Allocation of Fire Detector in Most Economical Way

by

Tan Fang Yee

Dissertation submitted in partial fulfillment of

the requirements for the

Bachelor of Engineering (Hons)

(Electrical and Electronics Engineering)

SEPTEMBER 2012

Univerisiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Allocation of Fire Detector in Most Economical Way

by

Tan Fang Yee

A project dissertation submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONIC ENGINEERING)

Approved by,

____

(DR VIJANTH SAGAYAN)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

September 2012

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except in the references and acknowledgements, and that the original work contained herein have been undertaken or done by unspecified sources or persons.

TAN FANG YEE

ABSTRACT

In oil and gas industry, fire outbreak is one of the main accidents happened in onshore and offshore plant. Hence, a well-designed fire detection system that offer an adequate coverage on the plant is crucial to safeguard the hazardous area and reduce the damage that caused by fire accident. Most of the modern day detector monitoring system uses visual detector as it provides superior advantage over other conventional detector, such as immunity against dust, reflection and flares. In real life situation, the placement of fire detector in some plant is being done manually by engineers and this might left out certain hazardous area and cause the plant to be on risk. Service to evaluate the effectiveness of fire detector placement is offered by certain professional companies such as Micropack yet they are very costly. Besides, idea to implement the assessment after placement might result in the redundant use of detector, which is unadvisable from the economy perspective. The aim of this project is to offer a simple assessment method that met the possible balance between safety and economy. This could be achieved by identify the best placement of detectors with optimum coverage, which minimum number of detector will be required. A methodology named detector mapping assessment is being introduced to determine the degree of fire detector coverage with respect to the given hazardous grade map.

ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude toward my supervisor, Dr. Vijanth Sagayan for his great supervision, continuous support, precious guidance and understanding which truthfully help me throughout completing this project. His previous experience in Oil and Gas field has offered me a better understanding on my project and beneficial for me to enhance the current solution in the industry.

My gratitude is extended to Mr Tan Ping Ying, Instrument Engineer at PETRONAS for his kind assistant and sharing in enhancing my technical knowledge on the current project. In addition, a special note of appreciation I would like to extend to Dr. Rosdiazli B Ibrahim, the interval evaluator of my final year project for his time and effort to evaluate my presentation and reports, and offering his valuable opinion in heighten my project.

I would like to convey my appreciation to all the lecturers from Electrical and Electronic Engineering Department for imparting the engineering knowledge to us throughout the five years of study, which provide me the skills needed to develop my final year project.

Lastly, I wish to express my love and gratitude to my beloved families and friends, for their understanding & endless love, through the duration of my studies.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

CHAPTER 1. INTRODUCTION

Fire Detector Mapping Allocation System is being chosen as my project in order to provide an alternative to current oil and gas industries by identifying the most economical placement of fire detectors that offer the optimum coverage. Nowadays the conventional fire detectors such as IR and UV detectors have been replaced by more advanced visual fire detector that operates with similar properties like CCTV camera. A methodology named fire detector mapping is being used to assess the placement of detectors throughout the sites, with the output of minimum number of detector that meet the objective of this project.

1.1 Background of Study

All the petrochemical plant are filled with equipment that sensitive toward flame and fuel pump that carries flammable products, hence fire detection system has become one of the most important tools in plant to safeguard hazardous area by monitoring the plant and notifying the control station on any fire accident so that safety precaution could be taken at the early stage before critical damage happened. Some well-designed fire detection system shall come along with the automatic flame mitigation features. In order to ensure the adequate detection on the hazardous area in plant, a proper placement of detectors is crucial when designing the fire safety system. A methodology named detector mapping assessment is being introduced to determine the degree of fire detector coverage with respect to a given hazardous grade map. For the existing fire system, this methodology is being used to certify the adequacy of the system and validate the accuracy of previous risk assessment. [1]

1.2 Problem Statement

Safety measure and precaution are very crucial in a petrochemical plant as most of the instrument are sensitive to flame and required high attention. Manual placement of the fire detector even by the experience field engineer might left out any hazardous area and cause the plant to be in risk. Hence in order to fulfill the safety requirement, every petrochemical plant shall undergo an assessment on fire safety system in the plant to ensure coverage of the detectors is up to the safety level. Companies like Micropack do offer the assessment software that able to determine the effectiveness of the existing detector placement. With this, the engineer would determine the need of installing additional detectors to safeguard the site. However in term of economy this would be unadvisable as outsourcing the assessment are very costly and introduce the assessment after the placement would result in the redundant use of detector.

Besides, the current study on the optimum detector placement has assumed the use of one single model of fire detector with common specification in every application. Although this method offers a more simple computation, yet it is impractical to assume every plant will share the same model of visual sensors. In addition, most researches have excluded the effect object occlusion in the study. These two limitations have deterred the effectiveness of the monitoring system.

Hence my project is meant to resolve these issues by developing a simple assessment of detector placement that able to achieve the best possible balance between safety and economy.

1.3 Objective and Scope of Study

The following shows the objectives of this project:

i. Develop an effective assessment technique that meets the economical purpose.

ii. Develop an assessment method of detector placement that meets the safety requirement.

The main focus in this paper is to identify the best way to compute the coverage of each detector location. Few articles are studied and analyzed to understand the existing assessment method in the industry. This project is looking from the economical perspective by evaluate the minimum number of fire detector required while maintaining the safety level.

1.4 Relevancy and Feasibility of the Project

This project is contributing to the oil and gas industry as it provides a simple alternative to improve the safety system in field. Besides, it is very helpful for me as a final year student who is ready to go into oil and gas field by offering me a basic exposure to safety precaution system in the oil and gas plant.

This project is given an adequate time frame which is nine months to be accomplished. A gantt chart is constructed in order to ensure the smooth flow of the whole project. In short, I could say this project is feasible and could be completed in time.

CHAPTER 2. LITERATURE REVIEW

Before conducting the research on optimum placement method of visual sensors, a brief study on the current existing flame detector has been done. There are three common flame-sensing technologies in use today, all come with their own advantages and disadvantages [2]:

i. Ultraviolet (UV): As all fire emits radiation, this sensor could act as good general-purpose fire detectors. However, the response of this device could be inhibited by the hydrocarbon films.

ii. Infrared (IR): This sensor operates by analyzing the amplitude and flicker frequency of the heat element from the fire. It able to counter the main limitation of UV, yet black body radiation and water on the optical surface will decrease the sensitivity of the device.

iii. Visual Flame Detectors: This device offer video imagining based technique, just like CCTV. The advantages of this device are it is free from the distortion by flare and reflection as well as the effect of hydrocarbon films, black body radiation and water. The only downside of this device is it only responds to a fire with certain level of brightness.

Hence the use of visual flame detectors is recommended as it able to counter the drawback of UV and IR sensors. Besides than issuing the flame detection, this device also provide a visual indication to the operators in control room. Through the video image, operators could have better knowledge on the fire condition and able to make a correct execution before endangering personnel.

Every visual sensor has their own range of view and intrinsic parameter, but all of them share a common shape of field of view which is conical shape as shown in Figure 1. The assessment method proposed in this paper is in two dimensions dual views, which are the top view and side view. The field of view of the visual sensors is assumed to be in triangle shape from top, circle from side.

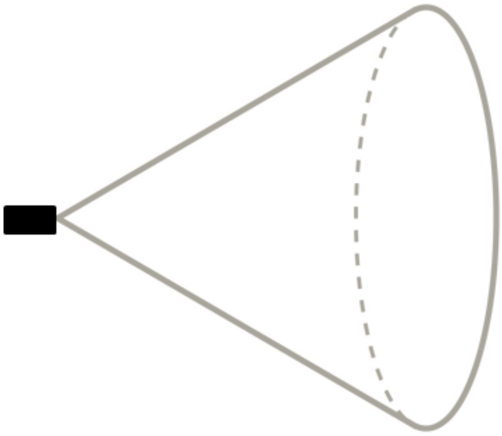


Figure 1: Field of view of visual detector in three dimensions.

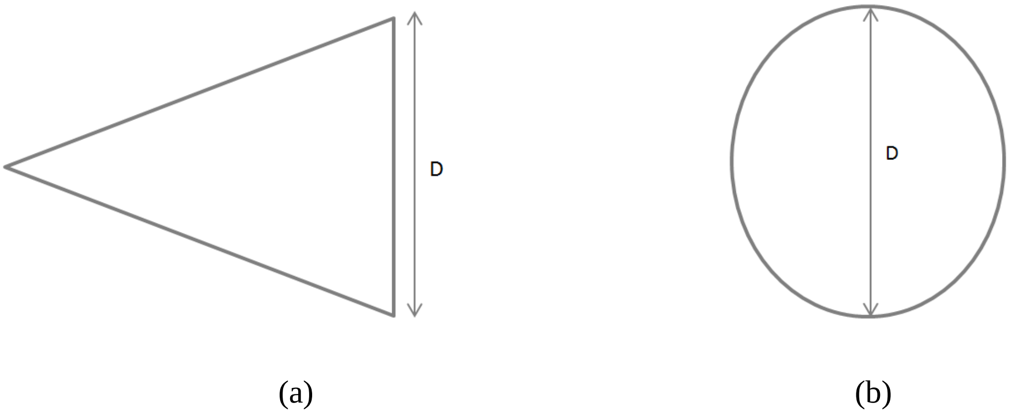


Figure 2: Field of view of visual sensor from top (a) and side (b) in two dimension.

An intense study has been done on current research papers that related to optimal sensor placement. Previous Final Year Project under supervision of Dr. Vijanth has become one of my main references. Yet that project has poses few limitation as it does not take the occlusion of object into account. Besides, the testing is being done in a small scale and the possibilities of the detector position appear to be very little. Analysis on coverage of detector based on the detector specification has been excluded hence reduced the effectiveness of the assessment. His project offer a 2.5 dimension analysis from the top view and side view. However both of the analysis is being done separately without a combine analysis afterward.

Art gallery problem [3] is the classical example introduced by Victor Klee in 1973 regard to the placement of the guard in such a way that the whole gallery could be observed by minimum number of guard. However, the weakness in this study is there is no restriction on the field-of-view of the visual sensors, which is impossible in real world application.

Later, Horster and Lienhart [4] have work on the similar assessment as they focus the placement of visual sensors with maximum coverage. Their objective is to achieve the maximum surveillance coverage in the best camera poses with minimum number of sensors. Different algorithms are being proposed and results are presented in their study in order to obtain the best solution and achieve the objective. However their study has neglected the occlusion of the object.

Backer et al [5] proposed the used of voting scheme to identify the coverage of the video sensors in 3D form. The assessment place sensors on a point in the volume of interest and rotated along the three axes to identify the number of perceptible point in volume of interest. By using accumulator array and greedy heuristic, this system will sort the minimum number of sensors; later identify the location of the sensor that covers the most number of perceptible points. Yet, object blockage does not being included in their study as well. In some application, a more simplified analysis is preferred due to time constrain. This study is in 3D scare hence the computation has high level of complexity and time consuming, which is not favorable in some application.

Development of genetic algorithm is being developed by Indu.S, Chaitanya and Manoj [6] to resolve the optimum sensor placement issue. They proposed that this method converge faster with in a simpler way compared to linear programing approach. Further study on the Evolutionary computing method recommended by G. Olgue [7] has been done to optimize the position and the tilt angle of the visual sensors. Besides, this paper has included the consideration on the priority of certain area in the analysis, which is aligning with the criteria of the assessment of my project.

The common weakness of these few paper is that they did not consider the object blockage in the assessment. A testing using exhaustive search algorithm [8] on the effect of occlusion has been conducted by P.Y.Tan [9]. The percentage of coverage with and without occlusion is compared and the result shows that taking into account on the occlusion give a significant difference on the percentage of coverage. Yet, this proposed method has neglected the different of field-of-view of different type of detectors.

In order to compute the area of blockage, the object need to be extract from the hazardous map by one of the image segmentation technique, edge detection. [10] Edge detection is an imaging technique that able to provide the information regard to the boundary of the object. Comparison has been done among different type of edge detection technique by Nadernejad [11] and the result has shown that Canny edge detectors so offer the better overall performance.

CHAPTER 3. METHODOLOGY

3.1 Research Methodology

After the intense literature review, an assessment methodology is proposed with the features as shown:

i. The assessment will be done in two dimensions dual views, which are top view and side view.
ii. Static detector is used.
iii. Placement of detectors on the top edge of side wall only.
iv. Visual of detector has a triangle field of view from the top, circle coverage from the side.
v. As different detectors are having different range of field of view, hence user are allowed to define the angle of field of view of the detector and the angle of detector placement. The program will generate the coverage with respect to these user defined parameters.

The input of the assessment is the hazard map and user defined angle of placement and angle of field of view of the detector; while the output is the minimum number of required detectors as well as the placement of detectors that perform the desired result. All possible placements that met the requirement are displayed in the window, but the best one is suggested by the program. In this assessment, top view analysis would be the main reference. The program will only proceed to the side view analysis when the coverage of particular placement has met the safety requirement. The purpose of side view analysis is to provide the engineers a better view to visualize the actual coverage of the particular detector placement.

3.1.1 Input of the assessment

Grade map is a hazard map that being used by engineers to evaluate the hazardous level of an area in the plant. The standard of the grade map as shown in Figure 3 is being set by Micropack [12], a fire and gas detections Consultancy Company that specializes in fire and gas mapping studies for oil and gas industry. The hazard level is based on the fire grading of the object. Table 1 shows the colour scheme for different hazard level in grade map and the minimum coverage required at respective level to meet the safety requirement.

Table 1: Colour scheme for each hazard level in grade map and the respective minimum safety level

Colour	Red	Orange	Green
Hazard level	High	Moderate	None
Minimum Coverage	0.95	0.75	0.35

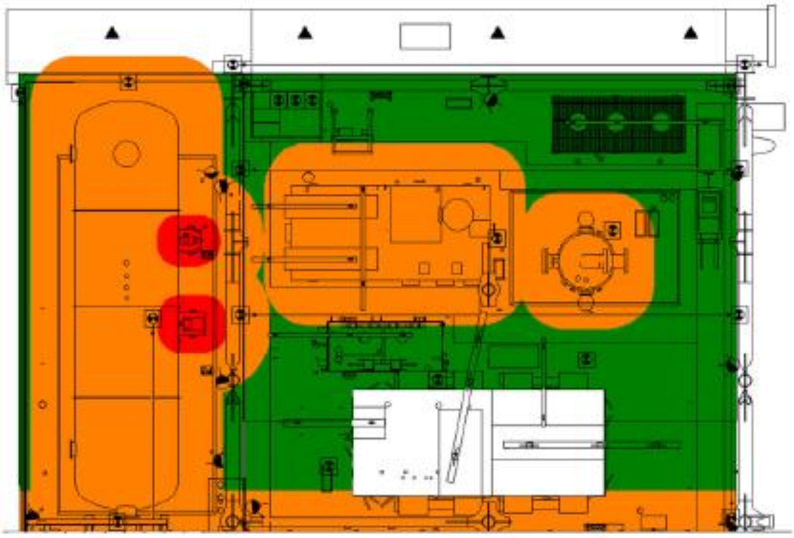


Figure 3: Grade map according to Micropack standard.

Since the range of field of view of the visual sensors is vary according to the specification of detector model and placement angle, the program required user to input the angle of the field of view of the detector together with the angle of detector placement.

.

3.1.2 Assessment

A flow chart for the algorithm is constructed at the beginning stage to give a brief idea on the flow of the assessment.

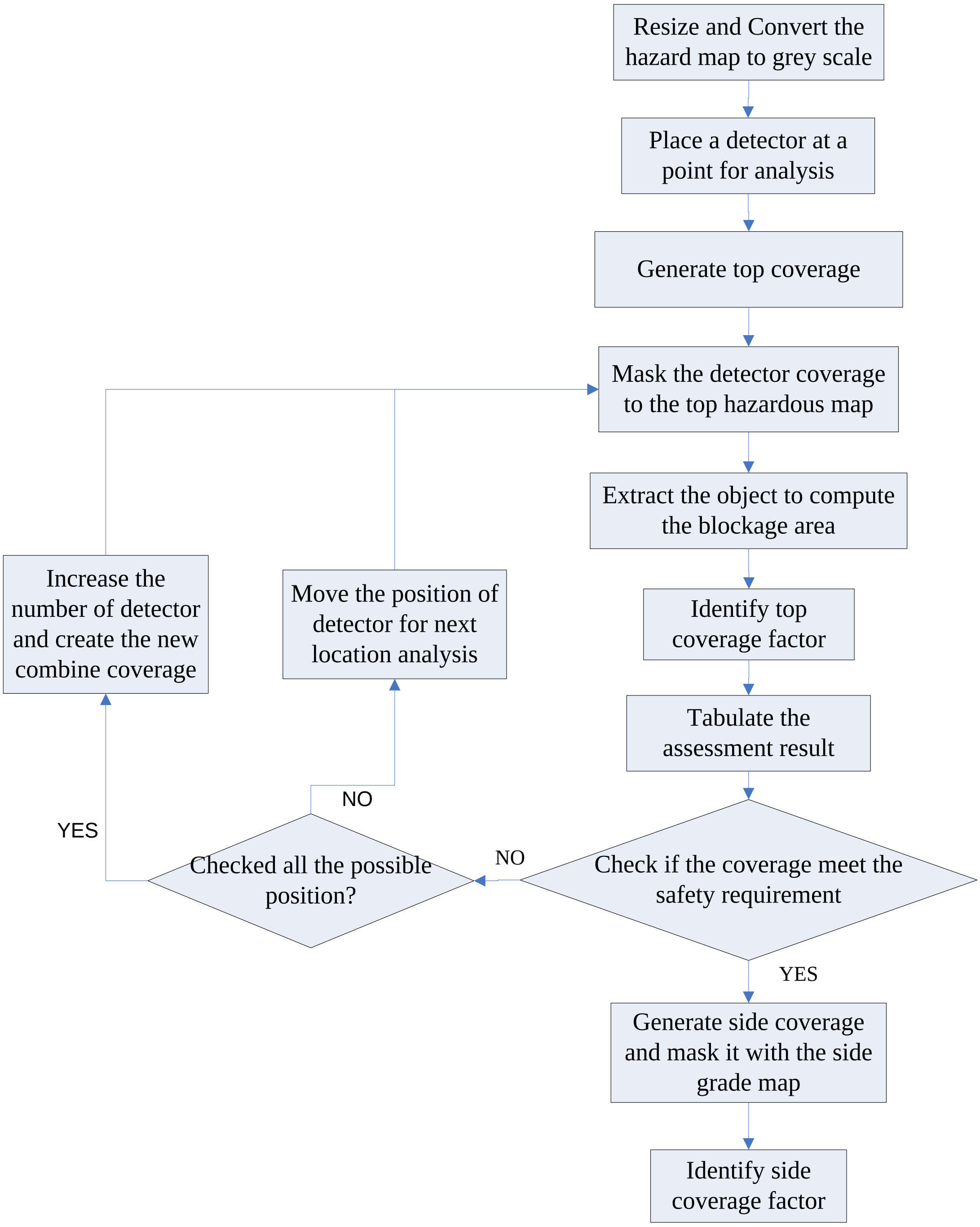
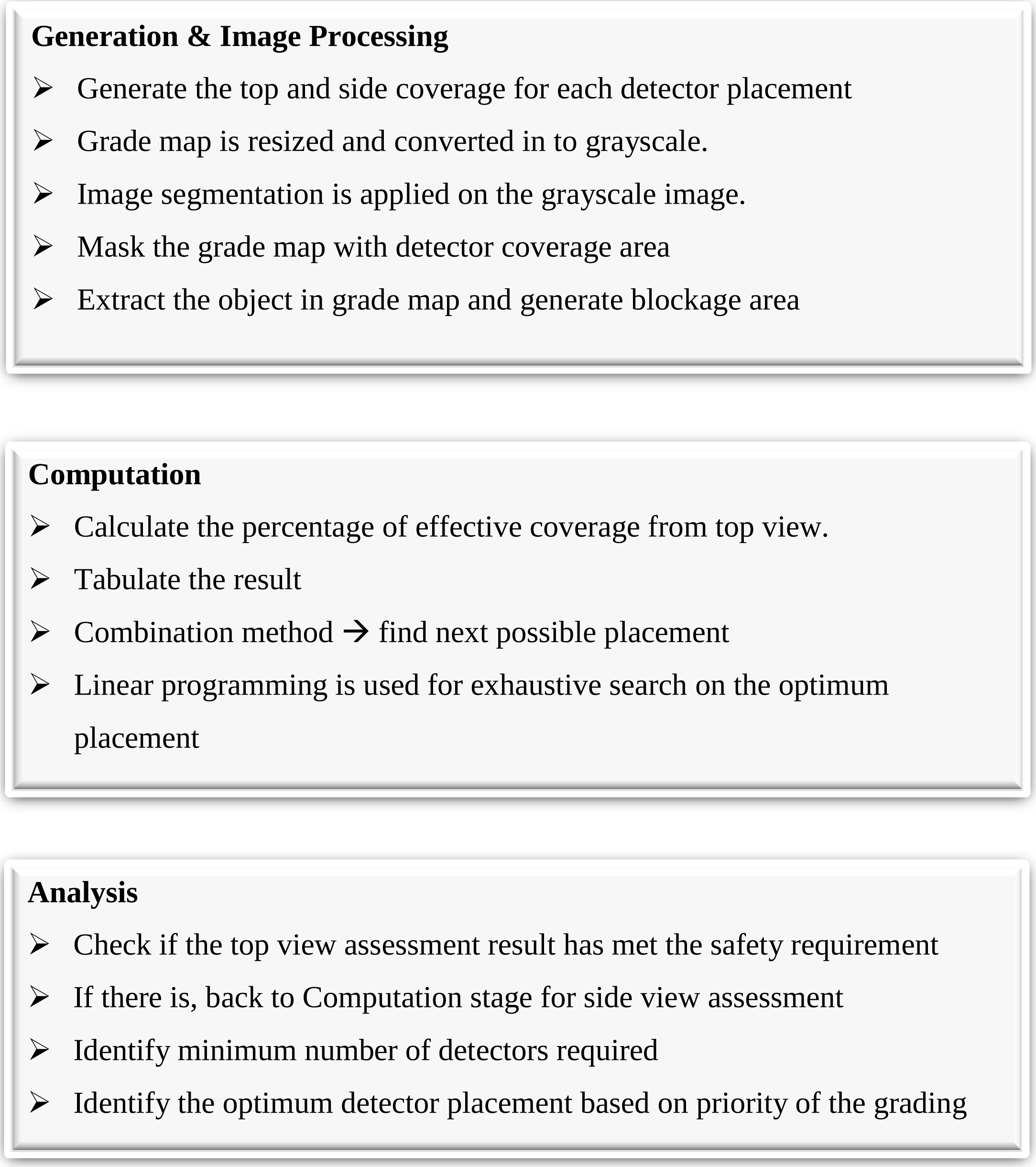

Resize and Convert the hazard map to grey scale
Place a detector at a point for analysis
Generate top coverage
Mask the detector coverage to the top hazardous map
Extract the object to compute the blockage area
Identify top coverage factor
Tabulate the assessment result
Check if the coverage meet the safety requirement
YES
Generate side coverage and mask it with the side grade map
Identify side coverage factor
NO
Checked all the possible position?
NO
Move the position of detector for next location analysis
YES
Increase the number of detector and create the new combine coverage

Figure 4: Flow chart of the program.

Firstly the grade map needs to be resized and converted into grey scale for the ease of analysis. A triangle shape of detector coverage is generated for the top view analysis. This coverage is then masked to top view grade map and the overlapping area of coverage and grade map is being segmented to for the blockage computation. Next, the outline of the object within the map is extracted by Canny edge detection technique for the computation the blockage area. After subtracting the area of blockage from the area of coverage, the effective area of coverage is obtained and analyzed by the program whether it has met the safety requirement. If it is, the program will proceeds with the side view analysis. Else, the program will move to the next position and proceed with the same assessment. If none of the possible placement of one detector has met the safety requirement, the program will increase the number of detector and continue with the similar assessment. Coverage of detector would be generated again based on the new combination. Linear programming is being used for this exhaustive search to identify the optimum detector placement.

