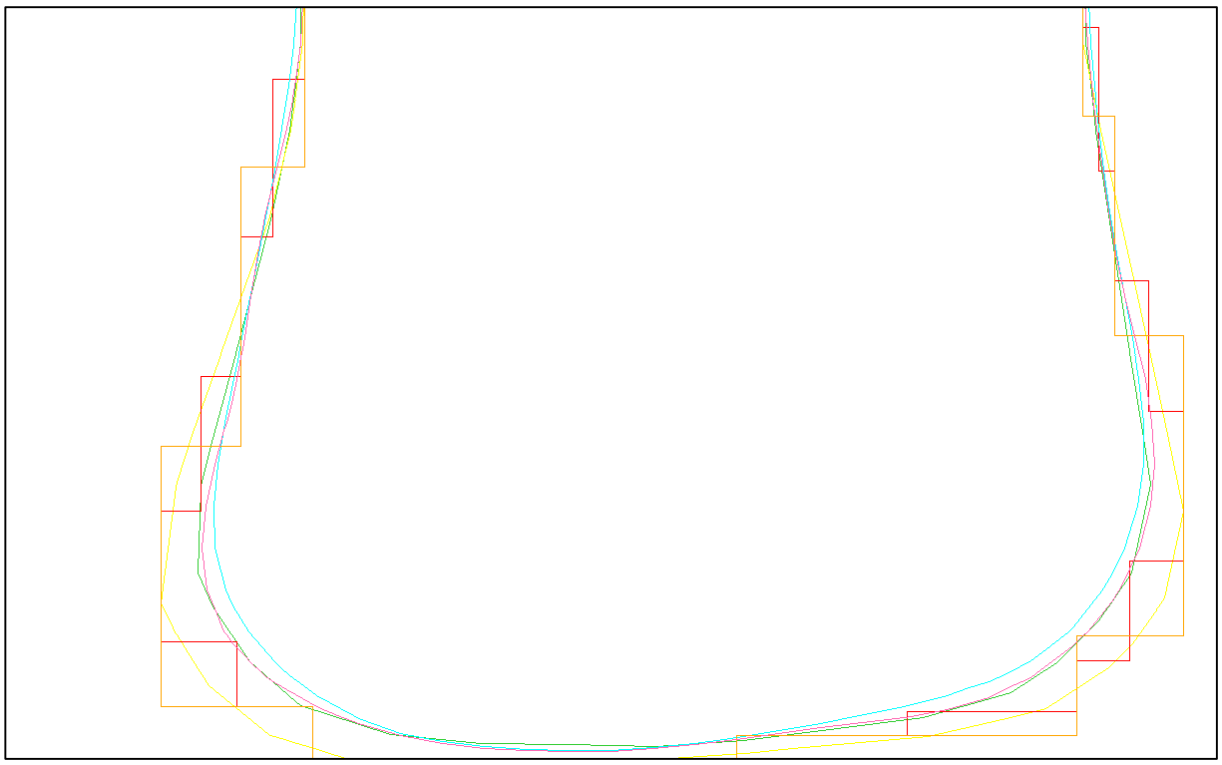

Figure 4.3 Zoom in contours of Pattern2

In Fig. 4.3, we can observe that contour shape was changing only at a places where error was observed. Along the vertical line and horizontal line, as the contour was close to the initial contour. Fig. 4.4 shows us the entire polygon. On the right side of the polygon, error was reduced to a greater extent. Whereas in the left below corner, error was not reduced. This is because we choose evaluation points at 100nm distances. The evaluation point may not be in the area was occurred. So, if we reduce 100nm distance, we may be able to reduce error at more number of places.