Finally, the program will gather the point of interest of all possible detectors' placements and identify the best detector placement that achieves the safety requirement with minimum number of detectors.

Later, the development of the assessment is divided into three stages according to the program flow, which are generation & image processing, computation and analysis as shown in Figure 5 and would be explained thoroughly in next section.



Generation & Image Processing

- Generate the top and side coverage for each detector placement
- Grade map is resized and converted in to grayscale.
- Image segmentation is applied on the grayscale image.
- Mask the grade map with detector coverage area
- Extract the object in grade map and generate blockage area

Computation

- Calculate the percentage of effective coverage from top view.
- Tabulate the result
- Combination method → find next possible placement
- Linear programming is used for exhaustive search on the optimum placement

Analysis

- Check if the top view assessment result has met the safety requirement
- If there is, back to Computation stage for side view assessment
- Identify minimum number of detectors required
- Identify the optimum detector placement based on priority of the grading

Figure 5: Three stages of algorithm development.

3.1.2.1 Generation & Image Processing

Figure 6 shows side view on how the detector would be placed at the top edge of the side wall. The two user defined parameters which are the angel of field of view and the angle of placement are illustrated in the figure as well. The triangle field of view of detector is generated based of these two parameters.

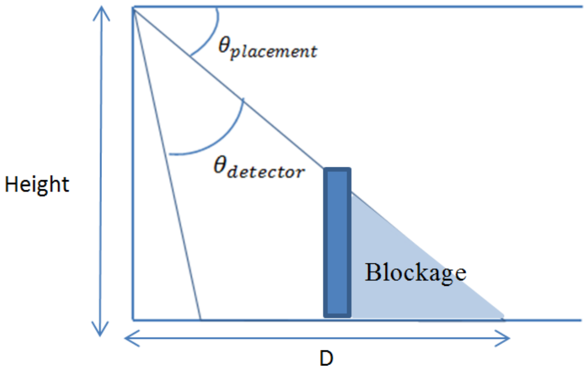


Figure 6: Detector placement.

The distance of the coverage could be compute by Eq. (1). This distance would be the length of perpendicular bisector of the triangle coverage as shown in Figure 7. The program will compute the three point of tip of triangle, and generate the triangle coverage as an image.

$$D = \frac{Height}{\tan \theta_{placement}} \tag{1}$$

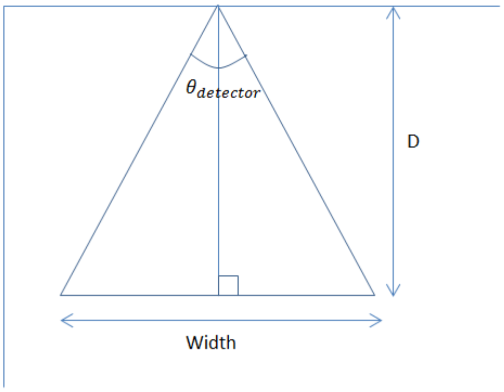


Figure 7: Coverage of detector from top view.

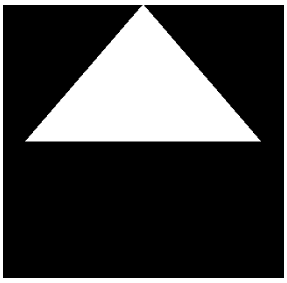


Figure 8: Top view coverage of detector generated by the algorithm.

The radius circle coverage from the side view would be generated based on the width of the triangle with Eq. (2) and Eq. (3). Figure 9 shows the circle coverage of detector generated by the algorithm.

$$width = 2 \times R \tag{2}$$

$$R = \frac{D}{2}\ \tan(\frac{\theta_{detector}}{2}) \tag{3}$$

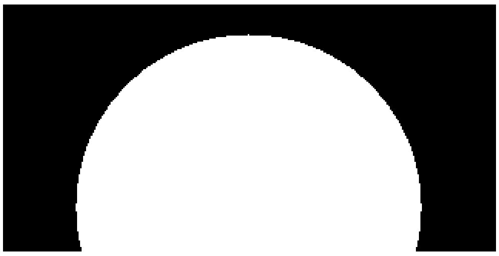


Figure 9: Side view coverage of detector generated by the algorithm.

The grade map that generated with Paint according to the colour scheme defined by Micropack is resized into the size of 320x320 pixels for top grade map, 320x160 pixels for side grade map. Nearest-neighbor interpolation method is imployed as it offers the fastest operation. Later the image is converted into grey scale as shown in figure 10 for the ease of assessment.

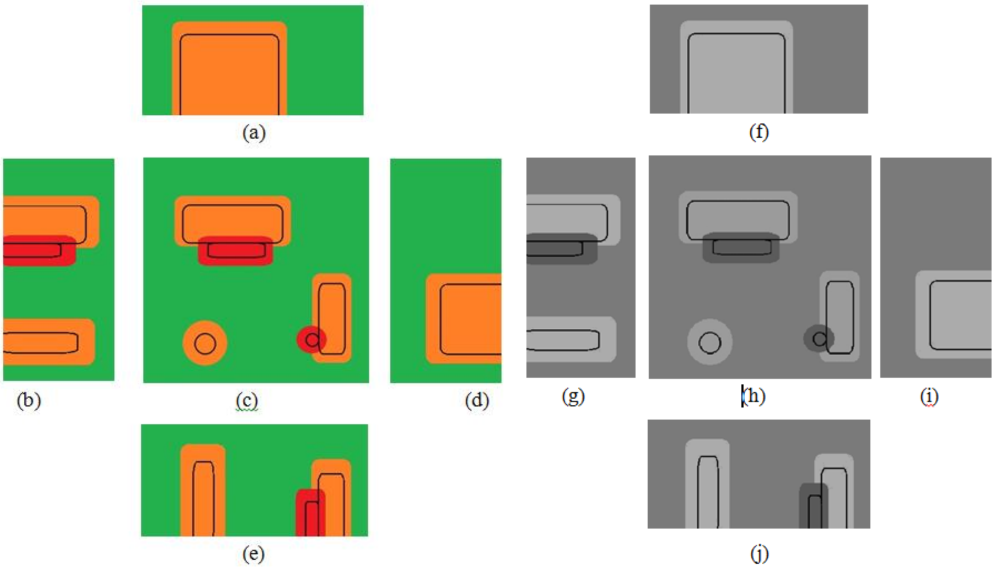


Figure 10: Grade map from top view (c) and side view (a), (b), (d), (e) in colour scheme is converted into gray scale (f), (g), (h), (i), (j).

In gray scale, maximum bit which is 1 indicate the white bit, while minimum bit which is 0 indicate the black bit. Image thresholding method is applied to the grayscale grade map to segment the colour of the map. Pixel value range from 0.3 to 0.4 would be store as 0.35 indicate the red pixel; pixel value range from 0.43 to 0.53 would be store as 0.48 indicate the green pixel while pixel value range from 0.60 to 0.73 would be store as 0.67 indicate the orange pixel.

i. Red: 0.3 - 0.4 → 0.35
ii. Green: 0.43 - 0.53 → 0.48
iii. Orange: 0.60 – 0.73 → 0.67

The monochrome images of grade map and detector coverage are masked by storing each of them in different frame of RGB. The grade map will be store in red frame while the coverage area will be store in green frame as shown in Figure 11. In this assessment, there will be 12 possible placements as shown in Figure 12.

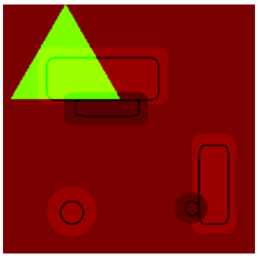
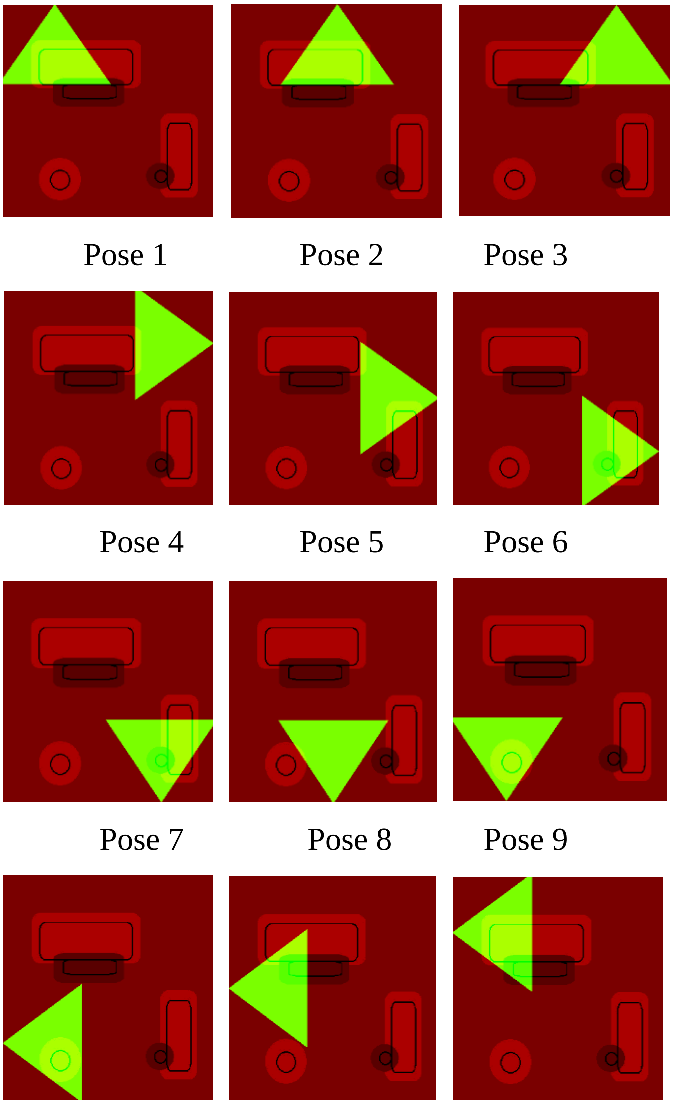


Figure 11: Masked image of grade map and detector coverage.



Pose 1 Pose 2 Pose 3

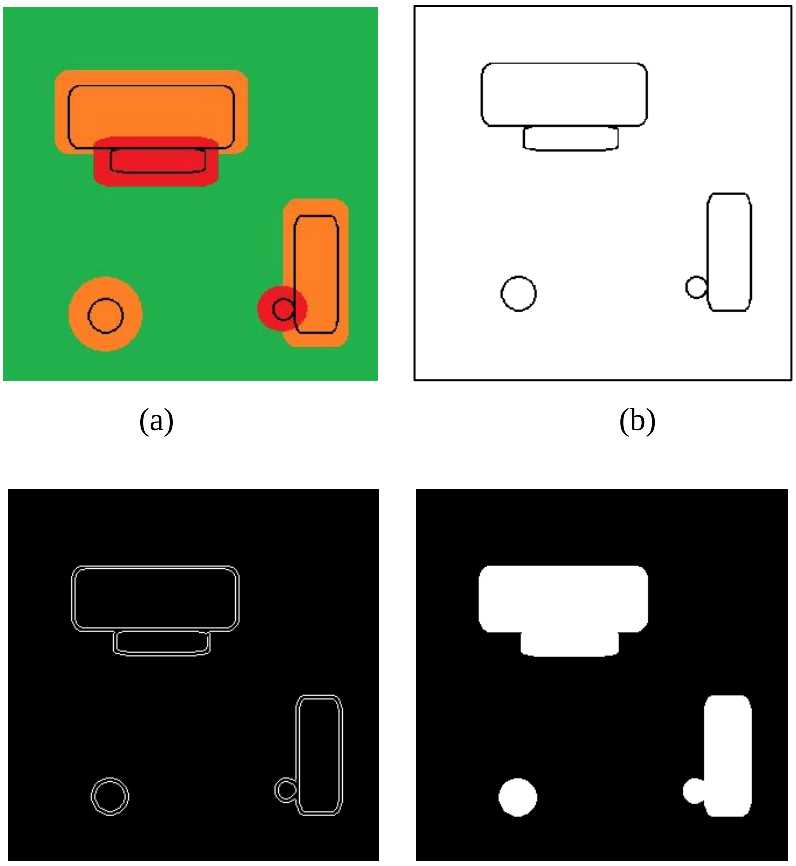
Pose 4 Pose 5 Pose 6

Pose 7 Pose 8 Pose 9

Pose 10 Pose 11 Pose 12

Figure 12: Twelve possible placements for one detector.

The object in the plant which outlined by black pixel is extracted by edge detection method to generate the blockage area of the object. As mentioned in the literature review, Canny edge detection method has been identified to be the most effective method that offer higher precision. Hence, this method is being introduced to for the object extraction as shown in Figure 13.

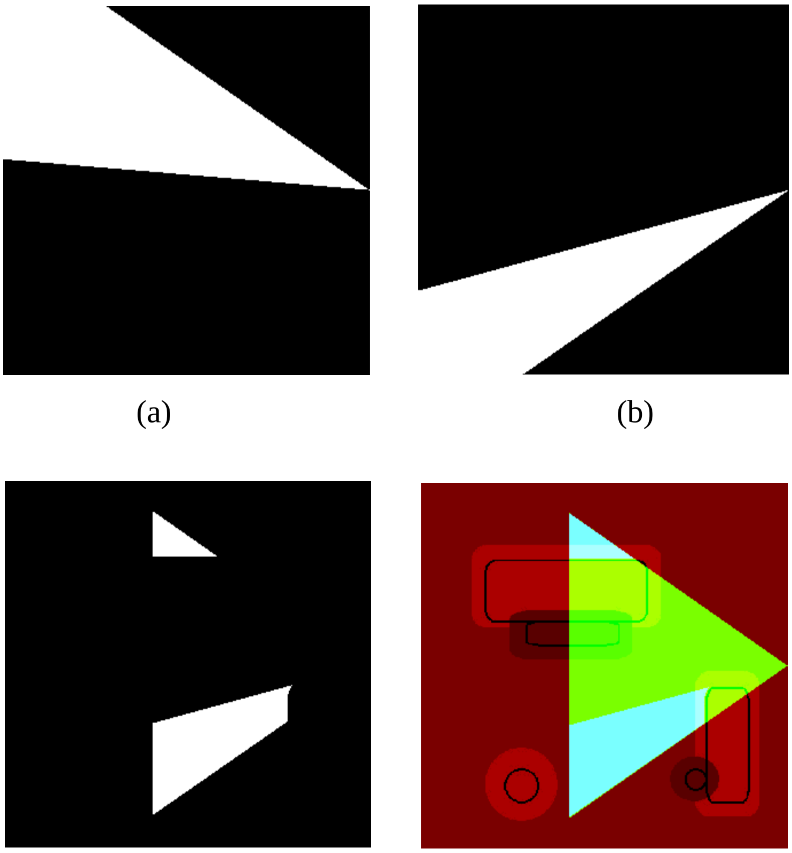


(a) (b)

(c) (d)

Figure 13: Object extractions by Canny edge detection method.

Later, the outline of the blockage area is computed from the point of detector placement as shown in Figure 14(a) & (b). Those pixel reside behind the object and within the outline would be set to white pixel indicate the blockage area as shown in Figure 14(c). After the blockage area is generated, it masked with the grade map and detector coverage for the computation of coverage factor in the next stage. In the masking process, the grade map is stored in red frame; detector coverage is store in green frame, while the blockage is being stored in blue frame as shown in Figure 14(d).



(a) (b)

(c) (d)

Figure 14: Generation of blockage area.

The blockage area within the detector coverage for all possible detector placements is computed with the same technique. Each of them is masked one by one for the computation of effective coverage in next stage, and analysis on the following stage.

If the analysis shows that none of coverage factor of single detector placement has met the safety requirement, the program will increase the number of detector, and generate the new coverage based on the possible combination of placement for more detectors. The combination of placement could be determined by combination method, whereas the number possible combination of placement is determined by Eq (4):

$$C_r^n = \frac{n\,!}{r!(n-r)!} \qquad (4)$$

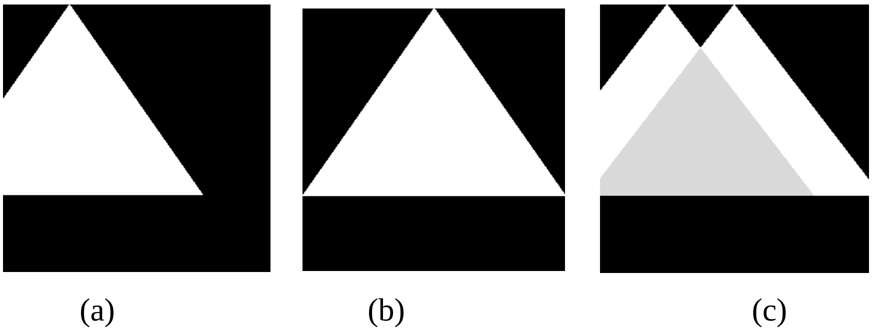
Where n is the total number of possible placement, r is the number of detector.

With twelve possible placements for one detector, there will be sixty-six possible placements for two detectors, two-hundred and twenty possible placements for three detectors, four hundred and ninety five possible placements for four detectors and so on. Table 2 shows all the possible combinations of placement for two detectors.

Table 2: Possible combination of placement for two detectors.

No.	Position 1	Position 2	No.	Position 1	Position 2	No.	Position 1	Position 2
1	1	2	**23**	3	5	**45**	5	12
2	1	3	**24**	3	6	**46**	6	7
3	1	4	**25**	3	7	**47**	6	8
4	1	5	**26**	3	8	**48**	6	9
5	1	6	**27**	3	9	**49**	6	10
6	1	7	**28**	3	10	**50**	6	11
7	1	8	**29**	3	11	**51**	6	12
8	1	9	**30**	3	12	**52**	7	8
9	1	10	**31**	4	5	**53**	7	9
10	1	11	**32**	4	6	**54**	7	10
11	1	12	**33**	4	7	**55**	7	11
12	2	3	**34**	4	8	**56**	7	12
13	2	4	**35**	4	9	**57**	8	9
14	2	5	**36**	4	10	**58**	8	10
15	2	6	**37**	4	11	**59**	8	11
16	2	7	**38**	4	12	**60**	8	12
17	2	8	**39**	5	6	**61**	9	10
18	2	9	**40**	5	7	**62**	9	11
19	2	10	**41**	5	8	**63**	9	12
20	2	11	**42**	5	9	**64**	10	11
21	2	12	**43**	5	10	**65**	10	12
22	3	4	**44**	5	11	**66**	11	12

The overall coverage of each combination is generated by the overlapping the two coverage at different position. Figure 15 shows the coverage at position one, position two and the combine coverage from both positions.



(a) (b) (c)

Figure 15: Detector’s coverage at position one (a), position two (b) and combine coverage from both positions (c).

Next, the object blockage is created based on the new generated coverage. However, the overall blockages of the new combination could not be generated in the same way as generation of the combined coverage. Some of the blockage within the coverage of position one might be covered by the detector at position two. In fact, the blockage area that fall outside the intersection of two detectors coverage would remain excluded; while for the area within the intersection of two coverage, if the area that is unseen by two detectors would be remain excluded, otherwise if the area is only hidden from one detector, it would be visible when two detectors are being used. Refer to Figure 16, area (a), (b) and (c) are the blockage area from the view of detector one, area (a) is the blockage area from the view of detector two, while area (a) and (b) are as the blockage area if two detectors are being used. After the new coverage and blockage are generated, they were masked with the top grade map with the same method as shown in Figure 17.

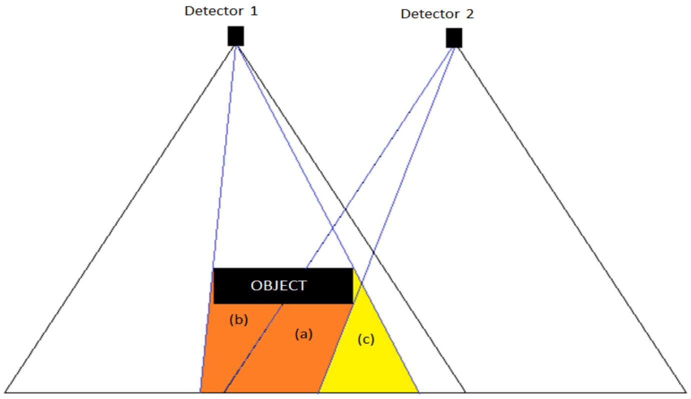


Figure 16: Blockage area from the view of two detectors.

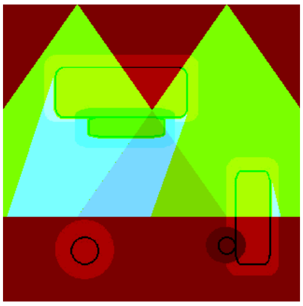


Figure 17: Coverage of two detectors on the top grade map.

If two detectors are insufficient to deliver the adequate coverage, the program will increase the number of detector to three and so forth. Similar method is employed for the generation of coverage and blockage. Figure 18 shows the coverage of three detectors placement and four detectors placement.

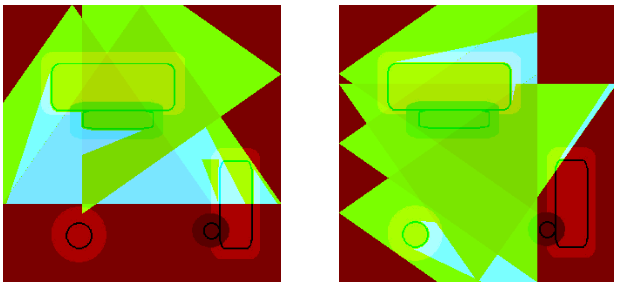


Figure 18: Coverage of three detectors (a) and four detectors (b) on the top grade map.

3.1.2.2 Computation

After obtaining the masked map that comprise the grade map, detector coverage and object blockage, the program proceeds with the computation of coverage factor at respective level of hazardous. Linear programming is used for exhaustive search on the optimum placement. As mentioned in previous stage, combination method is being used to determine the possible combination of placement. Same computation is being employed for all the possible combination of placement to obtain the quantitative data on the hazard coverage. The result is tabulated for the analysis in next stage.

3.2.4.3 Analysis

In this stage, the program gathers the point of interest of all possible placements and identify whether the placement has met the minimum safety requirement. If there is no qualified result, the program will increase the number of detector, back to the generation and image processing stage to implement to same assessment for more number of detectors. Else, it will proceed with the side view assessment. The method of side view assessment is similar to the top view assessment.

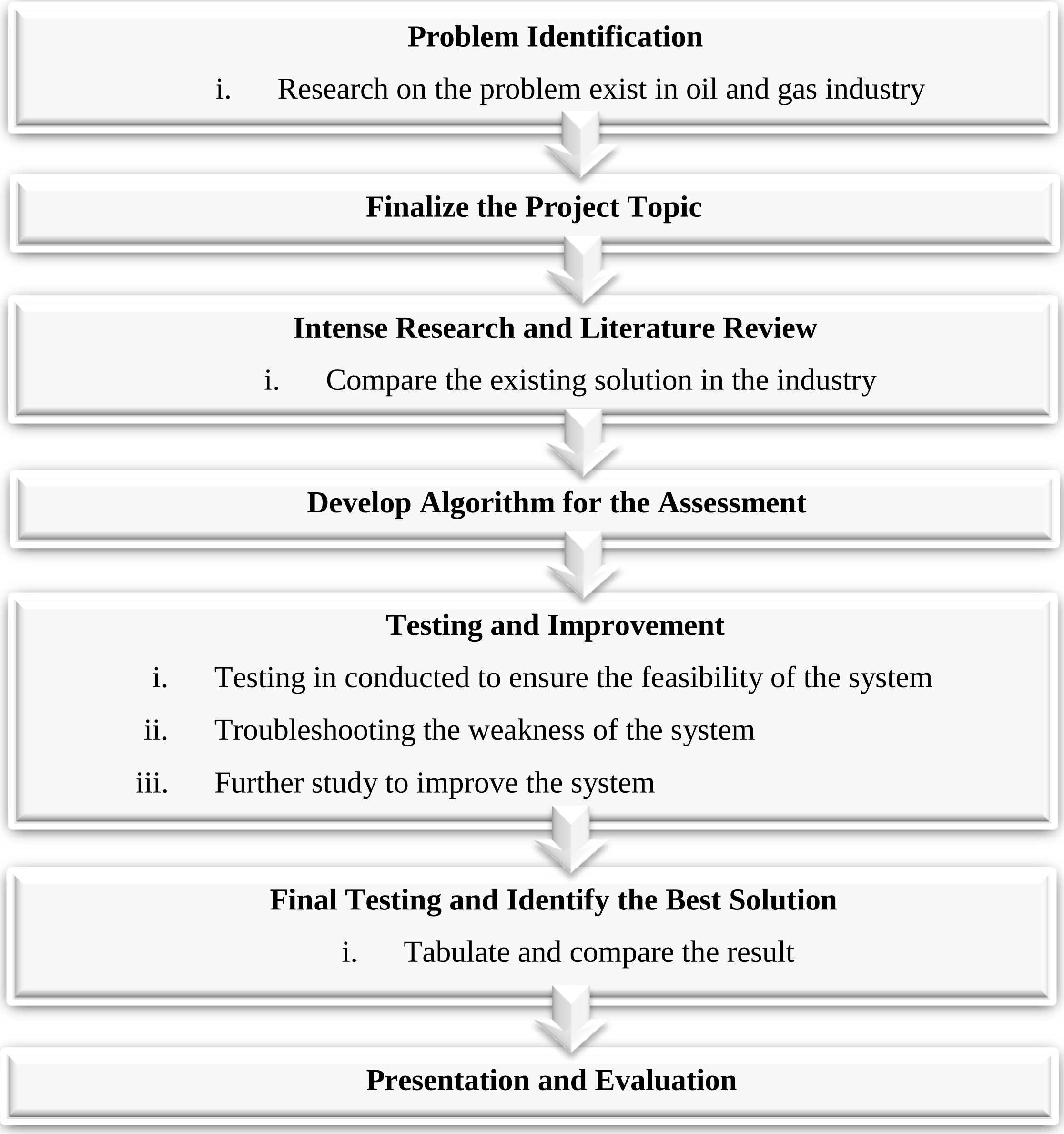
With these, the program able to identify minimum number of detectors required as well as the position of the detector that offer the coverage that meet the minimum safety requirement.

3.1.3 Output of the assessment

As the result of the assessment, the program display all the possible result that have met the safety requirement, including the minimum number of detectors that required, coordination of the recommended location of placement as well as the coverage factor of specific placement. The program is also suggesting the user which would be the best placement of all. The best placement is identified based on the priority of each fire grading, which means the best placement will have highest coverage factor on red region, if both placement have the same coverage factor on red region, the better score will goes to the one have higher factor on orange region.

3.3 Overall Project Flow Chart

In the first stage of the project, a flow chart of the project is being constructed to have a brief idea on whole project flow within the timeline.



Problem Identification

i. Research on the problem exist in oil and gas industry

Finalize the Project Topic

Intense Research and Literature Review

i. Compare the existing solution in the industry

Develop Algorithm for the Assessment

Testing and Improvement

i. Testing in conducted to ensure the feasibility of the system
ii. Troubleshooting the weakness of the system
iii. Further study to improve the system

Final Testing and Identify the Best Solution

i. Tabulate and compare the result

Presentation and Evaluation

Figure 19: Flow of the project.

3.3 Key Milestone

Table 3: Key Milestone.

	Task	Proposed Completed week	Status	Remarks
FYP 1	Finalize the project topic	Week 2	Completed	
	Brief by master student regard to the project topic	Week 3	Completed	Have a brief idea on the fire detection system in actual oil and gas field
	Research on the topic	Week 5	Completed	Literature review and comparing the existing method
	Submission of Extended Proposal	Week 6	Completed	
	Attend Proposal Defense	Week 9	Completed	Obtained the feedback from supervisor and panel for the improvement of the project
	Further research based on the feedback from panel	Week 12	Completed	Modify and improve the literature review
	Interim Report	Week 13	Completed	
FYP 2	Develop the algorithm	Week 7	Completed	
	Complete the progress report	Week 8	Completed	
	Improve the algorithm and collect the new data	Week 10	Completed	Approach the industrial employee to collect the opinion regard to the reliability of the project on the actual environment.
	Pre-SEDEX	Week 11	Completed	
	Submission of draft report	Week 13	Completed	
	Analyse the latest data	Week 13	Completed	
	Identify the recommendation for further study	Week 14	Completed	Discuss with supervisor with the future improvement of the project

3.4 Gantt chart

Refer to appendix A for the Gantt chart of the project.

3.5 Tool

MATLAB R2008a is being used as the main environment to develop the algorithm. It is an ideal tools as it does included image processing tools box that provide strong support for image processing, analysis and visualization.

CHAPTER 4. RESULT AND DISCUSSION

The algorithm of the assessment is included in Appendix B. A top grade map and four side maps were inputted into the program together with the user-defined parameters for the purpose of testing.

i. Angle of detector placement: 35 degree

ii. Angle of field of view: 70 degree

4.1 Top view analysis for one detector

The coverage of a detector from twelve possible positions is generated by the program as shown in Figure 20. The coverage factor at each level of hazardous area is computed and tabulated in Table 4. Figure 16 shows the graph of the tabulated result.

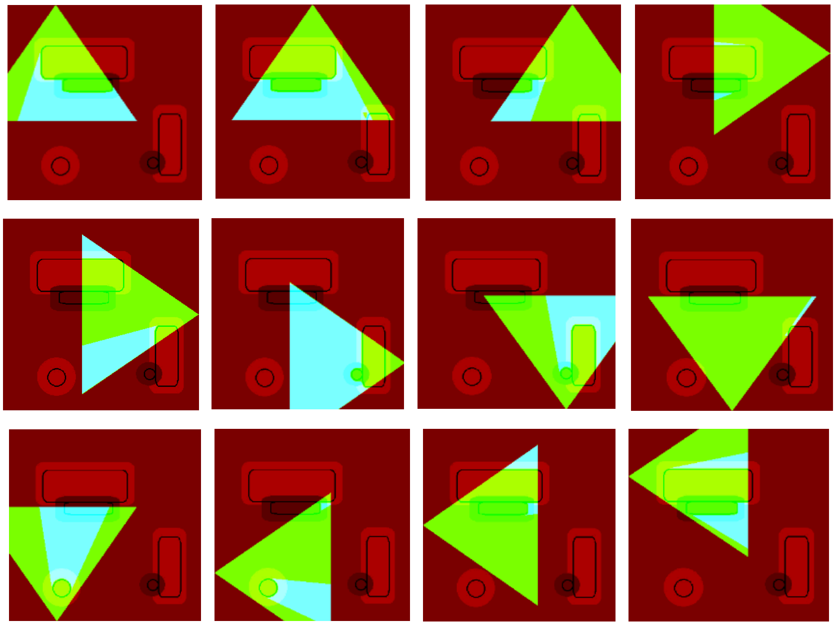


Figure 20: Coverage for one detector placement.

Table 4: Coverage factor at each level of hazardous area for one detector placement.

Position	% on RED	% on ORANGE	% on GREEN
1	44	28	8
2	48	41	6
3	14	26	26
4	57	35	27
5	68	36	16
6	11	14	1
7	73	32	10
8	78	24	35
9	13	15	17
10	24	14	26
11	79	25	28
12	55	41	13

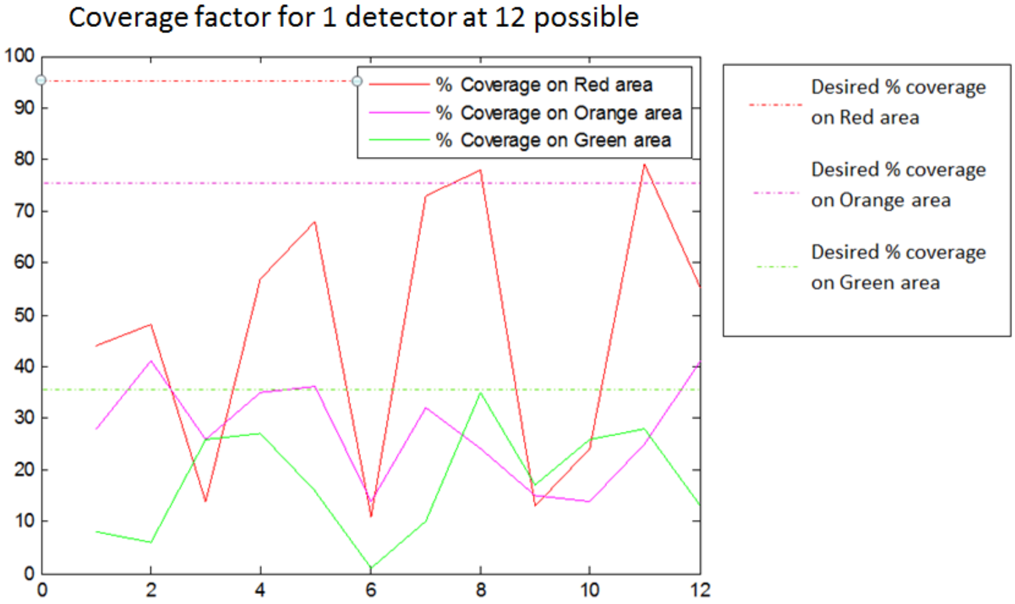
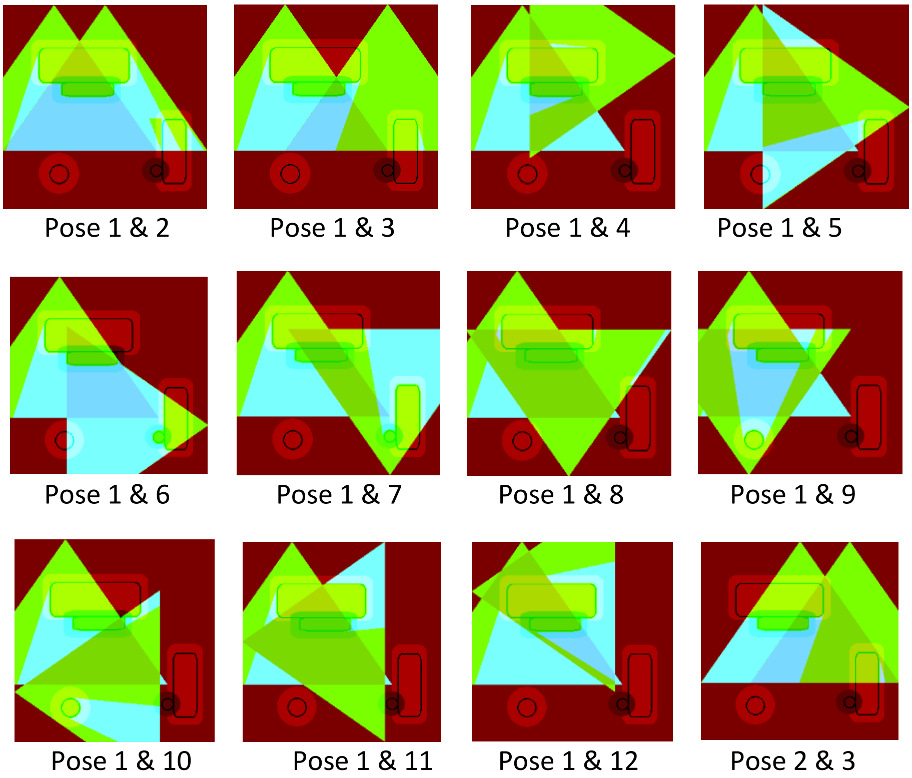


Figure 21: Graph of coverage factor at twelve possible detector placements.

The dotted line in the graph indicated the minimum coverage factor required to achieve the safety requirement, which is 95% for red region, 75% for orange region and 35% for green region. From the graph it can be seen clearly that none of the placement has met the requisite value. Hence the program increases the number of detector in the assessment and proceed with the evaluation on effectiveness of the new possible combination of detector placement.

4.2 Top view analysis for two detectors

Combination method is used to identify the new possible combination of detector placement. Figure 22 shows the coverage of detector at twelve out of sixty-six possible placements for two detectors. The coverage factor on respective fire grading for each possible placement is obtained and tabulated. A graph is plotted in Figure 23 based on the tabulated result. Due to the large number of possibility, Table 5 shows only the coverage factor of twelve possible placements. The complete result has been included in Appendix C.



Pose 1 & 2 Pose 1 & 3 Pose 1 & 4 Pose 1 & 5

Pose 1 & 6 Pose 1 & 7 Pose 1 & 8 Pose 1 & 9

Pose 1 & 10 Pose 1 & 11 Pose 1 & 12 Pose 2 & 3

Figure 22: Coverage of placement for two detectors.

Table 5: Coverage factor for two detectors at twelve out of sixty-six possible placements.

No.	Position 1	Position 2	% of Red	% of Orange	% of Green
1	1	2	48	52	13
2	1	3	48	44	29
3	1	4	59	49	28
4	1	5	63	51	21
5	1	6	54	42	8
6	1	7	77	46	18
7	1	8	71	31	36
8	1	9	52	39	19
9	1	10	58	41	26
10	1	11	74	40	27
11	1	12	55	41	15
12	2	3	48	41	24

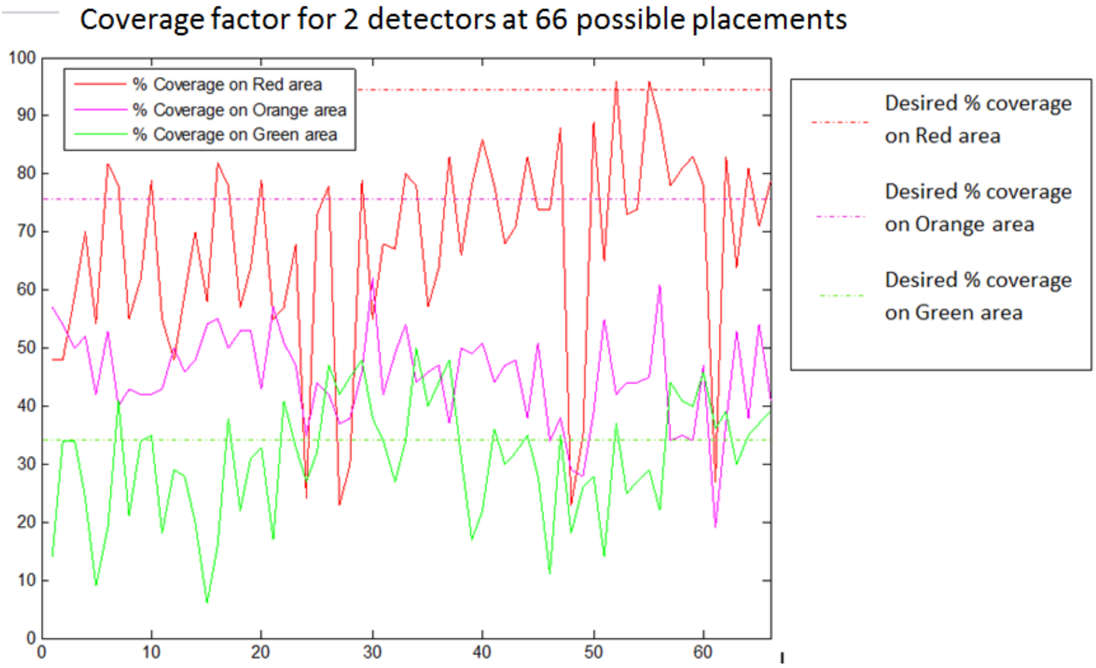
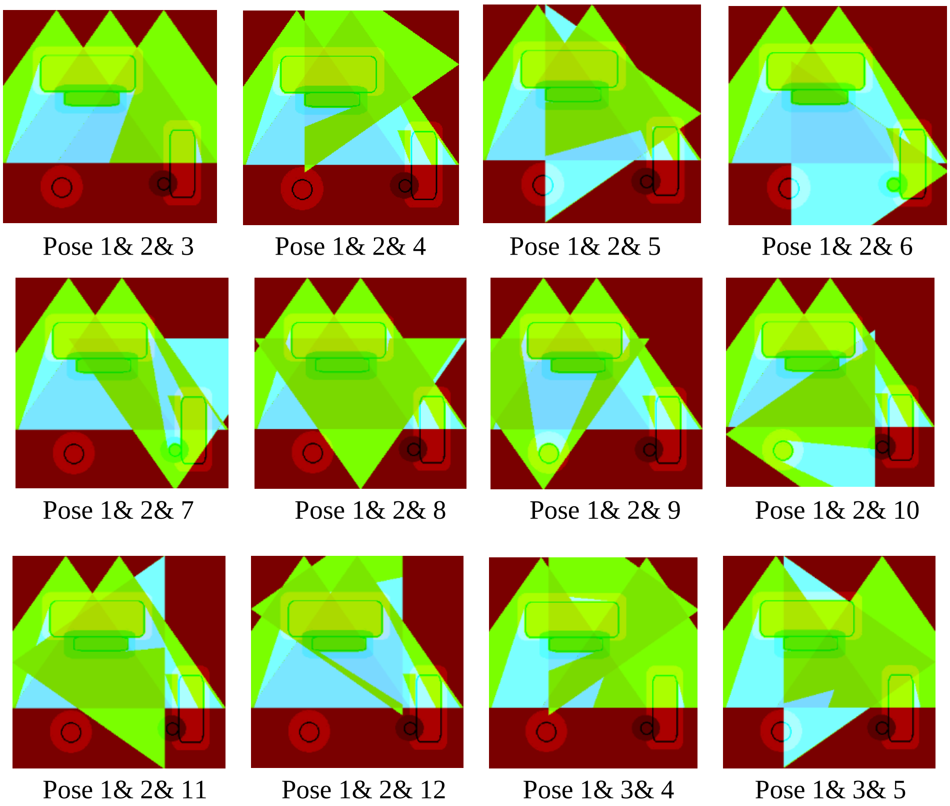


Figure 23: Graph of coverage factor for two detectors at sixty-six possible placements.

According to the graph in Figure 23, there is no combination of detector placement that meets the minimum requirement for coverage factor at orange grading area. Thus, the program increases the number of detector to three and undergoes the same assessment again.

4.3 Top view assessment for three detectors

There are two hundred and twenty-two new combination of placement for three detectors. As the number of combination is a too large, Figure 24 and Table 6 show only the coverage and coverage factor on respective fire grading area of the first twelve combination of the detector placement. The complete result has been included in Appendix D. A graph plotted based on the tabulated result is shown in Figure 25.



Pose 1& 2& 3

Pose 1& 2& 4

Pose 1& 2& 5

Pose 1& 2& 6

Pose 1& 2& 7

Pose 1& 2& 8

Pose 1& 2& 9

Pose 1& 2& 10

Pose 1& 2& 11

Pose 1& 2& 12

Pose 1& 3& 4

Pose 1& 3& 5

Figure 24: Coverage of twelve possible placement for three detectors.

Table 6: Coverage factor for three detectors at twelve out of two hundred and twenty possible placements.

No.	Position 1	Position 2	Position 3	% on Red	% on Orange	% on Green
1	1	2	3	48	67	37
2	1	2	4	59	60	35
3	1	2	5	70	63	28
4	1	2	6	58	70	14
5	1	2	7	82	71	24
6	1	2	8	78	62	45
7	1	2	9	57	70	26
8	1	2	10	64	70	39
9	1	2	11	79	58	41
10	1	2	12	55	59	21
11	1	3	4	59	67	49
12	1	3	5	70	62	41

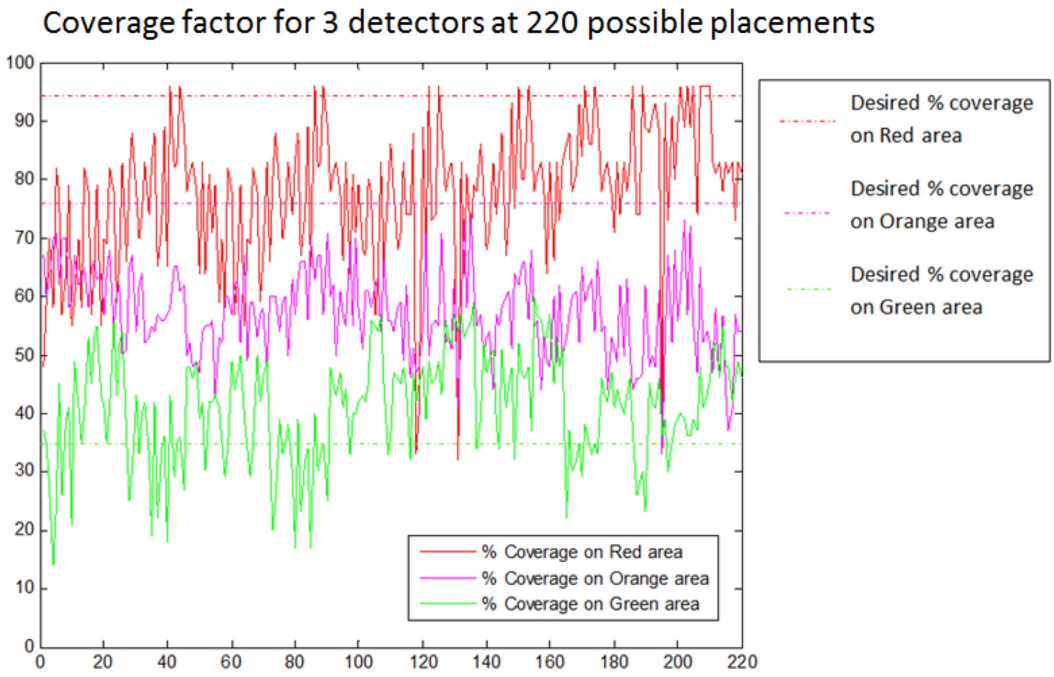
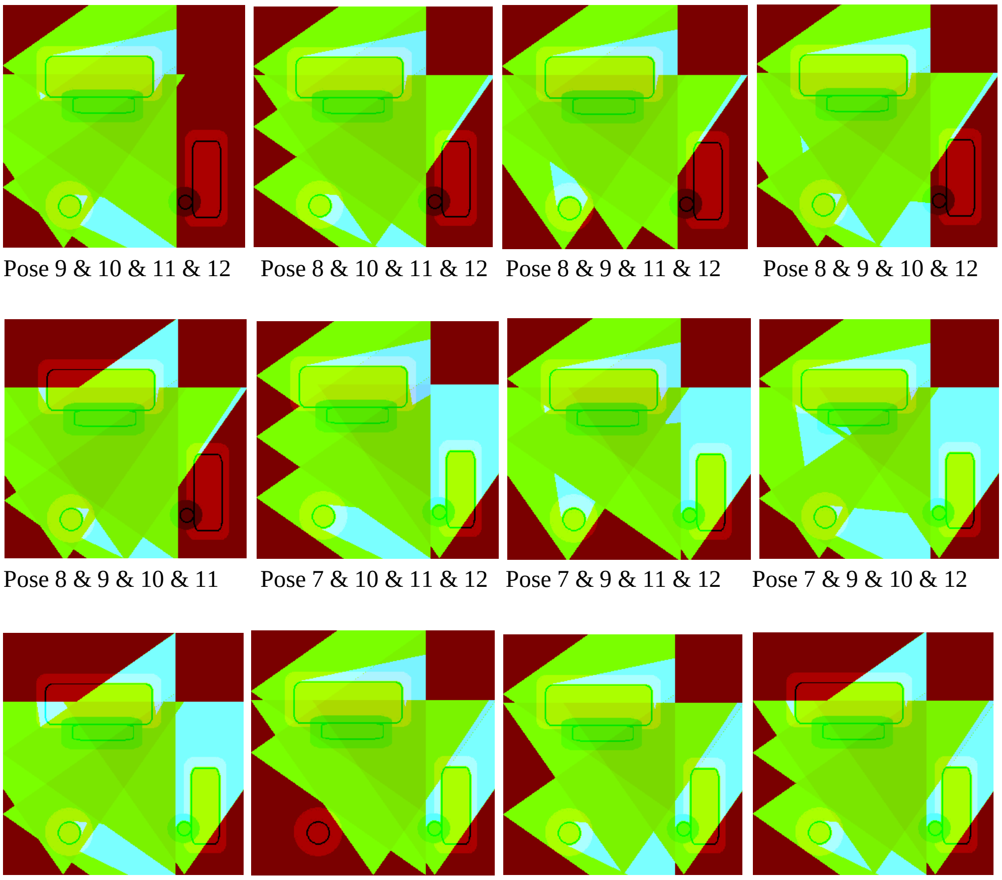


Figure 25: Graph of coverage factor for three detectors at two hundred and twenty possible placements.