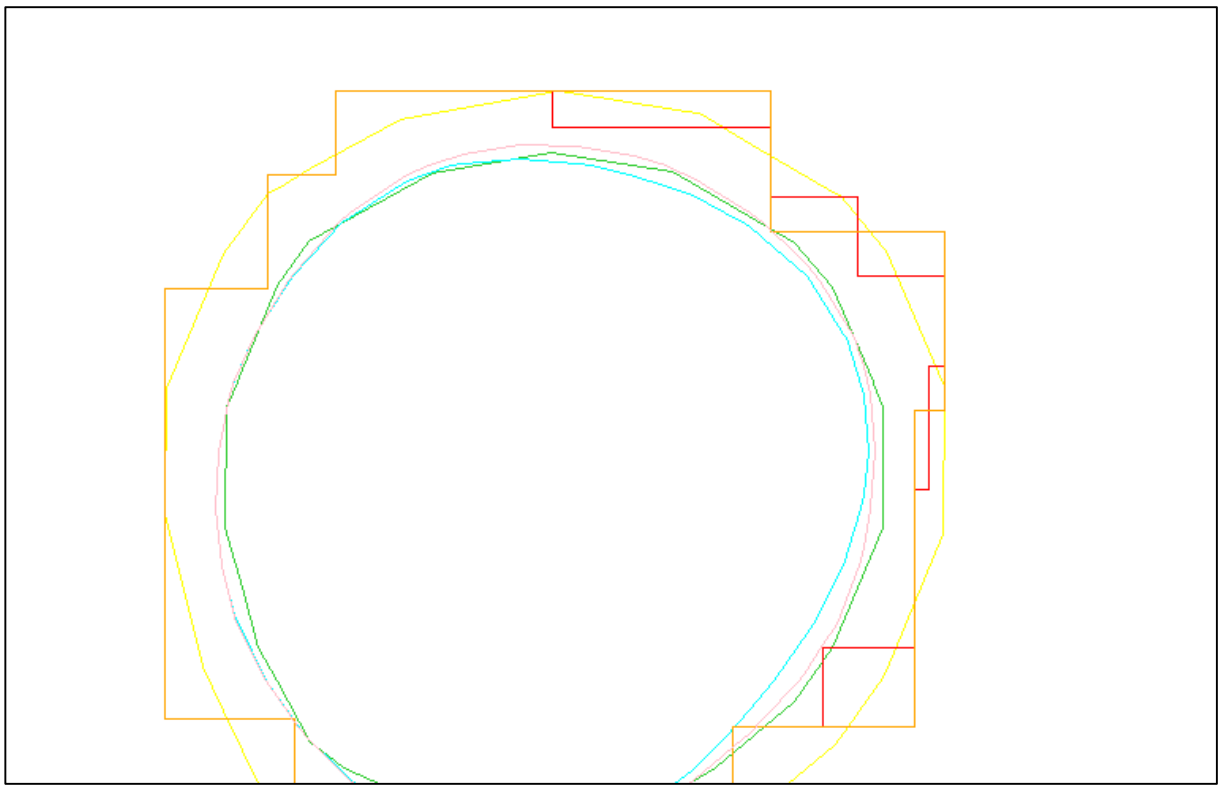

Figure 4.4 Circle shaped Pattern

4.2 Lesser Kernel Size Edges

In section 4.1, edges were shortened to reduce the error between the patterns. Now, in this section, edges were replaced with the edges formed by keeping the kernel size small compared to the ones before. Green contour represents target contour. Sky blue one represents normal simulated contour. Pink one represents simulated contour after fine-manhattanization. Red one represents fine-manhattanized edges.