However, none coverage factor at orange grading area has met the safety requirement. So the program increases the number of detector again, and implements the similar assessment.

4.4 Top view analysis for four detectors

There are four hundred and ninety-five new combinations of detector placement with four detectors. Since the number of combination is too big, Figure 26 shows only the first twelve combination of the detector placement. The coverage factors of 4 detector placement on respective fire grading area are tabulated in Table 7 and a graph plotted based on the tabulated result is shown in Figure 27. Complete result for four detectors analysis has been included in Appendix E.



Pose 9 & 10 & 11 & 12 Pose 8 & 10 & 11 & 12 Pose 8 & 9 & 11 & 12 Pose 8 & 9 & 10 & 12

Pose 8 & 9 & 10 & 11 Pose 7 & 10 & 11 & 12 Pose 7 & 9 & 11 & 12 Pose 7 & 9 & 10 & 12

Pose 7 & 9 & 10 & 11 Pose 7 & 8 & 11 & 12 Pose 7 & 8 & 10 & 12 Pose 7 & 8 & 10 & 11

Figure 26: Coverage of twelve placements for four detectors.

Table 7: Coverage factor for four detectors at twelve out of four hundred and ninety five possible placements.

No	Position 1	Position 2	Position 3	Position 4	% on Red	% on Orange	% on Green
1	9	10	11	12	83	57	53
2	8	10	11	12	83	59	56
3	8	9	11	12	83	57	58
4	8	9	10	12	81	61	59
5	8	9	10	11	83	48	50
6	7	10	11	12	96	73	47
7	7	9	11	12	96	72	50
8	7	9	10	12	89	76	48
9	7	9	10	11	96	60	43
10	7	8	11	12	96	65	52
11	7	8	10	12	96	77	54
12	7	8	10	11	96	64	47

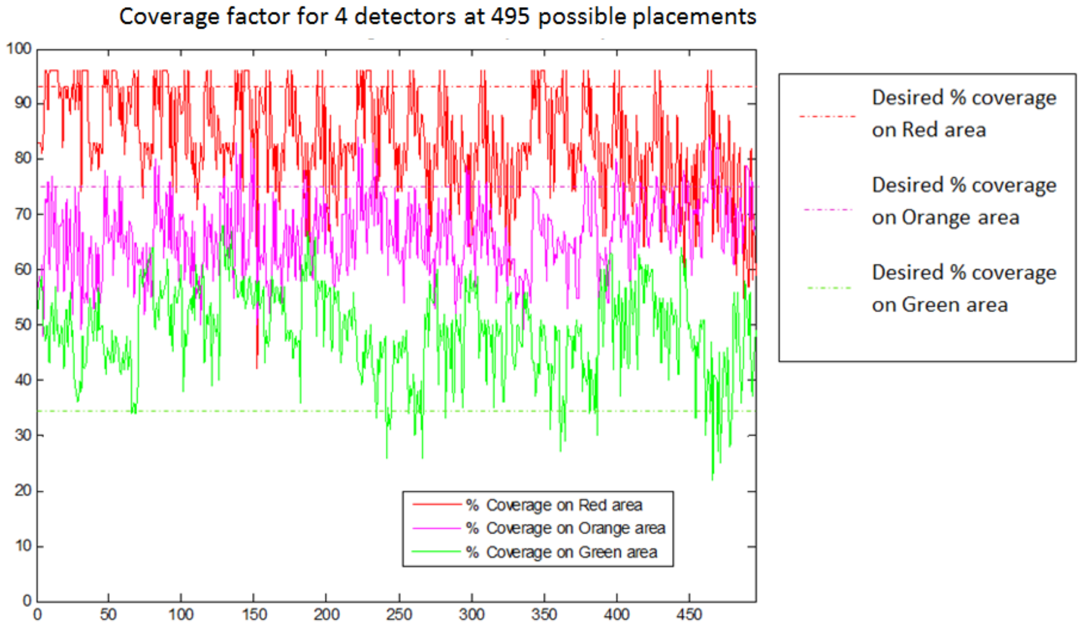


Figure 27: Graph of coverage factor for four detectors at four hundred and ninety five possible placements.

With 4 detectors, the program has found the combination placement of detector that meet the required performance target for the hazard map. As the desired result has found, the program will proceed with the side view analysis of the specific placement.

4.5 Side view analysis

The assessment method on side view analysis is similar to the method being used for top view analysis. The program has generated the side coverage of detector on respective side grade map correspond to the effective placement found from the top view analysis.

As shown in Figure 28, 4 possible placements that meet the safety requirement have been found. The results are displayed in the window, comprising the coverage area on the grade map from top view and side view, the number of detector required, as well as the coverage factor at each fire grading. The best placement that has the greatest point of interest correspond to the priority of each fire grading has suggested by the program in Figure 29. The coverage factor of the four recommended placements is tabulated in Table 9.

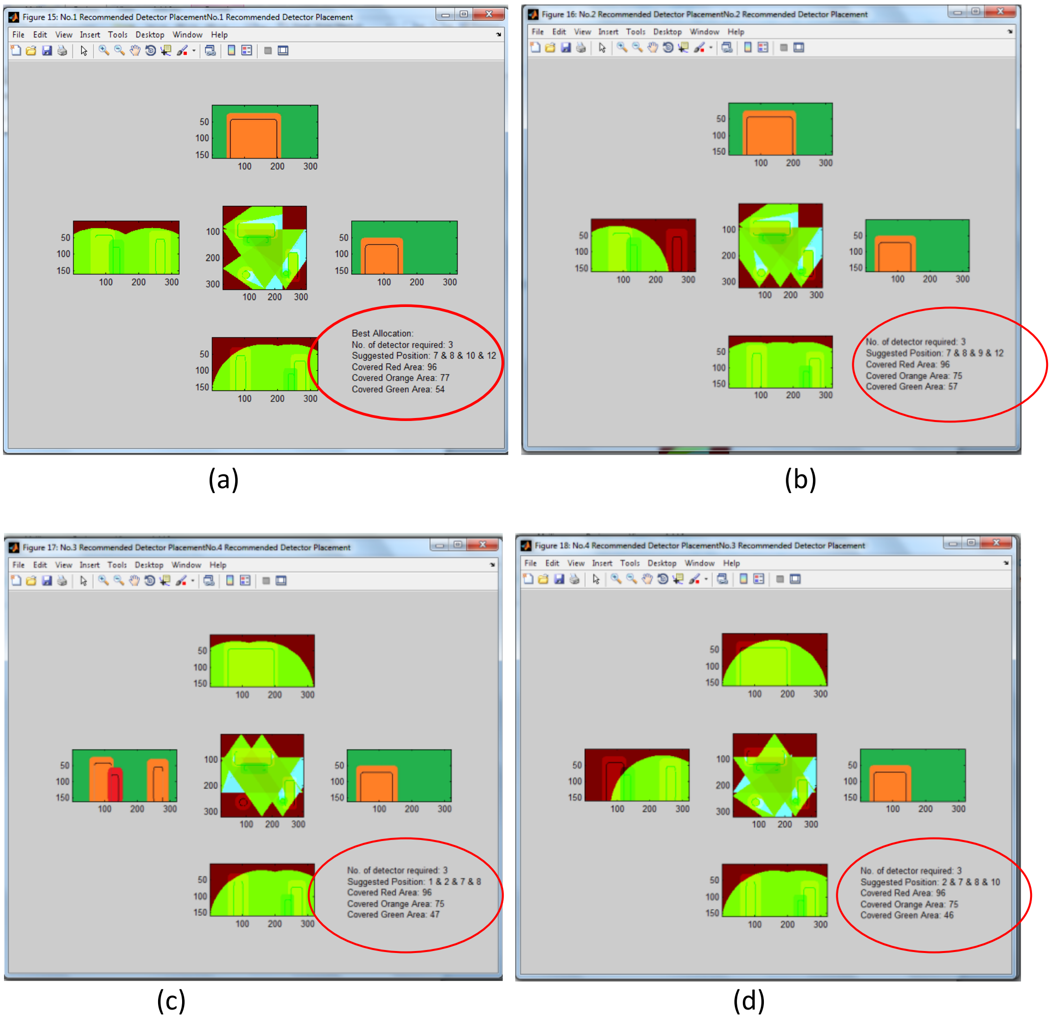


Figure 15: No.1 Recommended Detector PlacementNo.1 Recommended Detector Placement
Figure 16: No.2 Recommended Detector PlacementNo.2 Recommended Detector Placement
Figure 17: No.3 Recommended Detector PlacementNo.4 Recommended Detector Placement
Figure 18: No.4 Recommended Detector PlacementNo.3 Recommended Detector Placement
File Edit View Insert Tools Desktop Window Help
50
100
150
100
200
300

```
Best Allocation:
No. of detector required: 3
Suggested Position: 7 & 8 & 10 & 12
Covered Red Area: 96
Covered Orange Area: 77
Covered Green Area: 54
```

(a)

```
No. of detector required: 3
Suggested Position: 7 & 8 & 9 & 12
Covered Red Area: 96
Covered Orange Area: 75
Covered Green Area: 57
```

(b)

```
No. of detector required: 3
Suggested Position: 1 & 2 & 7 & 8
Covered Red Area: 96
Covered Orange Area: 75
Covered Green Area: 47
```

(c)

```
No. of detector required: 3
Suggested Position: 2 & 7 & 8 & 10
Covered Red Area: 96
Covered Orange Area: 75
Covered Green Area: 46
```

(d)

Figure 28: Output result of the assessmet.

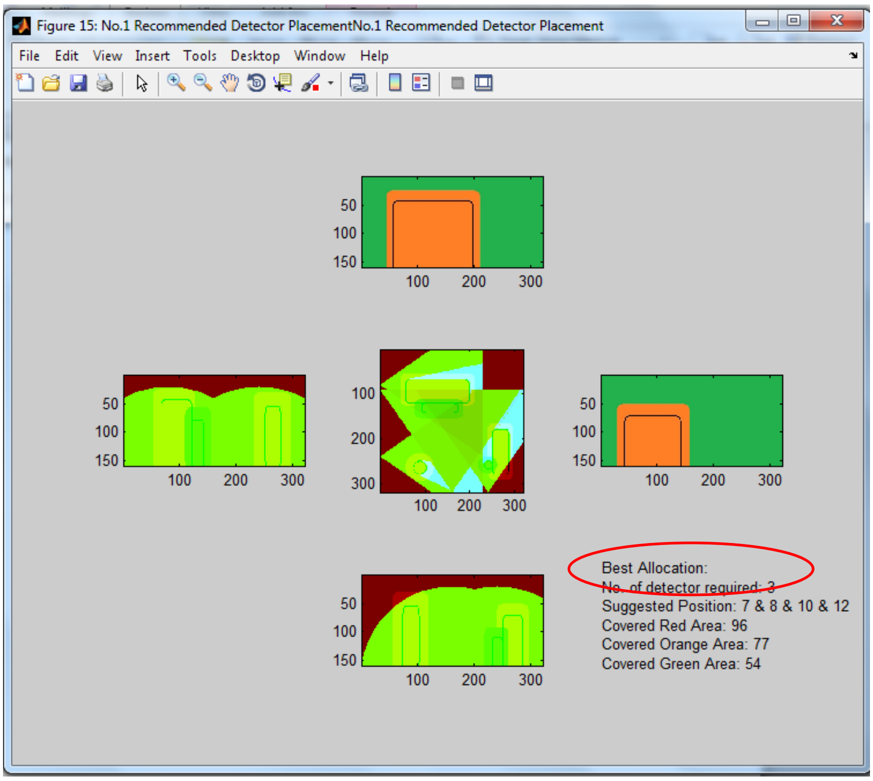


Figure 15: No.1 Recommended Detector PlacementNo.1 Recommended Detector Placement
File Edit View Insert Tools Desktop Window Help
50
100
150
100 200 300
100
200
300
100 200 300

```
Best Allocation:
No. of detector required: 3
Suggested Position: 7 & 8 & 10 & 12
Covered Red Area: 96
Covered Orange Area: 77
Covered Green Area: 54
```

Figure 29: Best Placement of the detectors.

Table 8: Coverage factor of the effective detectors' placement from the top view and side view.

P1	P2	P3	P4	% on Red (from Top)	% on Orange (from Top)	% on Green (from Top)	% on Red (from upper side)	% on Orange (from upper side)	% on Green (from upper side)	% on Red (from Right side)	% on Orange (from Right side)	% on Green (from Right side)	% on Red (from Bottom side)	% on Orange (from Bottom side)	% on Green (from Bottom side)	% on Red (from Left side)	% on Orange (from Left side)	% on Green (from Left side)
7	8	10	12	96	77	54	0	0	0	0	0	0	37	18	18	63	25	16
7	8	9	12	96	75	57	0	0	0	0	0	0	37	19	21	63	13	11
1	2	7	8	96	75	47	0	34	12	0	0	0	37	18	18	0	0	0
2	7	8	10	96	75	46	0	32	10	0	0	0	37	18	18	58	15	10

CHAPTER 5. CONCLUSION

As safety issue is always one of the main concerns in the industry world, effective safety system is required by the industry to ensure the smooth production in the plant. Main objective of this project is to seek for an alternative to improve the safety system in plant in the most economical way.

This paper proposes an effective fire allocation system for petrochemical plant such that the area of coverage achieved the minimum safety requirement with the least number of fire detector. Intense literature review has been done to study the current technique that proposed by the researcher. As the every region within the plant has its own hazardous level, hence priority of the coverage on certain area needs to be taken into account. Visual sensors have a weakness which its coverage will be blocked by any object. In the assessment, the object within the coverage area will be extracted to compute the area of the blockage to increase the effectiveness of the assessment.

In conclusion, this project provides an effective reference for the fire detector placement in petrochemical plant. As safety and economy are high on the agenda throughout the industry, this project able to contribute to the industry by reducing the risk of the plant and achieving cost saving purposes.

5.1 Recommendation

The proposed assessment is in two dimensions that neglect the effect of elevation of the object in the third dimension. Hence for future study, the assessment could be done in 3 dimensions to increase the accuracy of the assessment. Further research could be done on the method to reduce the complexity of the computation as well as the computation time.

REFERENCES

[1] "Fire and Gars Consulting Services," Kenexis, 28 June 2011. [Online]. Available: http://www.kenexis.com/services/FireGas.aspx.

[2] Micropack (Engineering) Ltd, "Flame Detector Selection," 2006.

[3] Rourke and J.O., "Art Gallery Theoroms and Algorithms," in *Oxford University Press*, 1987.

[4] E. Horster and R. Lienhart, "On the Optimal Placement of Multiple Visual Sensors," pp. pp. 111-120, 2006.

[5] E. Becker, G. Guerra-Filho. and F. Makedon, "Automatic Sensor Placement in a 3D volume," in *Proc. Of the Second International Conference on Pervasive Technologies Related to Assistive Environments*, 2009.

[6] S.indu, Chaitanya, Manoj, P. S. Chaudhury and P. A. Bhattacharryya, "Optimal Visual Sensor Placement using Evolutionary Algorithm," in *Delhi College of Enfineering*, 2007.

[7] E. D. Gustavo O., "The Photogrammetric Record," *Development of a practical Photogrammetric Network Design using Evolutionary Computing*, vol. 22, no. 117, pp. 22-38, March 2007.

[8] J. Nievergelt, "Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power," in *In SOFSEM 2000, number 1963 in LNCS*, Springer, 2000, pp. 18-35.

[9] P.Y.Tan, S.A.Vijanth and B.S.Nordin, "Using Multiple View In Optimizing Detector Plcament for Petrochemical Plant," in *IEEE 8th International Colloqium on Signal Processing and its Applications*, 2012.

[10] T. P. Yang, "Optimising Detector Placement and Mapping using Imaging Techniques," in *IEEE 8th International Colloqium on Signal Processing and its Applications*, 2011.

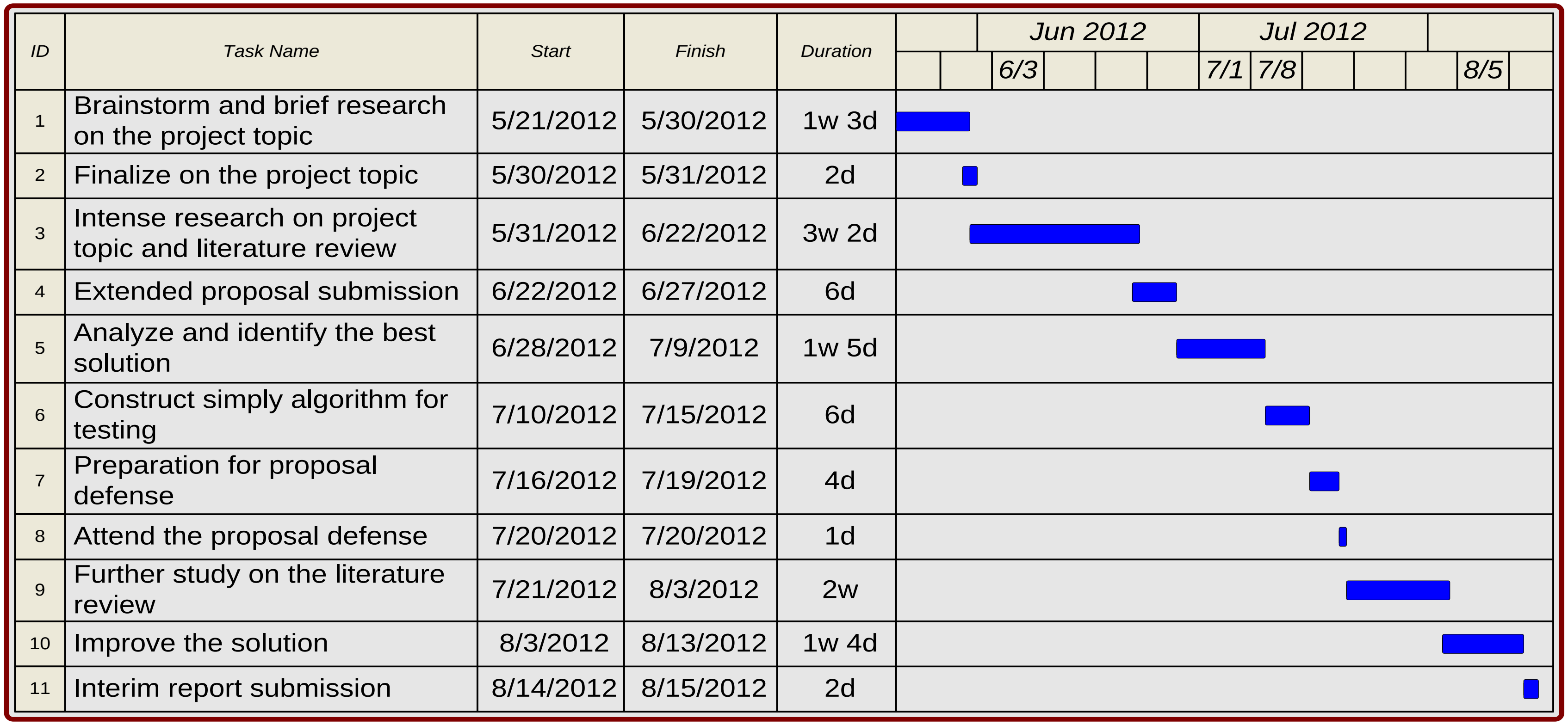
[11] Nadernejad, Ehsan, Sharifzadeh, Sara, Hassanpour and Hamid, "Edge Detection technique: Evaluation and Comparisons," *Applied Mathematical Sciences*, vol. 2, no. 31, pp. 1507-1520, 2008.

[12] MICROPACK (Engineering) Ltd., "Mapping Assessment Guideline," 2008. [Online]. Available: http://www.micropackamericas.com/library/whitepapers/LCN_I1_2008.pdf.

APPENDIX A

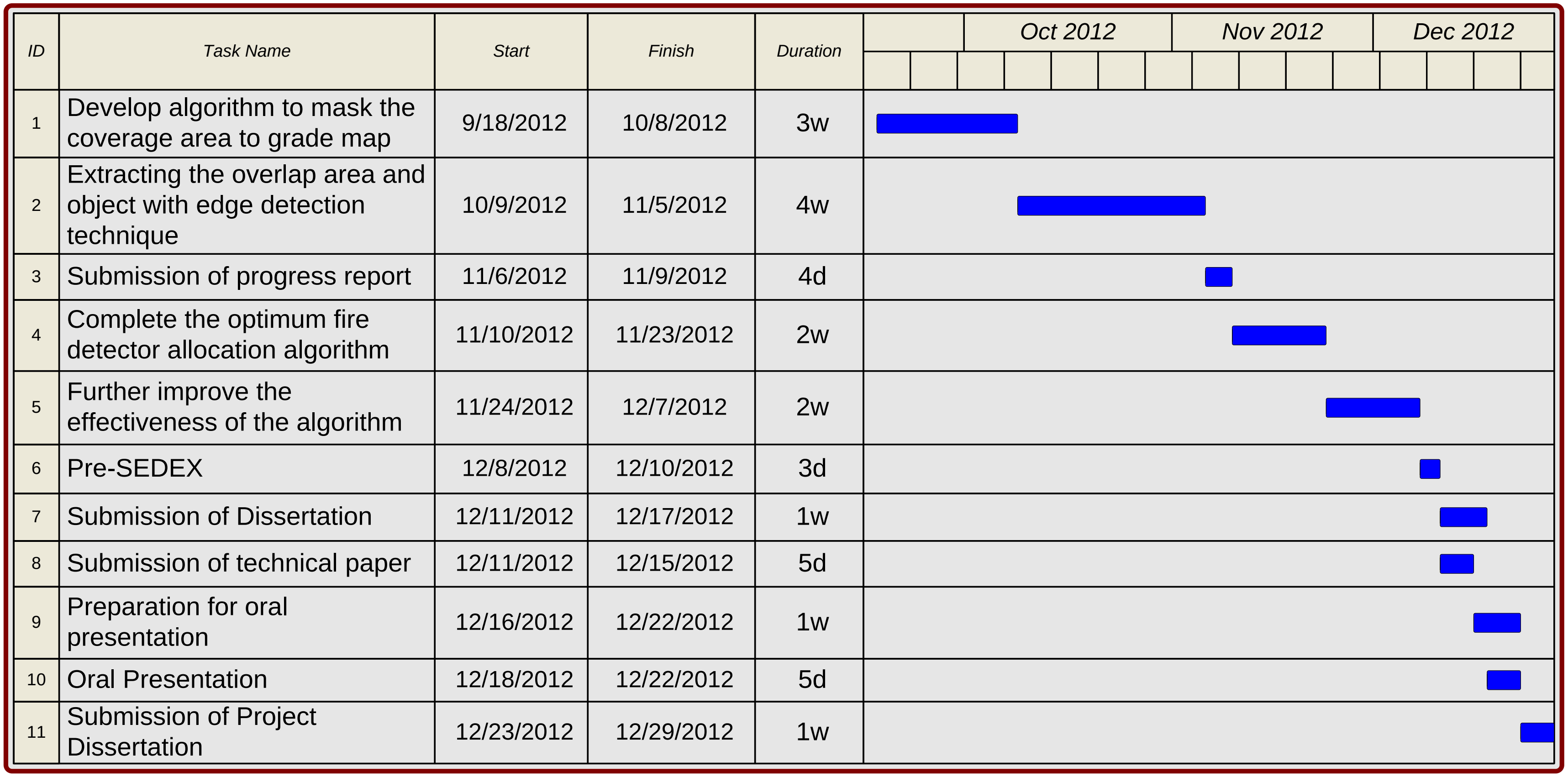
GANNT CHART

A.1 Gannt chart for FYP1



ID	Task Name	Start	Finish	Duration
1	Brainstorm and brief research on the project topic	5/21/2012	5/30/2012	1w 3d
2	Finalize on the project topic	5/30/2012	5/31/2012	2d
3	Intense research on project topic and literature review	5/31/2012	6/22/2012	3w 2d
4	Extended proposal submission	6/22/2012	6/27/2012	6d
5	Analyze and identify the best solution	6/28/2012	7/9/2012	1w 5d
6	Construct simply algorithm for testing	7/10/2012	7/15/2012	6d
7	Preparation for proposal defense	7/16/2012	7/19/2012	4d
8	Attend the proposal defense	7/20/2012	7/20/2012	1d
9	Further study on the literature review	7/21/2012	8/3/2012	2w
10	Improve the solution	8/3/2012	8/13/2012	1w 4d
11	Interim report submission	8/14/2012	8/15/2012	2d

A.2 Gannt chart for FYP2



ID	Task Name	Start	Finish	Duration
1	Develop algorithm to mask the coverage area to grade map	9/18/2012	10/8/2012	3w
2	Extracting the overlap area and object with edge detection technique	10/9/2012	11/5/2012	4w
3	Submission of progress report	11/6/2012	11/9/2012	4d
4	Complete the optimum fire detector allocation algorithm	11/10/2012	11/23/2012	2w
5	Further improve the effectiveness of the algorithm	11/24/2012	12/7/2012	2w
6	Pre-SEDEX	12/8/2012	12/10/2012	3d
7	Submission of Dissertation	12/11/2012	12/17/2012	1w
8	Submission of technical paper	12/11/2012	12/15/2012	5d
9	Preparation for oral presentation	12/16/2012	12/22/2012	1w
10	Oral Presentation	12/18/2012	12/22/2012	5d
11	Submission of Project Dissertation	12/23/2012	12/29/2012	1w

APPENDIX B

ALGORITHM OF THE ASSESSMENT

```
%---------------------MAIN PROGRAM------------------------------------
anglePlacement=input('Enter the angle of detector placement:');
angleDetector=input('Enter the angle of the field of view:');

%Input the grade map
TopGrademap=imread('topgrademap.jpg');
UpGrademap=imread('upcoverage.jpg');
RightGrademap=imread('rightcoverage.jpg');
BottomGrademap=imread('bottomcoverage.jpg');
LeftGrademap=imread('leftcoverage.jpg');

figure(1)
subplot(3,3,2), subimage(UpGrademap)
subplot(3,3,4), subimage(LeftGrademap)
subplot(3,3,5), subimage(TopGrademap)
subplot(3,3,6), subimage(RightGrademap)
subplot(3,3,8), subimage(BottomGrademap)

%Define the map size
topXmax=320;
topYmax=320;
sideXmax=320;
sideYmax=160;

%defination
red=0.35;
green=0.48;
orange=0.67;
SideGreenArea=0;
SideRedArea=0;
SideOrangeArea=0;
TopRedArea=0;
TopOrangeArea=0;
TopGreenArea=0;

RedMin=95;
OrangeMin=75;
GreenMin=35;

angle=angleDetector/2;
distance=sideYmax*tand(90-anglePlacement);

%Convert the grade map to grayscale
[TopGrademapRGB,TopGrademapBW] = imagesegmentation (TopGrademap,
topXmax, topYmax);
[SideGrademapRGB(:,:,:,1),SideGrademapBW(:,:,1)]= imagesegmentation
(UpGrademap, sideXmax, sideYmax);
```

```
[SideGrademapRGB(:,:,:,2),SideGrademapBW(:,:,2)]= imagesegmentation
(RightGrademap, sideXmax, sideYmax);
[SideGrademapRGB(:,:,:,3),SideGrademapBW(:,:,3)]= imagesegmentation
(BottomGrademap, sideXmax, sideYmax);
[SideGrademapRGB(:,:,:,4),SideGrademapBW(:,:,4)]= imagesegmentation
(LeftGrademap, sideXmax, sideYmax);

ResultFlag=0;
n=0;

%Generate top coverage
[TopCov, NumTopPosition,ReferencePoint]
=topcoverage(angle,distance,topXmax,topYmax);

%Generate side coverage
[SideCov, NumSidePosition]=sidecoverage(angle,distance);

%Generate blockage
[TopBlockageArea]= blockage(TopGrademapBW, TopCov, NumTopPosition,
ReferencePoint, topXmax,topYmax);

%compute the total pixel frm Top grademap
 for x = 1:topXmax
     for y = 1:topYmax
         if (TopGrademapBW(y,x) == red);
             TopRedArea=TopRedArea+1;
         end
         if (TopGrademapBW(y,x) == orange);
             TopOrangeArea=TopOrangeArea+1;
         end
         if (TopGrademapBW(y,x) == green);
             TopGreenArea=TopGreenArea+1;
         end
     end
 end

%compute the total pixel for side grademap
for i=1:4
    for x = 1:sideXmax
        for y = 1:sideYmax
            if (SideGrademapBW(y,x,i) == red);
                SideRedArea=SideRedArea+1;
            end
            if (SideGrademapBW(y,x,i) == orange);
                SideOrangeArea=SideOrangeArea+1;
            end
            if (SideGrademapBW(y,x,i) == green);
                SideGreenArea=SideGreenArea+1;
            end
        end
    end

end
TopGradeArea = [TopRedArea TopOrangeArea TopGreenArea];
SideGradeArea = [SideRedArea SideOrangeArea SideGreenArea];
```

```
%----------------------- 1 detector-----------------------------------
NumDetector=1;

 fprintf(1, '\nTOP view coverage with %d detector \n',NumDetector);
 fprintf(1, 'PointOne  RED  ORANGE  GREEN \n');

n=0; %number of good result
for j=1:NumTopPosition

    %compute the percentage of coverage for top view
    [TopIMout,TOPredcov(j), TOPorangecov(j),
TOPgreencov(j)]=ComputeTopCoverage(TopGrademapBW,TopCov(:,:,j),TopBlock
ageArea(:,:,j),TopGradeArea,topXmax,topYmax);

    TopCovResult(j,:)=[j TOPredcov(j) TOPorangecov(j) TOPgreencov(j)];

    %check if there is any result that met the requirement
    %find the side coverage if there is any
    if (TOPredcov(j)>=RedMin && TOPorangecov(j)>=OrangeMin &&
TOPgreencov(j)>=GreenMin)
        n=n+1;
        TResult(:,:,:,n)=TopIMout;
        [Frame,SPosition]=SidePosition(j);

        SResult(:,:,:,1,n)=SideGrademapRGB(:,:,:,1);
        SResult(:,:,:,2,n)=SideGrademapRGB(:,:,:,2);
        SResult(:,:,:,3,n)=SideGrademapRGB(:,:,:,3);
        SResult(:,:,:,4,n)=SideGrademapRGB(:,:,:,4);

        SMap(:,:,1)=SideGrademapBW(:,:,Frame);
        SCov(:,:,1)=SideCov(:,:,SPosition);
        Loop=1;

[SideIMout,SIDEredcov,SIDEorangecov,SIDEgreencov]=ComputeCoverage(SMap,
SCov,Loop,SideGradeArea,sideXmax,sideYmax);
        SResult(:,:,:,Frame,n)=im2uint8(SideIMout(:,:,:,1));

        OverallResult(n,:)=[j TOPredcov(j) TOPorangecov(j)
TOPgreencov(j) SIDEredcov SIDEorangecov SIDEgreencov n];
        ResultFlag=1;
    end
end

    disp(TopCovResult)

    a=1:NumTopPosition;
    figure(30)
    plot(a,TOPredcov(a),'r'),    hold on
    plot(a,TOPorangecov(a),'m')
    plot(a,TOPgreencov(a),'g'),    hold off
    axis([0 NumTopPosition 0 100])
    title('% of detector coverage from 12 possible point');
    h = get(gca, 'title');
    set(h, 'FontSize', 15)
    legend('% Coverage on Red area','% Coverage on Orange area','%
Coverage on Green area')

if ResultFlag==1
    fprintf('\n');
    fprintf('Obtained %d possible placement that met safety
requirement!\n\n',n);
    fprintf('Result being sort in descending order:\n');
    fprintf(1, 'Position  R(T)  O(T)  G(T)  R(S)  O(S)  G(S) \n');
    Rearrange=sortrows(OverallResult,[-3 -4]);
    disp(Rearrange)

    for j=1:n
            Position=Rearrange(j,8);
            figure('name',sprintf('No.%d Recommended Detector
Placement',j));
            subplot(3,3,2), subimage(SResult(:,:,:,1,Position))
            subplot(3,3,4), subimage(SResult(:,:,:,4,Position))
            subplot(3,3,5), subimage(TResult(:,:,:,Position))
            subplot(3,3,6), subimage(SResult(:,:,:,2,Position))
            subplot(3,3,8), subimage(SResult(:,:,:,3,Position))
            subplot(3,3,9)
                if j==1
                text(.05,1.3,['Best Allocation:'])
                end
                text(.05,1.1,['No. of detector required: 1'])
                s=sprintf('Suggested Position: %d',Position);
                text(.05,0.9,s)
                s=sprintf('Covered Red Area: %d',Rearrange(j,2));
text(.05,0.7,s)
                s=sprintf('Covered Green Area: %d',Rearrange(j,3));
text(.05,0.5,s)
                s=sprintf('Covered Orange Area: %d',Rearrange(j,4));
text(.05,0.3,s)
                axis([0 5 0 1.5])
                axis off
    end
    return
end

 %--------------------------------2 detector----------------------------
--
TOPredcov=0;
TOPgreencov=0;
TOPorangecov=0;

NumDetector=2;
TwoCombi=combnk([1:NumTopPosition],NumDetector);
NumTopCov=length(TwoCombi);

fprintf(1, 'TOP view coverage with %d detector \n',NumDetector);
fprintf(1, '    PointOne PointTwo RED  ORANGE  GREEN \n');


for j=1:NumTopCov
```

```
    [DuoTopCov] = MultipleTopCoverage(TopCov, TwoCombi(j,:),
NumDetector, topXmax, topYmax); %generate top coverage

[DuoTopBlockage]=MultipleTopBlockage(DuoTopCov,TopBlockageArea,TwoCombi
(j,:),NumDetector,topXmax,topYmax);
    TPositionOne=TwoCombi(j,1);
    TPositionTwo=TwoCombi(j,2);

    %compute the POC for top view
    [TopIMout,TOPredcov(j), TOPorangecov(j),
TOPgreencov(j)]=ComputeTopCoverage(TopGrademapBW,DuoTopCov,DuoTopBlocka
ge,TopGradeArea,topXmax,topYmax);

    TopTwoCovResult(j,:)=[j TPositionOne TPositionTwo TOPredcov(j)
TOPorangecov(j) TOPgreencov(j)];

    %check if there is any result that met the requirement
    %find the side coverage if there is any
    if (TOPredcov(j)>=RedMin && TOPorangecov(j)>=OrangeMin &&
TOPgreencov(j)>=GreenMin)
        n=n+1;
        TResult(:,:,:,n)=TopIMout;

        [FrameOne,SPositionOne]= SidePosition (TPositionOne);
        [FrameTwo,SPositionTwo]= SidePosition (TPositionTwo);

        SResult(:,:,:,1,n)=SideGrademapRGB(:,:,:,1);
        SResult(:,:,:,2,n)=SideGrademapRGB(:,:,:,2);
        SResult(:,:,:,3,n)=SideGrademapRGB(:,:,:,3);
        SResult(:,:,:,4,n)=SideGrademapRGB(:,:,:,4);

        if FrameOne==FrameTwo
            SPosition(1)=SPositionOne;
            SPosition(2)=SPositionTwo;
            NumCov=2;
            [SCov]=MultipleSideCoverage(SideCov, SPosition, NumCov,
sideXmax, sideYmax);
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);

            Loop=1;
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,1));

        else
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);
            SMap(:,:,2)=SideGrademapBW(:,:,FrameTwo);
            SCov(:,:,1)=SideCov(:,:,SPositionOne);
            SCov(:,:,2)=SideCov(:,:,SPositionTwo);

            Loop=2;
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,1));
            SResult(:,:,:,FrameTwo,n)= im2uint8(SideIMout(:,:,:,2));
        end
        OverallResult(n,:)=[j TPositionOne TPositionTwo TOPredcov(j)
TOPorangecov(j) TOPgreencov(j) SIDEredcov SIDEorangecov SIDEgreencov n];
        ResultFlag=1;
    end



end
disp(TopTwoCovResult)
    a=1:NumTopCov;
    figure(31)
    plot(a,TOPredcov(a),'r'),    hold on
    plot(a,TOPorangecov(a),'m')
    plot(a,TOPgreencov(a),'g'),    hold off
    axis([0 NumTopCov 0 100])
    title('% of detector coverage from 66 possible point');
    h = get(gca, 'title');
    set(h, 'FontSize', 15)
    legend('% Coverage on Red area','% Coverage on Orange area','%
Coverage on Green area')

if ResultFlag==1
    fprintf('\n');
    fprintf('Obtained %d possible placement that met safety
requirement!\n\n',n);
    fprintf('Result being sort in descending order:\n');
    fprintf(1, '\tNo. PointOne PointTwo  R(T)  O(T)  G(T)  R(S)  O(S)
G(S) \n');
    Rearrange=sortrows(OverallResult,[-5 -6]);
    disp(Rearrange)

    for j=1:n
            Position=Rearrange(j,10);
            PointOne=Rearrange(j,2);
            PointTwo=Rearrange(j,3);
            figure('name',sprintf('No.%d Recommended Detector
Placement',j));
            subplot(3,3,2), subimage(SResult(:,:,:,1,Position))
            subplot(3,3,4), subimage(SResult(:,:,:,4,Position))
            subplot(3,3,5), subimage(TResult(:,:,:,Position))
            subplot(3,3,6), subimage(SResult(:,:,:,2,Position))
            subplot(3,3,8), subimage(SResult(:,:,:,3,Position))
            subplot(3,3,9)
                if j==1
                text(.05,1.3,['Best Allocation:'])
                end
                text(.05,1.1,['No. of detector required: 2'])
                text(.05,0.9,sprintf('Suggested Position: %d & %d
& %d',PointOne,PointTwo,PointThree))
                text(.05,0.7,sprintf('Covered Red
Area: %d',Rearrange(j,5)))
```

```
                text(.05,0.5,sprintf('Covered Orange 
Area: %d',Rearrange(j,6)))
                text(.05,0.3,sprintf('Covered Green 
Area: %d',Rearrange(j,7)))
                axis([0 5 0 1.5])
                axis off
    end
    return
end
%----------------------3 Detectors-------------------------------------
----
TOPredcov=0;
TOPgreencov=0;
TOPorangecov=0;
NumTopCov=0;

NumDetector=3;
ThreeCombi=combnk([1:NumTopPosition],NumDetector);
NumTopCov=length(ThreeCombi); %220 for 3 detector

fprintf(1, 'TOP view coverage with %d detector \n',NumDetector);
fprintf(1, '    PointOne PointTwo PointThree RED ORANGE  GREEN \n');

for j=1:NumTopCov

    [TriTopCov] = MultipleTopCoverage(TopCov, ThreeCombi(j,:), 
NumDetector, topXmax, topYmax);

[TriTopBlockage]=MultipleTopBlockage(TriTopCov,TopBlockageArea,ThreeCom
bi(j,:),NumDetector,topXmax,topYmax);
    TPositionOne=ThreeCombi(j,1); %first position
    TPositionTwo=ThreeCombi(j,2); %second position
    TPositionThree=ThreeCombi(j,3); %third position

    %compute the POC for top view
    [TopIMout,TOPredcov(j), TOPorangecov(j), 
TOPgreencov(j)]=ComputeTopCoverage(TopGrademapBW,TriTopCov,TriTopBlocka
ge,TopGradeArea,topXmax,topYmax);

    TopThreeCovResult(j,:)=[j TPositionOne TPositionTwo TPositionThree 
TOPredcov(j) TOPorangecov(j) TOPgreencov(j)];

    %check if there is any result that met the requirement
    %find the side coverage if there is any
    if (TOPredcov(j)>=RedMin && TOPorangecov(j)>=OrangeMin && 
TOPgreencov(j)>=GreenMin)

        n=n+1;
        TResult(:,:,:,n)=TopIMout;

        [FrameOne,SPositionOne]= SidePosition (TPositionOne);
        [FrameTwo,SPositionTwo]= SidePosition (TPositionTwo);
        [FrameThree,SPositionThree]= SidePosition (TPositionThree);

        SResult(:,:,:,1,n)=SideGrademapRGB(:,:,:,1);
        SResult(:,:,:,2,n)=SideGrademapRGB(:,:,:,2);
        SResult(:,:,:,3,n)=SideGrademapRGB(:,:,:,3);
        SResult(:,:,:,4,n)=SideGrademapRGB(:,:,:,4);

        %check if three location is located at the same side
        if ((FrameOne==FrameTwo)&&(FrameOne==FrameThree))
            NumCov=3; %three coverage at same side
            [SCov]=MultipleSideCoverage(SideCov, SPosition, NumCov,
sideXmax, sideYmax);
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);

            Loop=1; %one side map is involved
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameOne,n)=
im2uint8(SideIMout(:,:,:,1)); %replace the ori map with analysed map

        elseif ((FrameOne==FrameTwo)&& (FrameOne~=FrameThree))
            SPosition(1)=SPositionOne;
            SPosition(2)=SPositionTwo;

            %coverage at frame one
            NumCov=2; %two coverage at same side
            [SCov(:,:,1)]=MultipleSideCoverage(SideCov, SPosition,
NumCov, sideXmax, sideYmax);
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);

            %coverage at frame three
            SCov(:,:,2)=SideCov(:,:,SPositionThree);
            SMap(:,:,2)=SideGrademapBW(:,:,FrameThree);

            Loop=2; %two side map is involved
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,1));
            SResult(:,:,:,FrameThree,n)= im2uint8(SideIMout(:,:,:,2));

        elseif ((FrameOne==FrameThree)&& (FraneOne~=FrameTwo))
            SPosition(1)=SPositionOne;
            SPosition(2)=SPositionThree;

            NumCov=2;
            [SCov(:,:,1)]=MultipleSideCoverage(SideCov, SPosition,
NumCov, sideXmax, sideYmax);
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);

            SCov(:,:,2)=SideCov(:,:,SPositionTwo);
            SMap(:,:,2)=SideGrademapBW(:,:,FrameTwo);

            Loop=2; %two side map is involved
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
```

```
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,1));
            SResult(:,:,:,FrameTwo,n)= im2uint8(SideIMout(:,:,:,2));

        elseif ((FrameTwo==FrameThree)&& (FrameTwo~=FrameOne))
            SPosition(1)=SPositionTwo;
            SPosition(2)=SPositionThree;
            NumCov=2;
            [SCov(:,:,1)]=MultipleSideCoverage(SideCov, SPosition, 
NumCov, sideXmax, sideYmax);
            SCov(:,:,2)=SideCov(:,:,SPositionOne);
            SMap(:,:,1)=SideGrademapBW(:,:,FrameTwo);
            SMap(:,:,2)=SideGrademapBW(:,:,FrameOne);

            Loop=2; %two side map is involved
            [SideIMout, SIDEredcov, SIDEorangecov, 
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameTwo,n)= im2uint8(SideIMout(:,:,:,1));
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,2));

        else
            SMap(:,:,1)=SideGrademapBW(:,:,FrameOne);
            SCov(:,:,1)=SideCov(:,:,SPositionOne);

            SMap(:,:,2)=SideGrademapBW(:,:,FrameTwo);
            SCov(:,:,2)=SideCov(:,:,SPositionTwo);

            SMap(:,:,3)=SideGrademapBW(:,:,FrameThree);
            SCov(:,:,3)=SideCov(:,:,SPositionThree);

            Loop=3;
            [SideIMout, SIDEredcov, SIDEorangecov, 
SIDEgreencov]=ComputeCoverage(SMap,SCov,Loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,FrameOne,n)= im2uint8(SideIMout(:,:,:,1));
            SResult(:,:,:,FrameTwo,n)= im2uint8(SideIMout(:,:,:,2));
            SResult(:,:,:,FrameThree,n)= im2uint8(SideIMout(:,:,:,3));
        end

        %result that met the safety requirement
        FinalResult(n,:)=[j TPositionOne TPositionTwo TPositionThree 
TOPredcov(j) TOPorangecov(j) TOPgreencov(j) SIDEredcov SIDEorangecov 
SIDEgreencov n];
        ResultFlag=1;
    end
end

Rearrange=sortrows(TopThreeCovResult,[-5 -6 -7]);
disp(TopThreeCovResult)

    a=1:NumTopCov;
    figure(32)
    plot(a,TOPredcov(a),'r'),    hold on
    plot(a,TOPorangecov(a),'m')
    plot(a,TOPgreencov(a),'g'),    hold off
    axis([0 NumTopCov 0 100])
    title('% of detector coverage from 220 possible point');
    h = get(gca, 'title');
    set(h, 'FontSize', 15)
    legend('% Coverage on Red area','% Coverage on Orange area','%
Coverage on Green area')

%if there is result met the requirement
 if ResultFlag==1
    fprintf('\n');
    fprintf('Obtained %d possible placement that met safety
requirement!\n\n',n);
    fprintf('Result being sort in descending order:\n');
    fprintf(1, '\tNo.    P1    P2   P3   R(T)   O(T)   G(T)   R(S)
O(S)   G(S) \n');

    Rearrange=sortrows(FinalResult,[-5 -6 -7]); %sort the table
according to the % of orange cov (as red cov is the max 100%)
    disp(Rearrange)

    for j=1:n
            Position=Rearrange(j,11); %out of 220 possibility

            PointOne=Rearrange(j,2);
            PointTwo=Rearrange(j,3);
            PointThree=Rearrange(j,4);

            figure('name',sprintf('No.%d Recommended Detector
Placement',j,Position));
            subplot(3,3,2), subimage(SResult(:,:,:,1,Position))
            subplot(3,3,4), subimage(SResult(:,:,:,4,Position))
            subplot(3,3,5), subimage(TResult(:,:,:,Position))
            subplot(3,3,6), subimage(SResult(:,:,:,2,Position))
            subplot(3,3,8), subimage(SResult(:,:,:,3,Position))

            %display the result in text form
            subplot(3,3,9)
                if j==1  %top result in table
                text(.05,1.3,['Best Allocation:'])
                end
                text(.05,1.1,['No. of detector required: 3'])
                text(.05,0.9,sprintf('Suggested Position: %d & %d
& %d',PointOne,PointTwo,PointThree))
                text(.05,0.7,sprintf('Covered Red
Area: %d',Rearrange(j,5)))
                text(.05,0.5,sprintf('Covered Orange
Area: %d',Rearrange(j,6)))
                text(.05,0.3,sprintf('Covered Green
Area: %d',Rearrange(j,7)))
                axis([0 5 0 1.5])
                axis off
    end
    return
 end
 %----------------------4 detectors------------------------------
TOPredcov=0;
```

```
TOPgreencov=0;
TOPorangecov=0;
NumTopCov=0;

NumDetector=4;
FourCombi=combnk([1:NumTopPosition],NumDetector);
NumTopCov=length(FourCombi); %220 for 3 detector

fprintf(1, 'TOP view coverage with %d detector \n',NumDetector);
fprintf(1, '    POne PTwo PThree PFour RED ORANGE  GREEN \n');



for j=1:NumTopCov

    [FourTopCov] = MultipleTopCoverage(TopCov, FourCombi(j,:),
NumDetector, topXmax, topYmax);

[FourTopBlockage]=MultipleTopBlockage(FourTopCov,TopBlockageArea,FourCo
mbi(j,:),NumDetector,topXmax,topYmax);
    TPositionOne=FourCombi(j,1); %first position
    TPositionTwo=FourCombi(j,2); %second position
    TPositionThree=FourCombi(j,3); %third position
    TPositionFour=FourCombi(j,4);

    %compute the POC for top view
    [TopIMout,TOPredcov(j), TOPorangecov(j),
TOPgreencov(j)]=ComputeTopCoverage(TopGrademapBW,FourTopCov,FourTopBloc
kage,TopGradeArea,topXmax,topYmax);

    TopFourCovResult(j,:)=[j TPositionOne TPositionTwo TPositionThree
TPositionFour TOPredcov(j) TOPorangecov(j) TOPgreencov(j)];

    %check if there is any result that met the requirement
    %find the side coverage if there is any
    if (TOPredcov(j)>=RedMin && TOPorangecov(j)>=OrangeMin &&
TOPgreencov(j)>=GreenMin)
        n=n+1;
        TResult(:,:,:,n)=TopIMout;
        [Frame(1),SPosition(1)]= SidePosition (TPositionOne);
        [Frame(2),SPosition(2)]= SidePosition (TPositionTwo);
        [Frame(3),SPosition(3)]= SidePosition (TPositionThree);
        [Frame(4),SPosition(4)]= SidePosition (TPositionFour);

        SResult(:,:,:,1,n)=SideGrademapRGB(:,:,:,1);
        SResult(:,:,:,2,n)=SideGrademapRGB(:,:,:,2);
        SResult(:,:,:,3,n)=SideGrademapRGB(:,:,:,3);
        SResult(:,:,:,4,n)=SideGrademapRGB(:,:,:,4);

        FrameOneFlag=0;
        FrameTwoFlag=0;
        FrameThreeFlag=0;
        FrameFourFlag=0;
        FrameOneIndex = find(Frame==1);
        FrameTwoIndex = find(Frame==2);
        FrameThreeIndex = find(Frame==3);
        FrameFourIndex = find(Frame==4);

        if (isempty(FrameOneIndex) == 0), FrameOneFlag=1; end
        if (isempty(FrameTwoIndex) == 0), FrameTwoFlag=1; end
        if (isempty(FrameThreeIndex) == 0), FrameThreeFlag=1; end
        if (isempty(FrameFourIndex) == 0), FrameFourFlag=1; end

        loop=1;

        if (FrameOneFlag == 1)
            SMap=SideGrademapBW(:,:,1);
            NumCov=length(FrameOneIndex);
            for i=1:NumCov,
SPositionFOne(i)=SPosition(FrameOneIndex(i)); end
            [SCov]=MultipleSideCoverage(SideCov, SPositionFOne, NumCov,
sideXmax, sideYmax);
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,1,n)= im2uint8(SideIMout);
            UpSidePercentage=[SIDEredcov SIDEorangecov SIDEgreencov];
        elseif FrameOneFlag == 0
            UpSidePercentage=[0 0 0];
        end

        if (FrameTwoFlag == 1)
            SMap=SideGrademapBW(:,:,2);
            NumCov=length(FrameTwoIndex);
            for i=1:NumCov,
SPositionFTwo(i)=SPosition(FrameTwoIndex(i)); end
            [SCov]=MultipleSideCoverage(SideCov, SPositionFTwo, NumCov,
sideXmax, sideYmax);
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,2,n)= im2uint8(SideIMout);
            RightSidePercentage=[SIDEredcov SIDEorangecov SIDEgreencov];
        elseif FrameTwoFlag==0
            RightSidePercentage=[0 0 0];
        end

        if (FrameThreeFlag == 1)
            SMap=SideGrademapBW(:,:,3);
            NumCov=length(FrameThreeIndex);
            for i=1:NumCov,
SPositionFThree(i)=SPosition(FrameThreeIndex(i)); end
            [SCov]=MultipleSideCoverage(SideCov, SPositionFThree,
NumCov, sideXmax, sideYmax);
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,3,n)= im2uint8(SideIMout);
            BottomSidePercentage=[SIDEredcov SIDEorangecov
SIDEgreencov];
        elseif FrameThreeFlag==0
            BottomSidePercentage=[0 0 0];
        end

        if (FrameFourFlag == 1)
            SMap=SideGrademapBW(:,:,4);
            NumCov=numel(FrameFourIndex);
            for i=1:NumCov,
SPositionFour(i)=SPosition(FrameFourIndex(i)); end
            [SCov]=MultipleSideCoverage(SideCov, SPositionFour, NumCov,
sideXmax, sideYmax);
            [SideIMout, SIDEredcov, SIDEorangecov,
SIDEgreencov]=ComputeCoverage(SMap,SCov,loop,SideGradeArea,sideXmax,sid
eYmax);
            SResult(:,:,:,4,n)= im2uint8(SideIMout);
            LeftSidePercentage=[SIDEredcov SIDEorangecov SIDEgreencov];
        elseif FrameFourFlag == 0
            LeftSidePercentage=[0 0 0];
        end

        %result that met the safety requirement
        FinalResult(n,:)=[j n TPositionOne TPositionTwo TPositionThree
TPositionFour TOPredcov(j) TOPorangecov(j) TOPgreencov(j)
UpSidePercentage RightSidePercentage BottomSidePercentage
LeftSidePercentage];
        ResultFlag=1;
    end
end

disp(TopFourCovResult)
    a=1:NumTopCov;
    figure(33)
    plot(a,TOPredcov(a),'r'),    hold on
    plot(a,TOPorangecov(a),'m')
    plot(a,TOPgreencov(a),'g'),    hold off
    axis([0 NumTopCov 0 100])
    title('% of detector coverage from 495 possible point');
    h = get(gca, 'title');
    set(h, 'FontSize', 15)
    legend('% Coverage on Red area','% Coverage on Orange area','%
Coverage on Green area')

%if there is result met the requirement
 if ResultFlag==1
    fprintf('found the optimum placement!');
    fprintf('\n');
    fprintf('Obtained %d possible placement that met safety
requirement!\n\n',n);
    fprintf('Result being sort in descending order:\n');
    fprintf(1, '\tNo. RefNo   P1    P2   P3   P4  R(T)   O(T)   G(T)
R(US)   O(US)   G(US)  R(RS)   O(RS)   G(RS)  R(BS)   O(BS)   G(BS)
R(LS)   O(LS)   G(LS)\n');
```

```
    Rearrange=sortrows(FinalResult,[-7 -8 -9]); %sort the table
according to the % of orange cov (as red cov is the max 100%)
    disp(Rearrange)

    for j=1:n
            Position=Rearrange(j,2); %out of 220 possibility

            PointOne=Rearrange(j,3);
            PointTwo=Rearrange(j,4);
            PointThree=Rearrange(j,5);
            PointFour=Rearrange(j,6);

            figure('name',sprintf('No.%d Recommended Detector
Placement',j,Position));
            subplot(3,3,2), subimage(SResult(:,:,:,1,Position))
            subplot(3,3,4), subimage(SResult(:,:,:,4,Position))
            subplot(3,3,5), subimage(TResult(:,:,:,Position))
            subplot(3,3,6), subimage(SResult(:,:,:,2,Position))
            subplot(3,3,8), subimage(SResult(:,:,:,3,Position))

            %display the result in text form
            subplot(3,3,9)
                if j==1  %top result in table
                text(.05,1.3,['Best Allocation:'])
                end
                text(.05,1.1,['No. of detector required: 3'])
                text(.05,0.9,sprintf('Suggested Position: %d & %d & %d
& %d',PointOne,PointTwo,PointThree,PointFour))
                text(.05,0.7,sprintf('Covered Red
Area: %d',Rearrange(j,7)))
                text(.05,0.5,sprintf('Covered Orange
Area: %d',Rearrange(j,8)))
                text(.05,0.3,sprintf('Covered Green
Area: %d',Rearrange(j,9)))
                axis([0 5 0 1.5])
                axis off
    end
    return
else  %if there is no result met the requirement
    fprintf('more detector is needed!\n');
end


%----------Subfunction One: imagesegmentation--------------------------

function [imRGB,imBW] = imagesegmentation (IMinput, maxX, maxY)

imRGB = imresize (IMinput, [maxY maxX], 'nearest');
imBW= im2double(rgb2gray(imRGB));

black=0;
red=0.35;
green=0.48;
orange=0.67;
```

```
 for x = 1:maxX
        for y = 1:maxY
            if (imBW(y,x)>=0 && imBW(y,x)<=0.1);
                imBW(y,x)=black;
            end
            if (imBW(y,x)>=0.30 && imBW(y,x)<=0.4);
                imBW(y,x)=red;
            end
            if (imBW(y,x)>=0.43 && imBW(y,x)<=0.53);
                imBW(y,x)=green;
            end
            if (imBW(y,x)>=0.60 && imBW(y,x)<=0.73);
                imBW(y,x)=orange;
            end
        end
 end

%------------Subfunction: topcoverage-------------------------
function [coveragearea, i, ReferencePoint] =
topcoverage(angle,length,xmax,ymax)

%create the image size
I=zeros([ymax xmax]);

%variable for edge computation
d=tand(angle)*length ;

%variable for corner computation
H=norm([d length]);
adj=cosd(angle)*H;
opp=sind(angle)*H;

i=0;
%ignore the location at corner due to the complex assumption when side
view
%is taking into consideration
%top
for x=80:80:(xmax-1)
    y=0;
    i=i+1;

    ReferencePoint(i,:)=[x y];

    %first point
    x1=x;
    y1=y;
    x2=x-d;
    y2=length;
    x3=x+d;
    y3=length;

    %create coverage
    c=[x1 x2 x3];
    r=[y1 y2 y3];
    coveragearea(:,:,i)=im2double(roipoly(I,c,r));

end

%right
for y=80:80:(ymax-1)
    x=xmax;
    i=i+1;

    ReferencePoint(i,:)=[x y];

    %first point
    x1=x;
    y1=y;
    x2=x-length;
    y2=y-d;
    x3=x-length;
    y3=y+d;

    c=[x1 x2 x3];
    r=[y1 y2 y3];
    coveragearea(:,:,i)=im2double(roipoly(I,c,r));
end

%bottom
for x=(xmax-80):-80:1
     y=ymax;
    i=i+1;

    ReferencePoint(i,:)=[x y];

    x1=x;
    y1=y;
    x2=x-d;
    y2=y-length;
    x3=x+d;
    y3=y-length;

    c=[x1 x2 x3];
    r=[y1 y2 y3];
    coveragearea(:,:,i)=im2double(roipoly(I,c,r));

end

%left
 for y=(ymax-80):-80:1
    x=0;
    i=i+1;

    ReferencePoint(i,:)=[x y];

    x1=x;
    y1=y;
    x2=x+length;
    y2=y+d;
    x3=x+length;
    y3=y-d;

    c=[x1 x2 x3];
    r=[y1 y2 y3];
    coveragearea(:,:,i)=im2double(roipoly(I,c,r));

 end


%----------------Subfunction: sidecoverage----------------------
function [CircleCov,i] =sidecoverage(angle,length, maxX, maxY)

d=tand(angle)*length ;
H=norm([d length]);
radius=sind(newangle)*H;

maxX = 320;
maxY = 160;
[columnsInImage rowsInImage] = meshgrid(1:maxX, 1:maxY);

i=0;
    for x=80:80:(maxX-1)
          i=i+1;
          centerX = x;
          centerY = 20+radius;
          circlePixels = (rowsInImage - centerY).^2 + (columnsInImage -
centerX).^2 <= radius.^2;
          CircleCov(:,:,i)=double(circlePixels);
    end


%----------------Subfunction: blockage-----------------
function [TopBlockageArea]= blockage(TopGrademap, TopCoverage,
NumTopCoverage, RefPoint, Xmax,Ymax)

%grade map segmentation & object extraction
for x=1:Xmax
    for y=1:Ymax
        if TopGrademap(y,x)<=0.1
            TopTempt(y,x)=0; %black
        else
            TopTempt(y,x)=1;
        end
    end
end
TopTemptFill=imfill(edge(TopTempt,'canny'),'holes');
[TopSegmented,NumObject]=bwlabel(TopTemptFill);

for a=1:NumTopCoverage

    BlockageArea=zeros([Ymax Xmax]);
    for i=1:NumObject
```

```
        Flag=0;
        %find the 1st & 2nd point from top
        for y=1:Ymax
            for x=1:Xmax
                if ((TopCoverage(y,x,a)>=0.8) && (TopSegmented(y,x) ==
i))
                    P(1,:)=[x y];
                    Flag=1;
                    break
                end
            end

            if Flag==1
                for x=Xmax:-1:1
                    if ((TopCoverage(y,x,a)>=0.8) && (TopSegmented(y,x)
== i)), P(2,:)=[x y]; break, end
                end
            end

            if Flag==1, break, end %found the 2 points
        end

        if Flag == 1
            temptFlag=0;
            %find the 3rd & 4th points from bottom
            for y=Ymax:-1:1
                for x=1:Xmax
                    if ((TopCoverage(y,x,a)>=0.8) && (TopSegmented(y,x)
== i))
                        P(3,:)=[x y];
                        temptFlag=1;
                        break
                    end
                end

                if temptFlag==1
                    for x=Xmax:-1:1
                        if ((TopCoverage(y,x,a)>=0.8) &&
(TopSegmented(y,x) == i)), P(4,:)=[x y]; break, end
                    end
                end

                if temptFlag==1, break, end %found the 2 points
            end

            temptFlag=0;
            %find the 5th & 6th points from left
            for x=1:Xmax
                for y=1:Ymax
                    if ((TopCoverage(y,x,a)>=0.8) && (TopSegmented(y,x)
== i))
                        P(5,:)=[x y];
                        temptFlag=1;
                        break
                    end
                end
                if temptFlag==1
                    for y=Ymax:-1:1
                        if ((TopCoverage(y,x,a)>=0.8) &&
(TopSegmented(y,x) == i)), P(6,:)=[x y]; break, end
                    end
                end

                if temptFlag==1, break, end %found the 2 points
            end

            temptFlag=0;
            %find the 7th & 8th points from right
            for x=Xmax:-1:1
               for y=1:Ymax
                    if ((TopCoverage(y,x,a)>=0.8) && (TopSegmented(y,x)
== i))
                        P(7,:)=[x y];
                        temptFlag=1;
                        break
                    end
                end

                if temptFlag==1
                    for y=Ymax:-1:1
                        if ((TopCoverage(y,x,a)>=0.8) &&
(TopSegmented(y,x) == i)), P(8,:)=[x y]; break, end
                    end
                end

                if temptFlag==1, break, end %found the 2 points
            end

            %find the boundary of the shadow
            for j=1:8
                if (RefPoint(a,2)== 0 || RefPoint(a,2)== Ymax)
                    if P(j,1)< RefPoint(a,1)
                        BlockageAngle(j)=180-atand(abs((RefPoint(a,2)-
P(j,2))/(RefPoint(a,1)-P(j,1))));
                    else
                        BlockageAngle(j)=atand(abs((RefPoint(a,2)-
P(j,2))/(RefPoint(a,1)-P(j,1))));
                    end
                elseif (RefPoint(a,1)== 0 || RefPoint(a,1)== Xmax)
                    if P(j,2)< RefPoint(a,2)
                        BlockageAngle(j)=180-atand(abs((RefPoint(a,1)-
P(j,1))/(RefPoint(a,2)-P(j,2))));
                    else
                        BlockageAngle(j)=atand(abs((RefPoint(a,1)-
P(j,1))/(RefPoint(a,2)-P(j,2))));
                    end
                end
            end

            [MaxAngle indexMax]=max(BlockageAngle);
            [MinAngle indexMin]=min(BlockageAngle);

            I=zeros([Ymax Xmax]);
            lineLength=500;

            %find the blockage area
            if     RefPoint(a,2) == 0,        state=1; %top
            elseif RefPoint(a,2) == Ymax,     state=2; %bottom
            elseif RefPoint(a,1) == 0,        state=3; %left
            elseif RefPoint(a,1) == Xmax,     state=4; %right
            end

            xP=[0 0 0];
            yP=[0 0 0];

            xP(1)=RefPoint(a,1);
            yP(1)=RefPoint(a,2);

            LeftFrontFlag=0;
            RightFrontFlag=0;
            UpFrontFlag=0;
            DownFrontFlag=0;

            switch state
                case 1
                    ylimit=min([P(indexMax,2) P(indexMin,2)]);
                    xP(2) = xP(1) + lineLength * cosd(MaxAngle);
                    yP(2) = yP(1) + lineLength * sind(MaxAngle);
                    xP(3) = xP(1) + lineLength * cosd(MinAngle);
                    yP(3) = yP(1) + lineLength * sind(MinAngle);
                    BlockageBound(:,:)=im2double(roipoly(I,xP,yP));

                    for x=1:Xmax
                        for y=Ymax:-1:ylimit
                            if (TopSegmented(y,x) == i), break,
                            elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                BlockageArea(y,x,a)=1;
                            end
                        end
                    end

                    %checking
                    if P(indexMin,1)<xP(1)
                        RightFrontFlag=1;
                        RightXlimit=P(indexMin,1);
                    elseif P(indexMax,1)>xP(1)
                        LeftFrontFlag=1;
                        LeftXlimit=P(indexMax,1);
                    end

                    if LeftFrontFlag==1
                         for x=1:LeftXlimit
                            for y=ylimit:Ymax
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                     if RightFrontFlag==1
                         for x=Xmax:-1:RightXlimit
                            for y=ylimit:Ymax
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                case 2
                    ylimit=max([P(indexMax,2) P(indexMin,2)]);

                    xP(2) = xP(1) + lineLength * cosd(MaxAngle);
                    yP(2) = yP(1) - lineLength * sind(MaxAngle);
                    xP(3) = xP(1) + lineLength * cosd(MinAngle);
                    yP(3) = yP(1) - lineLength * sind(MinAngle);
                    BlockageBound(:,:)=im2double(roipoly(I,xP,yP));

                    for x=1:Xmax
                        for y=1:ylimit
                            if (TopSegmented(y,x) == i), break,
                            elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                BlockageArea(y,x,a)=1;
                            end
                        end
                    end

                    %checking
                    if P(indexMin,1)<xP(1)
                        RightFrontFlag=1;
                        RightXlimit=P(indexMin,1);
                    elseif P(indexMax,1)>xP(1)
                        LeftFrontFlag=1;
                        LeftXlimit=P(indexMax,1);
                    end


                    if LeftFrontFlag==1
                         for x=1:LeftXlimit
                            for y=ylimit:-1:1
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
```

```
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                     if RightFrontFlag==1
                         for x=Xmax:-1:RightXlimit
                            for y=ylimit:-1:1
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                case 3
                    xlimit=min([P(indexMax,1) P(indexMin,1)]);
                    xP(2) = xP(1) + lineLength * sind(MaxAngle);
                    yP(2) = yP(1) + lineLength * cosd(MaxAngle);
                    xP(3) = xP(1) + lineLength * sind(MinAngle);
                    yP(3) = yP(1) + lineLength * cosd(MinAngle);
                    BlockageBound(:,:)=im2double(roipoly(I,xP,yP));

                    for y=1:Ymax
                        for x=Xmax:-1:xlimit
                            if (TopSegmented(y,x) == i), break,
                            elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                BlockageArea(y,x,a)=1;
                            end
                        end
                    end

                    %checking
                    if P(indexMin,2)<yP(1)
                        DownFrontFlag=1;
                        DownYlimit=P(indexMin,2);
                    elseif P(indexMax,2)>yP(1)
                        UpFrontFlag=1;
                        UpYlimit=P(indexMax,2);
                    end

                    if UpFrontFlag==1
                         for y=1:UpYlimit
                            for x=xlimit:Xmax
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end
                     if DownFrontFlag==1
                         for y=Ymax:-1:DownYlimit
                            for x=xlimit:Xmax
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                case 4
                    xlimit=max([P(indexMax,1) P(indexMin,1)]);
                    xP(2) = xP(1) - lineLength * sind(MaxAngle);
                    yP(2) = yP(1) + lineLength * cosd(MaxAngle);
                    xP(3) = xP(1) - lineLength * sind(MinAngle);
                    yP(3) = yP(1) + lineLength * cosd(MinAngle);
                    BlockageBound(:,:)=im2double(roipoly(I,xP,yP));

                    for y=1:Ymax
                        for x=1:xlimit
                            if (TopSegmented(y,x) == i), break,
                            elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                BlockageArea(y,x,a)=1;
                            end
                        end
                    end

                    %checking

                    if P(indexMin,2)<yP(1)
                        DownFrontFlag=1;
                        DownYlimit=P(indexMin,2);
                    elseif P(indexMax,2)>yP(1)
                        UpFrontFlag=1;
                        UpYlimit=P(indexMax,2);
                    end


                    if UpFrontFlag==1
                         for y=1:UpYlimit
                            for x=xlimit:-1:1
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end

                     if DownFrontFlag==1
```