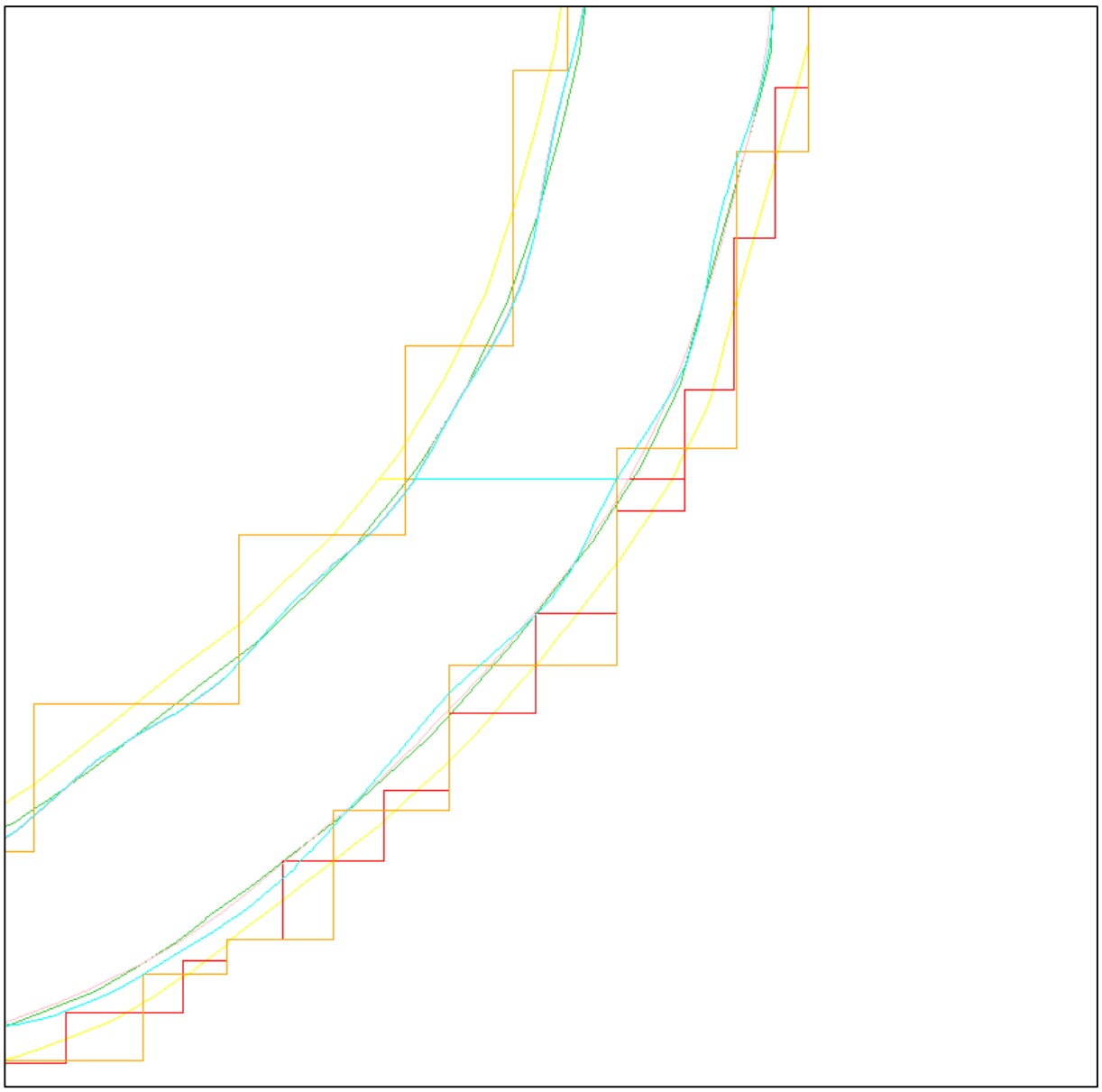

Figure 4.5 Zoom in contours of Pattern3

In Fig.4.5, error was reduced at all places, unlike in Fig.4.4, to a greater extent. This is because we choose evaluation points at 100nm distances. We would have got all the evaluation points where error was occurred. So, if we reduce 100nm distance, we may be able to reduce error at more number of places. But in this case, there is no need to reduce the distance as we got all the regions covered.

Figure 4.6 Vertical contours of pattern

In Fig.4.6, we can observe that contour shape was changing only at a places where error was observed. Along the vertical line and horizontal line, as the contour was close to the initial contour.