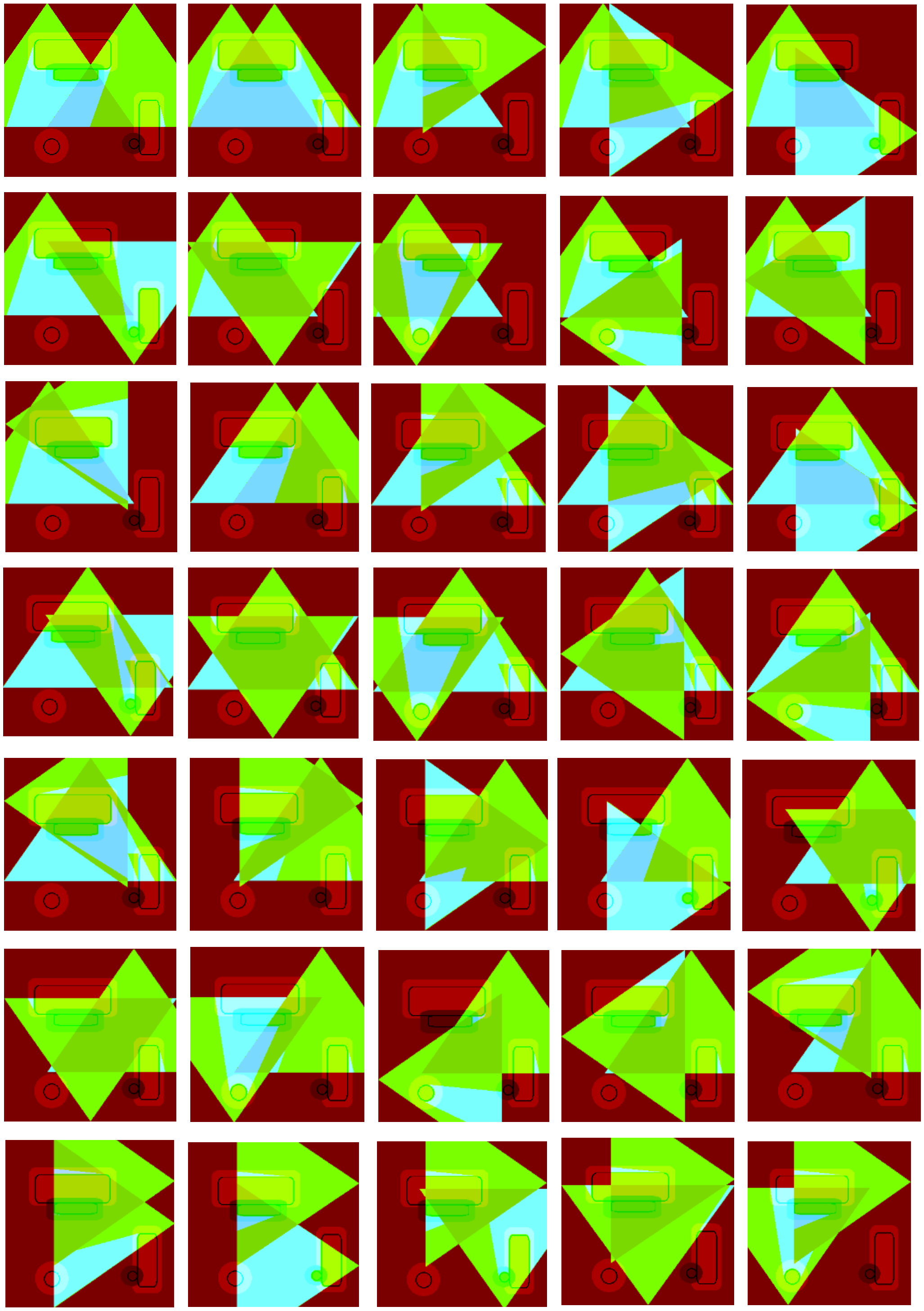
```
                         for y=Ymax:-1:DownYlimit
                            for x=xlimit:-1:1
                                if (TopSegmented(y,x) == i), break,
                                elseif ((BlockageBound(y,x)==1) &&
(TopCoverage(y,x,a)>0.8))
                                    BlockageArea(y,x,a)=0;
                                end
                            end
                         end
                     end
            end


        end

    end
      TopBlockageArea(:,:,a)=BlockageArea(:,:,a);
end
end

%-----------------Subfunction: ComputeTopCoverage-------------
function [IMout,redcoverage, orangecoverage,greencoverage] =
ComputeTopCoverage (map,coverage,blockage,TotalArea,xmax,ymax)
%initialize
red=0.35;
green=0.48;
orange=0.67;
pixelgreen=0;
pixelred=0;
pixelorange=0;

IMout(:,:,1)=map; %redframe
IMout(:,:,2)=coverage; %greenframe
IMout(:,:,3)=blockage;

        %compute percentage of coverage
 for x = 1:xmax
     for y = 1:ymax
         %calculate the area within coverage
         if (IMout(y,x,2)>0.8 && IMout(y,x,3)==0);
             if (IMout(y,x,1) == green);
                 pixelgreen=pixelgreen+1;
             end

             if (IMout(y,x,1) == red);
                 pixelred=pixelred+1;
             end

             if (IMout(y,x,1) == orange);
                 pixelorange=pixelorange+1;
             end
          end
     end
 end
```

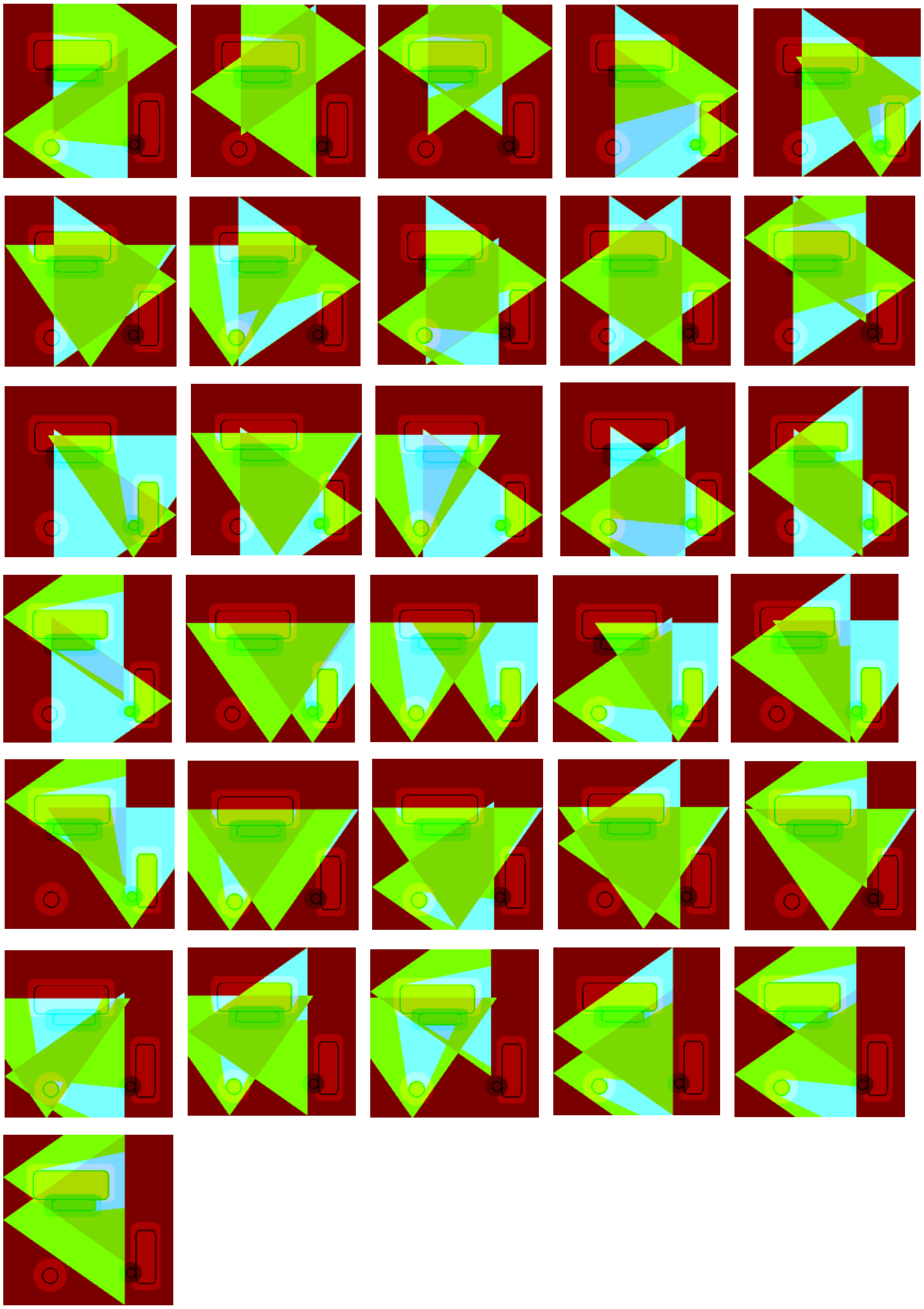
```
if TotalArea(1)==0
    redcoverage=0;
else
    redcoverage=round((pixelred/TotalArea(1))*100);
end

if TotalArea(2)==0
    orangecoverage=0;
else
    orangecoverage=round((pixelorange/TotalArea(2))*100);
end

if TotalArea(3)==0
    greencoverage=0;
else
    greencoverage=round((pixelgreen/TotalArea(3))*100);
end


%-------------------Subfunction: ComputeCoverage-----------
function [IMout,redcoverage, orangecoverage,greencoverage] =
ComputeCoverage (map,coverage,NumComputation,TotalArea,xmax,ymax)
    %initialize
    red=0.35;
    green=0.48;
    orange=0.67;
    pixelgreen=0;
    pixelred=0;
    pixelorange=0;

    for i=1:NumComputation
        IMout(:,:,1,i)=map(:,:,i); %redframe
        IMout(:,:,2,i)=coverage(:,:,i); %greenframe

        for x = 1:xmax
            for y = 1:ymax
                 IMout(y,x,3,i)=0; %blueframe
            end
        end

        %compute percentage of coverage
        for x = 1:xmax
            for y = 1:ymax
                %calculate the area within coverage
                if IMout(y,x,2,i)>0.8;
                    if (IMout(y,x,1,i) == green);
                        pixelgreen=pixelgreen+1;
                    end

                    if (IMout(y,x,1,i) == red);
                        pixelred=pixelred+1;
                    end

                    if (IMout(y,x,1,i) == orange);
                        pixelorange=pixelorange+1;
                    end
                 end
             end
        end
    end

if TotalArea(1)==0
    redcoverage=0;
else
    redcoverage=round((pixelred/TotalArea(1))*100);
end

if TotalArea(2)==0
    orangecoverage=0;
else
    orangecoverage=round((pixelorange/TotalArea(2))*100);
end

if TotalArea(3)==0
    greencoverage=0;
else
    greencoverage=round((pixelgreen/TotalArea(3))*100);
end


%-------------Subfunction: SidePosition---------------
function [frame,SideCoveragePosition] = SidePosition(TopPosition)

%Top side map is store in frame 1
%Right side map is store in frame 2
%Bottom side map is store in frame 3
%Left side map is store in frame 4
switch (TopPosition)
    case 1
        frame = 1; %UP
        SideCoveragePosition = 1;
    case 2
        frame = 1;
        SideCoveragePosition = 2;
    case 3
        frame = 1;
        SideCoveragePosition = 3;
    case 4
        frame = 2;
        SideCoveragePosition = 3;
    case 5
        frame = 2;
        SideCoveragePosition = 2;
    case 6
        frame = 2;
        SideCoveragePosition = 1;
    case 7
        frame = 3;
        SideCoveragePosition = 3;
    case 8
        frame = 3;
        SideCoveragePosition = 2;
    case 9
        frame = 3;
        SideCoveragePosition = 1;
    case 10
        frame = 4;
        SideCoveragePosition = 3;
    case 11
        frame = 4;
        SideCoveragePosition = 2;
    case 12
        frame = 4;
        SideCoveragePosition = 1;
end

%---------------MultipleTopCoverage---------------------
function [TopIMout] = MultipleTopCoverage(coverage, combination,
NumDetector, Xmax, Ymax)

 switch NumDetector
     case 2
             P1 = combination(1);
             P2 = combination(2);

             for x=1:Xmax
                 for y=1:Ymax
                     if (coverage(y,x,P1)>0.8 && coverage(y,x,P2)>0.8)
                         TopIMout(y,x) = 0.85;
                     elseif (coverage(y,x,P1)>0.8 ||
coverage(y,x,P2)>0.8)
                         TopIMout(y,x) = 1;
                     else
                         TopIMout(y,x) = 0;
                     end
                 end
             end
     case 3
             P1 = combination(1);
             P2 = combination(2);
             P3 = combination(3);

             for x=1:Xmax
                 for y=1:Ymax
                     if (coverage(y,x,P1)>0.8 && coverage(y,x,P2)>0.8
&& coverage(y,x,P3)>0.8)
                         TopIMout(y,x) = 0.85;
                     elseif ((coverage(y,x,P1)>0.8 &&
coverage(y,x,P2)>0.8) ||(coverage(y,x,P1)>0.8 &&
coverage(y,x,P3)>0.8)||(coverage(y,x,P2)>0.8 && coverage(y,x,P3)>0.8))
                         TopIMout(y,x) = 0.90;
                     elseif (coverage(y,x,P1)>0.8 ||
coverage(y,x,P2)>0.8 ||coverage(y,x,P3)>0.8)
                         TopIMout(y,x) = 1;
                     else
                         TopIMout(y,x) = 0;
                     end
                 end
             end

     case 4
             P1 = combination(1);
             P2 = combination(2);
             P3 = combination(3);
             P4 = combination(4);

             for x=1:Xmax
                 for y=1:Ymax
                     if (coverage(y,x,P1)>0.8 && coverage(y,x,P2)>0.8
&& coverage(y,x,P3)>0.8 && coverage(y,x,P4)>0.8)
                         TopIMout(y,x) = 0.85;
                     elseif (coverage(y,x,P1)>0.8 &&
coverage(y,x,P2)>0.8 && coverage(y,x,P3)>0.8)||(coverage(y,x,P1)>0.8 &&
coverage(y,x,P2)>0.8 && coverage(y,x,P4)>0.8)||(coverage(y,x,P1)>0.8 &&
coverage(y,x,P3)>0.8 && coverage(y,x,P4)>0.8)||(coverage(y,x,P2)>0.8 &&
coverage(y,x,P3)>0.8 && coverage(y,x,P4)>0.8)
                         TopIMout(y,x) = 0.90;
                     elseif (coverage(y,x,P1)>0.8 &&
coverage(y,x,P2)>0.8)||(coverage(y,x,P1)>0.8 &&
coverage(y,x,P3)>0.8)||(coverage(y,x,P1)>0.8 &&
coverage(y,x,P4)>0.8)||(coverage(y,x,P2)>0.8 &&
coverage(y,x,P3)>0.8)||(coverage(y,x,P2)>0.8 &&
coverage(y,x,P4)>0.8)||(coverage(y,x,P3)>0.8 && coverage(y,x,P4)>0.8)
                         TopIMout(y,x) = 0.95;
                     elseif (coverage(y,x,P1)>0.8 ||
coverage(y,x,P2)>0.8 || coverage(y,x,P3)>0.8 || coverage(y,x,P4)>0.8)
                         TopIMout(y,x) = 1;
                     else
                         TopIMout(y,x) = 0;
                    end
                 end
             end
 end

 end
%-------------Subfunction: MultipleSideCoverage--------------
function [SideIMout]= MultipleSideCoverage(coverage, Position,
NumDetector, Xmax, Ymax)

switch NumDetector
    case 1
        SideIMout=coverage(:,:,Position(1));
    case 2
        PositionOne=Position(1);
        PositionTwo=Position(2);
        for x=1:Xmax
            for y=1:Ymax
                if (coverage(y,x,PositionOne)>0.8 ||
coverage(y,x,PositionTwo)>0.8)
                    SideIMout(y,x)=1;
                else
                    SideIMout(y,x)=0;
                end
            end
        end

    case 3
        for x=1:Xmax
            for y=1:Ymax
                if (coverage(y,x,1)>0.8 || coverage(y,x,2)>0.8 ||
coverage(y,x,3)>0.8)
                    SideIMout(y,x)=1;
                else
                    SideIMout(y,x)=0;
                end
            end
        end
end


%--------------Subfunction: MultipleTopBlockage------------------
function
[BlockageIMout]=MultipleTopBlockage(coverage,blockage,combination,NumDe
tector,Xmax,Ymax)

 switch NumDetector
     case 2
             POne = combination(1);
             P2 = combination(2);

             for x=1:Xmax
                 for y=1:Ymax
                     if (blockage(y,x,POne)==1 && blockage(y,x,P2)==1)
                         BlockageIMout(y,x) = 1; %blockage
                     elseif (blockage(y,x,POne)==1 ||
blockage(y,x,P2)==1)
                         if coverage(y,x)==0.85,
BlockageIMout(y,x)=0; %no blockage
                         else                    BlockageIMout(y,x)=1;
                         end
                     else
                         BlockageIMout(y,x) = 0;
                     end
                 end
             end
             %imshow(TopIMout(:,:,i))


     case 3
             POne = combination(1);
             PTwo = combination(2);
             PThree = combination(3);

             for x=1:Xmax
                 for y=1:Ymax
                     if (blockage(y,x,POne)==1 && blockage(y,x,PTwo)==1
&& blockage(y,x,PThree)==1)
                         BlockageIMout(y,x) = 1; %blockage
                     elseif (blockage(y,x,POne)==1 &&
blockage(y,x,PTwo)==1) ||(blockage(y,x,POne)==1 &&
blockage(y,x,PThree)==1)||(blockage(y,x,PTwo)==1 &&
blockage(y,x,PThree)==1)
                         if coverage(y,x)==0.85, BlockageIMout(y,x) =
0; %no blockage
                         else                    BlockageIMout(y,x) = 1;
                         end
                     elseif (blockage(y,x,POne)==1 ||
blockage(y,x,PTwo)==1 || blockage(y,x,PThree)==1)
                         if (coverage(y,x)==0.85) ||
(coverage(y,x)==0.90), BlockageIMout(y,x) = 0; %no blockage
                         else
BlockageIMout(y,x) = 1;
                         end
                     else
                          BlockageIMout(y,x) = 0;
                     end
                 end
             end


     case 4
             POne = combination(1);
             PTwo = combination(2);
             PThree = combination(3);
             PFour = combination(4);

             for x=1:Xmax
                 for y=1:Ymax
                     if (blockage(y,x,POne)==1 && blockage(y,x,PTwo)==1
&& blockage(y,x,PThree)==1 && blockage(y,x,PFour)==1)
                         BlockageIMout(y,x) = 1;
                     elseif (blockage(y,x,POne)==1 &&
blockage(y,x,PTwo)==1 && blockage(y,x,PThree)==1) ||
(blockage(y,x,POne)==1 && blockage(y,x,PTwo)==1 &&
blockage(y,x,PFour)==1)||(blockage(y,x,POne)==1 &&
blockage(y,x,PThree)==1 &&
blockage(y,x,PFour)==1)||(blockage(y,x,PTwo)==1 &&
blockage(y,x,PThree)==1 && blockage(y,x,PFour)==1)
                         if coverage(y,x)==0.85, BlockageIMout(y,x) =
0; %no blockage
                         else                    BlockageIMout(y,x) = 1;
                         end
                     elseif (blockage(y,x,POne)==1 &&
blockage(y,x,PTwo)==1)||(blockage(y,x,POne)==1 &&
blockage(y,x,PThree)==1)||(blockage(y,x,POne)==1 &&
blockage(y,x,PFour)==1)||(blockage(y,x,PTwo)==1 &&
blockage(y,x,PThree)==1) ||(blockage(y,x,PTwo)==1 &&
blockage(y,x,PFour)==1)||(blockage(y,x,PThree)==1 &&
blockage(y,x,PFour)==1)
                         if (coverage(y,x)==0.85 ||
coverage(y,x)==0.90), BlockageIMout(y,x) = 0; %no blockage
                         else
BlockageIMout(y,x) = 1;
                         end
                     elseif (blockage(y,x,POne)==1 ||
blockage(y,x,PTwo)==1 ||blockage(y,x,PThree)==1 ||
blockage(y,x,PFour)==1)
                         if (coverage(y,x)==0.85 || coverage(y,x)==0.90
|| coverage(y,x)==0.95), BlockageIMout(y,x) = 0;
                         else
BlockageIMout(y,x) = 1;
                         end
                     else
                         BlockageIMout(y,x) = 0;
                     end
                 end
             end
 end
end
```

APPENDIX C

RESULT OF THE TOP VIEW ANALYSIS FOR TWO DETECTORS

C.1 All possible placement of two detectors





C.2 Coverage factor for all possible placement of two detectors at each hazardous level

No.	Position 1	Position 2	% of Red	% of Orange	% of Green
1	1	2	48	52	13
2	1	3	48	44	29
3	1	4	59	49	28
4	1	5	63	51	21
5	1	6	54	42	8
6	1	7	77	46	18
7	1	8	71	31	36
8	1	9	52	39	19
9	1	10	58	41	26
10	1	11	74	40	27
11	1	12	55	41	15
12	2	3	48	41	24
13	2	4	59	38	22
14	2	5	63	41	16
15	2	6	58	50	6
16	2	7	81	52	16
17	2	8	75	38	32
18	2	9	56	46	19
19	2	10	61	48	24
20	2	11	74	37	25
21	2	12	55	52	14
22	3	4	36	31	31
23	3	5	40	29	25
24	3	6	24	30	22
25	3	7	55	32	29
26	3	8	54	17	43
27	3	9	21	27	34
28	3	10	26	28	39
29	3	11	74	36	40
30	3	12	55	53	32
31	4	5	40	32	27
32	4	6	46	39	22
33	4	7	64	43	28
34	4	8	63	28	45
35	4	9	36	35	32

36	4	10	40	37	37
37	4	11	78	33	38
38	4	12	66	49	27
39	5	6	50	40	14
40	5	7	64	42	19
41	5	8	63	28	32
42	5	9	40	37	24
43	5	10	41	39	28
44	5	11	78	35	29
45	5	12	69	51	23
46	6	7	49	20	11
47	6	8	57	17	29
48	6	9	20	25	15
49	6	10	31	27	20
50	6	11	84	37	20
51	6	12	65	54	11
52	7	8	65	21	30
53	7	9	48	29	22
54	7	10	51	31	25
55	7	11	95	42	26
56	7	12	88	58	20
57	8	9	47	13	38
58	8	10	49	15	35
59	8	11	77	26	31
60	8	12	77	42	38
61	9	10	22	15	26
62	9	11	77	34	28
63	9	12	62	50	24
64	10	11	76	35	26
65	10	12	67	52	28
66	11	12	74	40	29

APPENDIX D

RESULT OF THE TOP VIEW ANALYSIS FOR THREE DETECTORS

D.1 Coverage factor for all possible placement of three detectors at each hazardous level

No.	Position 1	Position 2	Position 3	% on Red	% on Orange	% on Green
1	1	2	3	48	67	37
2	1	2	4	59	60	35
3	1	2	5	70	63	28
4	1	2	6	58	70	14
5	1	2	7	82	71	24
6	1	2	8	78	62	45
7	1	2	9	57	70	26
8	1	2	10	64	70	39
9	1	2	11	79	58	41
10	1	2	12	55	59	21
11	1	3	4	59	67	49
12	1	3	5	70	62	41
13	1	3	6	58	63	35
14	1	3	7	82	65	40
15	1	3	8	78	58	53
16	1	3	9	57	65	45
17	1	3	10	64	66	53
18	1	3	11	79	63	55
19	1	3	12	55	64	43
20	1	4	5	70	57	41
21	1	4	6	69	64	35
22	1	4	7	82	68	41
23	1	4	8	78	54	56
24	1	4	9	59	61	43
25	1	4	10	66	62	52
26	1	4	11	83	50	54
27	1	4	12	66	51	36
28	1	5	6	80	65	25
29	1	5	7	88	67	30
30	1	5	8	78	54	43
31	1	5	9	70	62	33
32	1	5	10	74	64	40

33	1	5	11	83	52	42
34	1	5	12	74	53	33
35	1	6	7	83	55	19
36	1	6	8	88	54	42
37	1	6	9	65	57	22
38	1	6	10	72	56	34
39	1	6	11	89	56	36
40	1	6	12	65	57	18
41	1	7	8	96	58	43
42	1	7	9	82	65	29
43	1	7	10	83	65	35
44	1	7	11	96	61	36
45	1	7	12	89	62	27
46	1	8	9	78	50	48
47	1	8	10	81	52	48
48	1	8	11	83	48	46
49	1	8	12	78	49	49
50	1	9	10	64	47	39
51	1	9	11	83	54	42
52	1	9	12	64	55	30
53	1	10	11	81	55	42
54	1	10	12	71	56	42
55	1	11	12	79	43	43
56	2	3	4	59	53	41
57	2	3	5	70	52	35
58	2	3	6	58	60	29
59	2	3	7	82	60	34
60	2	3	8	78	57	49
61	2	3	9	57	62	44
62	2	3	10	64	63	47
63	2	3	11	79	52	50
64	2	3	12	55	67	39
65	2	4	5	70	49	34
66	2	4	6	69	59	29
67	2	4	7	82	59	35
68	2	4	8	78	53	50
69	2	4	9	59	57	42
70	2	4	10	66	58	46
71	2	4	11	83	48	49
72	2	4	12	66	60	33
73	2	5	6	80	60	20

74	2	5	7	88	60	25
75	2	5	8	78	54	39
76	2	5	9	70	59	33
77	2	5	10	74	60	35
78	2	5	11	83	50	38
79	2	5	12	74	63	30
80	2	6	7	83	57	17
81	2	6	8	88	63	39
82	2	6	9	67	66	23
83	2	6	10	74	66	32
84	2	6	11	89	56	34
85	2	6	12	65	70	17
86	2	7	8	96	63	40
87	2	7	9	82	67	30
88	2	7	10	83	67	33
89	2	7	11	96	57	35
90	2	7	12	89	71	25
91	2	8	9	78	60	48
92	2	8	10	81	62	45
93	2	8	11	83	50	43
94	2	8	12	78	61	47
95	2	9	10	66	57	41
96	2	9	11	83	55	44
97	2	9	12	64	69	33
98	2	10	11	81	55	40
99	2	10	12	71	70	40
100	2	11	12	79	58	42
101	3	4	5	68	51	43
102	3	4	6	67	61	42
103	3	4	7	80	61	45
104	3	4	8	78	58	56
105	3	4	9	57	62	55
106	3	4	10	64	63	54
107	3	4	11	83	54	57
108	3	4	12	66	66	45
109	3	5	6	78	56	33
110	3	5	7	86	56	36
111	3	5	8	78	54	47
112	3	5	9	68	58	46
113	3	5	10	71	59	45
114	3	5	11	83	49	48