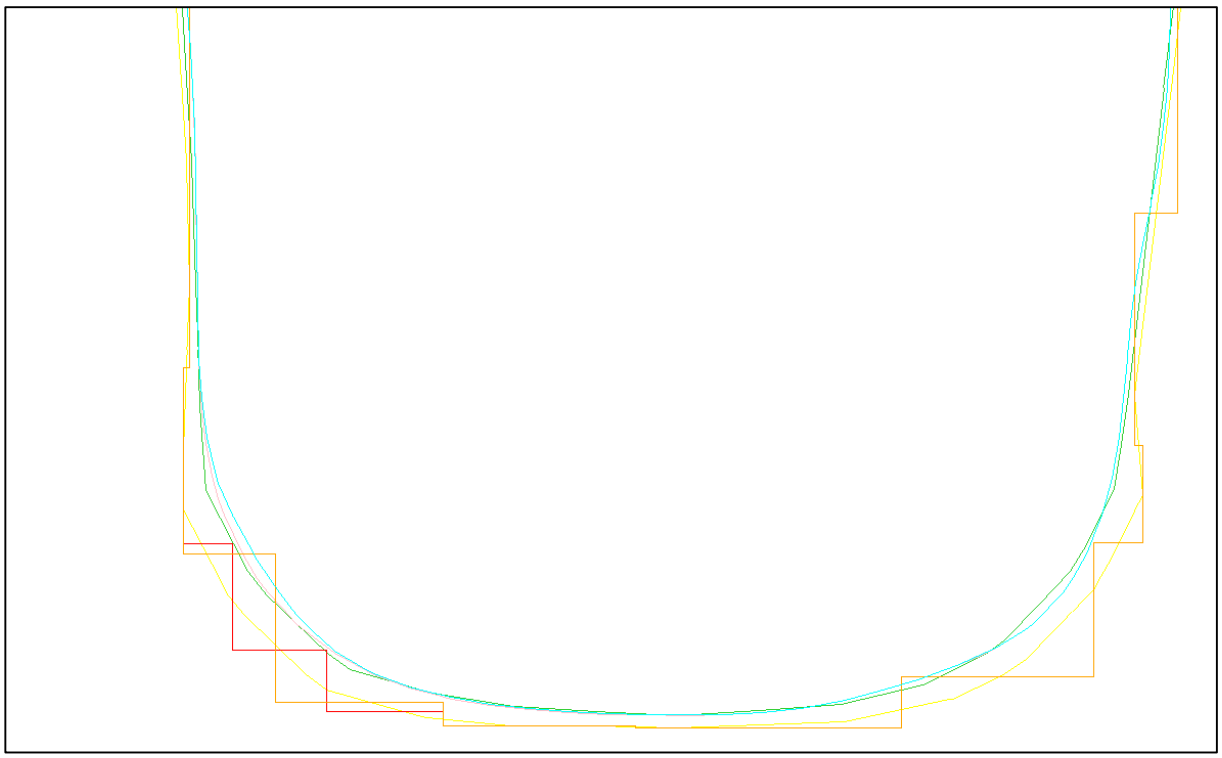

Figure 4.7 Zoom in contours of pattern5

Chapter 5

Future Work and Conclusions

Since we are choosing evaluation points on keeping them 100 nm distance apart, we were able to reduce errors in only some regions. So, error in the regions where error was more and we couldn't choose that point as our evaluation point was not reduced. So if we could choose more number of evaluation points, we can reduce error at more number of places. This can be done by reducing the 100 nm distance constraint. But when we reduce the 100 nm, it will take more time to process all the evaluation points and their assigned edges. So, instead of that, there is a scope to choose evaluation points at our will. If we can choose the evaluation points exactly where the error is occurring, we can get better results.

REFERENCES

[1] E. Guditz, "Machine Preparation of Printed-Wiring Masks," in IRE Transactions on Product Engineering and Production, vol. 5, no. 3, pp. 22-26, November 1961.
doi: 10.1109/TPEP.1961.1136126

[2] D. Z. Pan, "Lithography-aware physical design," *2005* 6th International Conference on ASIC, Shanghai, 2005, pp. 1172-1173.
doi: 10.1109/ICASIC.2005.1611242

[3] B. Yu, Jhih-Rong Gao and D. Z. Pan, "L-shape based layout fracturing for e-beam lithography," 2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC), Yokohama, 2013, pp. 249-254.

[4] https://www.azonano.com/article.aspx?ArticleID=3219

[5] D. K. Woo, K. Kane, S. K. Lee, 2008 Fabrication of a multi-level lens using independent-exposure lithography and FAB plasma etching, J. Opt. A: Pure Appl. Opt, 10 044001 1 6