115	3	5	12	74	62	44
116	3	6	7	74	46	32
117	3	6	8	88	51	47
118	3	6	9	33	47	42
119	3	6	10	40	48	45
120	3	6	11	89	55	48
121	3	6	12	65	71	39
122	3	7	8	96	50	49
123	3	7	9	73	55	45
124	3	7	10	74	56	46
125	3	7	11	96	55	49
126	3	7	12	89	71	43
127	3	8	9	78	52	56
128	3	8	10	81	53	53
129	3	8	11	83	51	52
130	3	8	12	78	63	57
131	3	9	10	32	41	54
132	3	9	11	83	57	57
133	3	9	12	64	73	53
134	3	10	11	81	58	54
135	3	10	12	71	74	56
136	3	11	12	79	62	59
137	4	5	6	78	55	34
138	4	5	7	86	57	39
139	4	5	8	78	50	52
140	4	5	9	68	53	47
141	4	5	10	71	54	49
142	4	5	11	83	44	51
143	4	5	12	74	57	38
144	4	6	7	80	55	34
145	4	6	8	88	58	51
146	4	6	9	67	60	41
147	4	6	10	74	61	45
148	4	6	11	93	51	48
149	4	6	12	75	64	32
150	4	7	8	96	62	52
151	4	7	9	80	65	47
152	4	7	10	80	66	46
153	4	7	11	96	56	48
154	4	7	12	89	68	37
155	4	8	9	78	55	60

156	4	8	10	81	56	57
157	4	8	11	83	44	56
158	4	8	12	78	53	53
159	4	9	10	64	49	53
160	4	9	11	83	48	57
161	4	9	12	66	60	45
162	4	10	11	83	49	53
163	4	10	12	73	62	48
164	4	11	12	83	50	51
165	5	6	7	86	53	22
166	5	6	8	88	57	37
167	5	6	9	78	60	30
168	5	6	10	81	61	32
169	5	6	11	93	52	35
170	5	6	12	84	65	29
171	5	7	8	96	59	38
172	5	7	9	86	62	35
173	5	7	10	86	63	33
174	5	7	11	96	54	35
175	5	7	12	92	66	33
176	5	8	9	78	54	46
177	5	8	10	81	55	43
178	5	8	11	83	44	42
179	5	8	12	78	53	47
180	5	9	10	71	51	41
181	5	9	11	83	49	44
182	5	9	12	74	62	42
183	5	10	11	83	50	40
184	5	10	12	78	63	43
185	5	11	12	83	51	46
186	6	7	8	96	44	37
187	6	7	9	74	46	26
188	6	7	10	74	46	28
189	6	7	11	96	47	30
190	6	7	12	89	62	23
191	6	8	9	88	48	45
192	6	8	10	91	50	42
193	6	8	11	93	48	41
194	6	8	12	88	61	46
195	6	9	10	37	33	36
196	6	9	11	93	52	39

197	6	9	12	73	67	30
198	6	10	11	91	52	35
199	6	10	12	80	68	38
200	6	11	12	89	56	39
201	7	11	12	96	61	40
202	7	10	12	89	73	39
203	7	10	11	96	57	36
204	7	9	12	89	72	36
205	7	9	11	96	56	39
206	7	9	10	74	47	37
207	7	8	12	96	65	48
208	7	8	11	96	52	41
209	7	8	10	96	54	43
210	7	8	9	96	52	46
211	8	11	12	83	47	51
212	8	10	12	81	58	52
213	8	10	11	83	46	46
214	8	9	12	78	57	55
215	8	9	11	83	45	48
216	8	9	10	81	37	48
217	9	10	11	83	41	42
218	9	10	12	73	57	46
219	9	11	12	83	54	49
220	10	11	12	81	54	46

APPENDIX E

RESULT OF THE TOP VIEW ANALYSIS FOR FOUR DETECTORS

E.1 Coverage factor for all possible placement of four detectors at each hazardous level

No	Position 1	Position 2	Position 3	Position 4	% on Red	% on Orange	% on Green	No	Position 1	Position 2	Position 3	Position 4	% on Red	% on Orange	% on Green
1	9	10	11	12	83	57	53	**251**	2	5	9	10	74	62	44
2	8	10	11	12	83	59	56	**252**	2	5	8	12	78	64	48
3	8	9	11	12	83	57	58	**253**	2	5	8	11	83	54	45
4	8	9	10	12	81	61	59	**254**	2	5	8	10	81	66	46
5	8	9	10	11	83	48	50	**255**	2	5	8	9	78	65	49
6	7	10	11	12	96	73	47	**256**	2	5	7	12	92	74	34
7	7	9	11	12	96	72	50	**257**	2	5	7	11	96	62	39
8	7	9	10	12	89	76	48	**258**	2	5	7	10	89	72	37
9	7	9	10	11	96	60	43	**259**	2	5	7	9	88	71	38
10	7	8	11	12	96	65	52	**260**	2	5	7	8	96	66	41
11	7	8	10	12	96	77	54	**261**	2	5	6	12	84	75	30
12	7	8	10	11	96	64	47	**262**	2	5	6	11	93	62	38
13	7	8	9	12	96	75	57	**263**	2	5	6	10	83	72	35
14	7	8	9	11	96	63	49	**264**	2	5	6	9	80	71	34
15	7	8	9	10	96	56	50	**265**	2	5	6	8	88	67	40

16	6	10	11	12	91	68	46	**266**	2	5	6	7	88	62	26
17	6	9	11	12	93	68	50	**267**	2	4	11	12	83	61	52
18	6	9	10	12	82	71	47	**268**	2	4	10	12	73	72	49
19	6	9	10	11	93	56	42	**269**	2	4	10	11	83	60	55
20	6	8	11	12	93	61	51	**270**	2	4	9	12	66	71	46
21	6	8	10	12	91	73	53	**271**	2	4	9	11	83	59	58
22	6	8	10	11	93	60	46	**272**	2	4	9	10	66	61	55
23	6	8	9	12	88	71	56	**273**	2	4	8	12	78	62	54
24	6	8	9	11	93	59	48	**274**	2	4	8	11	83	53	56
25	6	8	9	10	91	52	49	**275**	2	4	8	10	81	65	57
26	6	7	11	12	96	63	41	**276**	2	4	8	9	78	64	60
27	6	7	10	12	89	75	39	**277**	2	4	7	12	89	73	39
28	6	7	10	11	96	59	36	**278**	2	4	7	11	96	61	50
29	6	7	9	12	89	74	37	**279**	2	4	7	10	83	71	47
30	6	7	9	11	96	58	40	**280**	2	4	7	9	82	70	49
31	6	7	9	10	74	49	38	**281**	2	4	7	8	96	66	52
32	6	7	8	12	96	67	48	**282**	2	4	6	12	75	73	33
33	6	7	8	11	96	54	42	**283**	2	4	6	11	93	61	50
34	6	7	8	10	96	56	43	**284**	2	4	6	10	76	71	46
35	6	7	8	9	96	54	47	**285**	2	4	6	9	69	70	42
36	5	10	11	12	83	64	51	**286**	2	4	6	8	88	66	51
37	5	9	11	12	83	62	55	**287**	2	4	6	7	83	61	36
38	5	9	10	12	78	66	52	**288**	2	4	5	12	74	63	38
39	5	9	10	11	83	53	47	**289**	2	4	5	11	83	51	52
40	5	8	11	12	83	53	52	**290**	2	4	5	10	74	61	49
41	5	8	10	12	81	65	54	**291**	2	4	5	9	70	60	47

42	5	8	10	11	83	56	47	**292**	2	4	5	8	78	55	52
43	5	8	9	12	78	64	57	**293**	2	4	5	7	88	61	39
44	5	8	9	11	83	54	49	**294**	2	4	5	6	80	61	35
45	5	8	9	10	81	58	50	**295**	2	3	11	12	79	67	59
46	5	7	11	12	96	66	46	**296**	2	3	10	12	71	79	56
47	5	7	10	12	93	78	44	**297**	2	3	10	11	81	64	56
48	5	7	10	11	96	66	41	**298**	2	3	9	12	64	78	54
49	5	7	9	12	92	77	46	**299**	2	3	9	11	83	63	60
50	5	7	9	11	96	65	45	**300**	2	3	9	10	66	65	56
51	5	7	9	10	86	66	43	**301**	2	3	8	12	78	68	58
52	5	7	8	12	96	68	49	**302**	2	3	8	11	83	57	55
53	5	7	8	11	96	59	43	**303**	2	3	8	10	81	69	56
54	5	7	8	10	96	71	44	**304**	2	3	8	9	78	68	59
55	5	7	8	9	96	69	48	**305**	2	3	7	12	89	76	44
56	5	6	11	12	93	65	46	**306**	2	3	7	11	96	62	51
57	5	6	10	12	87	77	43	**307**	2	3	7	10	83	72	49
58	5	6	10	11	93	64	41	**308**	2	3	7	9	82	71	48
59	5	6	9	12	84	75	42	**309**	2	3	7	8	96	66	51
60	5	6	9	11	93	63	45	**310**	2	3	6	12	65	76	39
61	5	6	9	10	81	64	42	**311**	2	3	6	11	89	61	51
62	5	6	8	12	88	67	48	**312**	2	3	6	10	74	72	48
63	5	6	8	11	93	57	42	**313**	2	3	6	9	67	71	45
64	5	6	8	10	91	69	43	**314**	2	3	6	8	88	66	50
65	5	6	8	9	88	68	47	**315**	2	3	6	7	83	62	35
66	5	6	7	12	93	68	34	**316**	2	3	5	12	74	67	45
67	5	6	7	11	96	55	36	**317**	2	3	5	11	83	54	51

68	5	6	7	10	86	65	34	**318**	2	3	5	10	74	64	48
69	5	6	7	9	86	64	36	**319**	2	3	5	9	70	63	49
70	5	6	7	8	96	61	39	**320**	2	3	5	8	78	59	50
71	4	10	11	12	83	62	57	**321**	2	3	5	7	88	61	39
72	4	9	11	12	83	61	60	**322**	2	3	5	6	80	61	36
73	4	9	10	12	73	64	57	**323**	2	3	4	12	66	68	45
74	4	9	10	11	83	52	60	**324**	2	3	4	11	83	55	57
75	4	8	11	12	83	53	59	**325**	2	3	4	10	66	65	54
76	4	8	10	12	81	65	60	**326**	2	3	4	9	59	64	55
77	4	8	10	11	83	56	61	**327**	2	3	4	8	78	60	56
78	4	8	9	12	78	64	63	**328**	2	3	4	7	82	62	45
79	4	8	9	11	83	55	63	**329**	2	3	4	6	69	62	42
80	4	8	9	10	81	58	64	**330**	2	3	4	5	70	53	43
81	4	7	11	12	96	68	52	**331**	1	10	11	12	81	56	50
82	4	7	10	12	89	80	49	**332**	1	9	11	12	83	55	50
83	4	7	10	11	96	68	54	**333**	1	9	10	12	73	59	47
84	4	7	9	12	89	79	51	**334**	1	9	10	11	83	58	46
85	4	7	9	11	96	67	58	**335**	1	8	11	12	83	49	54
86	4	7	9	10	80	68	55	**336**	1	8	10	12	81	60	56
87	4	7	8	12	96	71	55	**337**	1	8	10	11	83	60	52
88	4	7	8	11	96	62	56	**338**	1	8	9	12	78	59	56
89	4	7	8	10	96	74	58	**339**	1	8	9	11	83	58	51
90	4	7	8	9	96	73	62	**340**	1	8	9	10	81	54	52
91	4	6	11	12	93	64	52	**341**	1	7	11	12	96	63	44
92	4	6	10	12	82	76	48	**342**	1	7	10	12	89	75	43
93	4	6	10	11	93	63	54	**343**	1	7	10	11	96	74	43

94	4	6	9	12	75	75	45	**344**	1	7	9	12	89	74	37
95	4	6	9	11	93	62	57	**345**	1	7	9	11	96	73	43
96	4	6	9	10	74	64	54	**346**	1	7	9	10	83	68	41
97	4	6	8	12	88	67	54	**347**	1	7	8	12	96	67	51
98	4	6	8	11	93	58	56	**348**	1	7	8	11	96	66	47
99	4	6	8	10	91	70	57	**349**	1	7	8	10	96	70	50
100	4	6	8	9	88	69	61	**350**	1	7	8	9	96	69	50
101	4	6	7	12	89	70	38	**351**	1	6	11	12	89	58	44
102	4	6	7	11	96	58	49	**352**	1	6	10	12	80	70	42
103	4	6	7	10	81	67	46	**353**	1	6	10	11	91	69	43
104	4	6	7	9	80	66	48	**354**	1	6	9	12	73	69	31
105	4	6	7	8	96	64	53	**355**	1	6	9	11	93	69	43
106	4	5	11	12	83	57	55	**356**	1	6	9	10	74	61	40
107	4	5	10	12	78	69	52	**357**	1	6	8	12	88	63	50
108	4	5	10	11	83	56	57	**358**	1	6	8	11	93	62	47
109	4	5	9	12	74	67	51	**359**	1	6	8	10	91	66	49
110	4	5	9	11	83	55	61	**360**	1	6	8	9	88	65	49
111	4	5	9	10	71	56	58	**361**	1	6	7	12	89	64	27
112	4	5	8	12	78	59	55	**362**	1	6	7	11	96	63	37
113	4	5	8	11	83	50	57	**363**	1	6	7	10	83	67	36
114	4	5	8	10	81	61	58	**364**	1	6	7	9	83	66	29
115	4	5	8	9	78	60	61	**365**	1	6	7	8	96	60	44
116	4	5	7	12	92	72	43	**366**	1	5	11	12	83	53	50
117	4	5	7	11	96	60	52	**367**	1	5	10	12	78	65	48
118	4	5	7	10	86	69	50	**368**	1	5	10	11	83	64	48
119	4	5	7	9	86	68	52	**369**	1	5	9	12	74	64	42

120	4	5	7	8	96	65	54	**370**	1	5	9	11	83	63	48
121	4	5	6	12	84	70	39	**371**	1	5	9	10	74	66	45
122	4	5	6	11	93	58	52	**372**	1	5	8	12	78	55	51
123	4	5	6	10	81	67	49	**373**	1	5	8	11	83	55	48
124	4	5	6	9	78	66	48	**374**	1	5	8	10	81	66	49
125	4	5	6	8	88	63	52	**375**	1	5	8	9	78	65	50
126	4	5	6	7	86	59	40	**376**	1	5	7	12	92	68	38
127	3	10	11	12	81	74	65	**377**	1	5	7	11	96	67	43
128	3	9	11	12	83	73	68	**378**	1	5	7	10	89	79	41
129	3	9	10	12	73	77	64	**379**	1	5	7	9	88	77	39
130	3	9	10	11	83	61	60	**380**	1	5	7	8	96	70	45
131	3	8	11	12	83	63	63	**381**	1	5	6	12	84	67	34
132	3	8	10	12	81	75	64	**382**	1	5	6	11	93	66	42
133	3	8	10	11	83	62	58	**383**	1	5	6	10	83	77	40
134	3	8	9	12	78	74	67	**384**	1	5	6	9	80	76	34
135	3	8	9	11	83	61	60	**385**	1	5	6	8	88	68	44
136	3	8	9	10	81	55	60	**386**	1	5	6	7	88	68	30
137	3	7	11	12	96	71	60	**387**	1	4	11	12	83	52	55
138	3	7	10	12	89	83	57	**388**	1	4	10	12	73	63	53
139	3	7	10	11	96	67	55	**389**	1	4	10	11	83	62	60
140	3	7	9	12	89	81	56	**390**	1	4	9	12	66	62	46
141	3	7	9	11	96	66	58	**391**	1	4	9	11	83	61	60
142	3	7	9	10	74	58	55	**392**	1	4	9	10	66	64	56
143	3	7	8	12	96	72	59	**393**	1	4	8	12	78	55	57
144	3	7	8	11	96	59	53	**394**	1	4	8	11	83	54	61
145	3	7	8	10	96	62	55	**395**	1	4	8	10	81	66	63

146	3	7	8	9	96	61	58	**396**	1	4	8	9	78	65	63
147	3	6	11	12	89	71	59	**397**	1	4	7	12	89	70	42
148	3	6	10	12	80	83	56	**398**	1	4	7	11	96	69	55
149	3	6	10	11	91	67	54	**399**	1	4	7	10	83	80	53
150	3	6	9	12	73	82	53	**400**	1	4	7	9	82	79	50
151	3	6	9	11	93	66	58	**401**	1	4	7	8	96	72	58
152	3	6	9	10	42	50	54	**402**	1	4	6	12	75	66	37
153	3	6	8	12	88	72	58	**403**	1	4	6	11	93	65	55
154	3	6	8	11	93	60	53	**404**	1	4	6	10	76	76	52
155	3	6	8	10	91	63	54	**405**	1	4	6	9	69	75	44
156	3	6	8	9	88	61	57	**406**	1	4	6	8	88	68	57
157	3	6	7	12	89	72	43	**407**	1	4	6	7	83	70	42
158	3	6	7	11	96	57	49	**408**	1	4	5	12	74	58	43
159	3	6	7	10	74	58	47	**409**	1	4	5	11	83	58	58
160	3	6	7	9	74	57	46	**410**	1	4	5	10	74	69	56
161	3	6	7	8	96	52	49	**411**	1	4	5	9	70	68	50
162	3	5	11	12	83	62	59	**412**	1	4	5	8	78	60	58
163	3	5	10	12	78	74	56	**413**	1	4	5	7	88	72	46
164	3	5	10	11	83	61	54	**414**	1	4	5	6	80	70	42
165	3	5	9	12	74	73	57	**415**	1	3	11	12	79	64	63
166	3	5	9	11	83	60	57	**416**	1	3	10	12	71	76	60
167	3	5	9	10	71	61	54	**417**	1	3	10	11	81	75	61
168	3	5	8	12	78	63	57	**418**	1	3	9	12	64	75	54
169	3	5	8	11	83	54	52	**419**	1	3	9	11	83	74	61
170	3	5	8	10	81	65	53	**420**	1	3	9	10	66	69	57
171	3	5	8	9	78	64	57	**421**	1	3	8	12	78	65	61

172	3	5	7	12	92	71	47	**422**	1	3	8	11	83	64	58
173	3	5	7	11	96	58	49	**423**	1	3	8	10	81	70	60
174	3	5	7	10	86	68	47	**424**	1	3	8	9	78	68	60
175	3	5	7	9	86	67	49	**425**	1	3	7	12	89	73	48
176	3	5	7	8	96	62	49	**426**	1	3	7	11	96	72	56
177	3	5	6	12	84	71	44	**427**	1	3	7	10	83	77	54
178	3	5	6	11	93	58	48	**428**	1	3	7	9	82	76	49
179	3	5	6	10	81	68	46	**429**	1	3	7	8	96	67	55
180	3	5	6	9	78	67	47	**430**	1	3	6	12	65	73	43
181	3	5	6	8	88	63	47	**431**	1	3	6	11	89	72	55
182	3	5	6	7	86	58	36	**432**	1	3	6	10	74	75	53
183	3	4	11	12	83	67	61	**433**	1	3	6	9	67	74	46
184	3	4	10	12	73	78	58	**434**	1	3	6	8	88	67	54
185	3	4	10	11	83	66	63	**435**	1	3	6	7	83	67	40
186	3	4	9	12	66	77	58	**436**	1	3	5	12	74	64	49
187	3	4	9	11	83	65	67	**437**	1	3	5	11	83	63	55
188	3	4	9	10	64	66	63	**438**	1	3	5	10	74	75	53
189	3	4	8	12	78	67	59	**439**	1	3	5	9	70	73	50
190	3	4	8	11	83	58	61	**440**	1	3	5	8	78	64	53
191	3	4	8	10	81	70	63	**441**	1	3	5	7	88	71	44
192	3	4	8	9	78	69	66	**442**	1	3	5	6	80	72	41
193	3	4	7	12	89	75	48	**443**	1	3	4	12	66	68	50
194	3	4	7	11	96	63	58	**444**	1	3	4	11	83	67	64
195	3	4	7	10	80	72	56	**445**	1	3	4	10	66	78	61
196	3	4	7	9	80	71	58	**446**	1	3	4	9	59	77	58
197	3	4	7	8	96	67	58	**447**	1	3	4	8	78	69	62

198	3	4	6	12	75	75	46	**448**	1	3	4	7	82	75	52
199	3	4	6	11	93	63	58	**449**	1	3	4	6	69	76	49
200	3	4	6	10	74	72	55	**450**	1	3	4	5	70	67	50
201	3	4	6	9	67	71	56	**451**	1	2	11	12	79	59	46
202	3	4	6	8	88	68	57	**452**	1	2	10	12	71	71	45
203	3	4	6	7	80	62	45	**453**	1	2	10	11	81	70	48
204	3	4	5	12	74	66	46	**454**	1	2	9	12	64	71	33
205	3	4	5	11	83	54	57	**455**	1	2	9	11	83	70	47
206	3	4	5	10	71	63	54	**456**	1	2	9	10	66	73	44
207	3	4	5	9	68	62	56	**457**	1	2	8	12	78	63	51
208	3	4	5	8	78	58	56	**458**	1	2	8	11	83	62	49
209	3	4	5	7	86	61	46	**459**	1	2	8	10	81	74	51
210	3	4	5	6	78	61	43	**460**	1	2	8	9	78	72	51
211	2	10	11	12	81	70	49	**461**	1	2	7	12	89	73	30
212	2	9	11	12	83	70	52	**462**	1	2	7	11	96	72	42
213	2	9	10	12	73	73	49	**463**	1	2	7	10	83	84	41
214	2	9	10	11	83	59	47	**464**	1	2	7	9	82	83	34
215	2	8	11	12	83	61	52	**465**	1	2	7	8	96	75	47
216	2	8	10	12	81	73	53	**466**	1	2	6	12	65	72	22
217	2	8	10	11	83	62	49	**467**	1	2	6	11	89	71	41
218	2	8	9	12	78	71	56	**468**	1	2	6	10	74	83	40
219	2	8	9	11	83	60	51	**469**	1	2	6	9	67	82	27
220	2	8	9	10	81	64	51	**470**	1	2	6	8	88	75	45
221	2	7	11	12	96	72	44	**471**	1	2	6	7	83	73	25
222	2	7	10	12	89	84	42	**472**	1	2	5	12	74	64	34
223	2	7	10	11	96	69	41	**473**	1	2	5	11	83	63	45

224	2	7	9	12	89	83	39	**474**	1	2	5	10	74	75	43
225	2	7	9	11	96	69	45	**475**	1	2	5	9	70	74	37
226	2	7	9	10	83	71	42	**476**	1	2	5	8	78	65	46
227	2	7	8	12	96	74	49	**477**	1	2	5	7	88	74	33
228	2	7	8	11	96	63	44	**478**	1	2	5	6	80	75	28
229	2	7	8	10	96	75	46	**479**	1	2	4	12	66	62	38
230	2	7	8	9	96	73	50	**480**	1	2	4	11	83	61	56
231	2	6	11	12	89	70	43	**481**	1	2	4	10	66	72	53
232	2	6	10	12	80	83	41	**482**	1	2	4	9	59	71	45
233	2	6	10	11	91	68	41	**483**	1	2	4	8	78	63	56
234	2	6	9	12	73	82	33	**484**	1	2	4	7	82	73	43
235	2	6	9	11	93	68	44	**485**	1	2	4	6	69	73	36
236	2	6	9	10	76	70	41	**486**	1	2	4	5	70	63	41
237	2	6	8	12	88	74	47	**487**	1	2	3	12	55	68	44
238	2	6	8	11	93	63	44	**488**	1	2	3	11	79	68	58
239	2	6	8	10	91	75	45	**489**	1	2	3	10	64	79	55
240	2	6	8	9	88	73	48	**490**	1	2	3	9	57	78	48
241	2	6	7	12	89	73	26	**491**	1	2	3	8	78	69	56
242	2	6	7	11	96	59	35	**492**	1	2	3	7	82	76	42
243	2	6	7	10	83	69	33	**493**	1	2	3	6	58	76	37
244	2	6	7	9	83	69	31	**494**	1	2	3	5	70	67	43
245	2	6	7	8	96	65	41	**495**	1	2	3	4	59	68	49
246	2	5	11	12	83	63	47								
247	2	5	10	12	78	75	44								
248	2	5	10	11	83	62	44								
249	2	5	9	12	74	73	43								

250	2	5	9	11	83	61	47								