FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

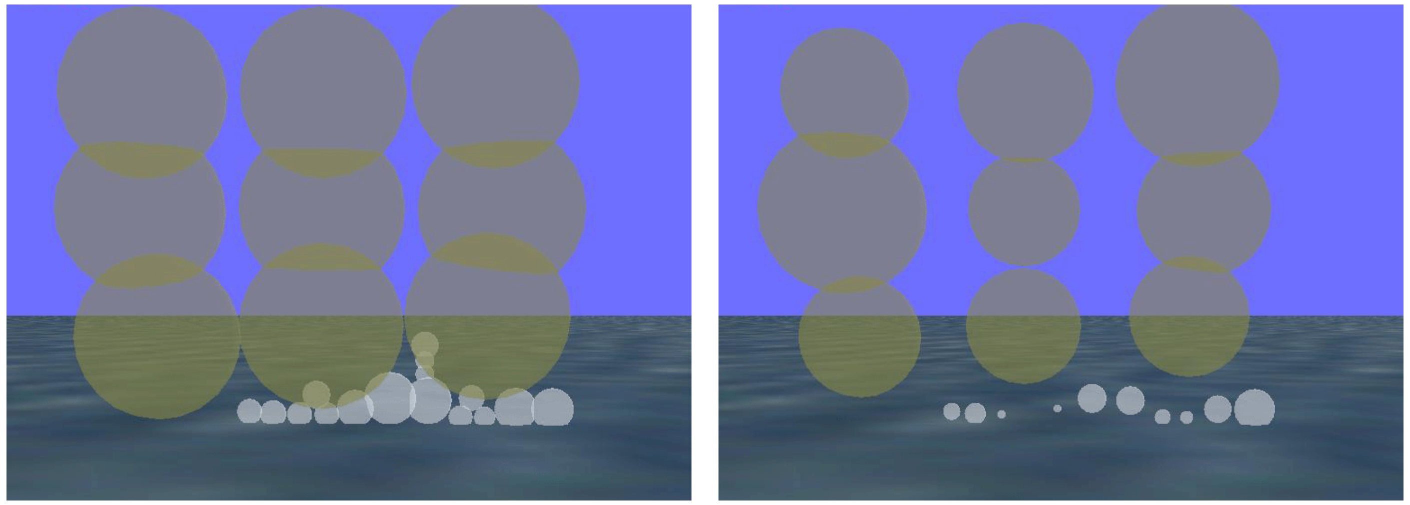
FOI
Defence Research Agency
Information Systems
P.O. Box 1165
SE-581 11 Linköping

Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

Removal of Obscured Surfaces in a Computer Model

LARS BERGLUND



FOI-R--2690--SE Methodology report **Information Systems**
ISSN 1650-1942 December 2008

Lars Berglund

Removal of Obscured Surfaces in a Computer Model

Titel	Borttagning av skymda ytor i en datormodell
Title	Removal of Obscured Surfaces in a Computer Model
Rapportnr/Report no	FOI-R--2690--SE
Rapporttyp Report Type	Metodrapport Methodology report
Sidor/Pages	33 p
Månad/Month	December
Utgivningsår/Year	2008
ISSN	ISSN 1650-1942
Kund/Customer	FM
Forskningsområde Programme area	6. Telekrig och vilseledning 6. Electronic Warfare
Delområde Subcategory	61 Telekrigföring med EM-vapen och skydd 61 Electronic Warfare including Electromagnetic Weapons and Protection
Projektnr/Project no	E7121, E7467
Godkänd av/Approved by	Mikael Sjöman

FOI, Totalförsvarets Forskningsinstitut
Avdelningen för Informationssystem
Box 1165
581 11 Linköping

FOI, Swedish Defence Research Agency
Information Systems
Box 1165
SE-581 11 Linköping

Summary

This report describes how spheres can be used to represent ships and decoys, such as flares and chaff clouds, as extended targets in simulations. The report also describes a detailed method for calculating how much, of each of the sphere used in the simulation scenario, is obscured by other spheres.

The described method gives a better representation of extended targets than point sources as well as better (more realistic) outcome of the simulations involving IR or radar seeking missiles, ships and decoys.

Keywords: ship, decoy, radar, IR, flare chaff, obscuring, spheres

Sammanfattning

Denna rapport beskriver hur sfärer kan utnyttjas i simuleringar för att representera utbredda mål som fartyg och skenmål. Med skenmål avses här facklor samt remsmoln. Rapporten redovisar en metod för beräkning av hur mycket av varje sfär som inte är dold av en annan sfär.

Den beskrivna metoden skapar möjligheter till en mer realisisk representation än användning av punktmål i simuleringar av duellen mellan målsökande missil (radar eller IR) och motmedelsskyddat fartyg.

Nyckelord: fartyg, skenmål, radar, IR, facklor, remsor, döljande, sfärer

Innehållsförteckning

1 Introduction

In the FOI project Technical Threat System Analysis a number of models have been produced to support the assessment of the duels between missiles and platforms protected by countermeasures. These models are used in simulations as a base for better understanding of the technical protection of Swedish platforms and also as a way to better develop the tactics used in combat [1].

In the simulation of the duel between anti-ship missiles and ship which can deploy decoys, the ship and decoys normally have been regarded as point sources. A point source is very unlikely to cover another point source. So there is normally no need to handle obscuration in a point source model. In a point source model the behaviour is rather digitally. Either an object is within the field of view, projected on the detector (the IR case) or not. In the radar case the objects are either in the search lobe or not.

In order to get result more realistic, the targets (ship and decoys) needs to be represented as extended targets. Therefore some work has been done in order to see how ship and decoys can be represented to describe extended targets. Moving to extended targets will make the situation become less and less digitally. In a previous report [2], a description is given how ship and decoys in the infrared area could be described as spheres, and their projective areas as circles. It would therefore not be far fetched to use the same conscript for the radar models.

However, the extended targets are bound to be obscuring each other (one way or another) so the problem to be faced is the handling of obscured surfaces. Thus, being the price to pay for extended targets in simulations.

Previous work that has been done within the field of obscuring parts has been described for triangles [3]. Yet, another report describes how to handle point sources and whether they are obscured by an extended target or not [4]. Since the chosen method described in this report is based on spheres neither of the previous work are applicable.

The method presented in this report is based upon the fact that the spheres are to be considered opaque. The reason for this being that it keeps the calculations simple, it also makes it possible to use and keep the vector structure which the simulation environment requires.

1.1 Acknowledges

The author of this report would like to thank the colleague Patrik Strand for his contribution and help with images, made with the software AFE [5], for this report.

2 The sphere model of a ship

The point source model of a ship is represented by a point with its coordinates. A table containing azimuth angles and the corresponding signature value (intensity values (IR) or radar cross section (RCS) values (radar)) are used in order to get the value that represents the current view. Normally, interpolation is used to get values not given in the table.

When making an extended model of the ship, considerations have to be taken to aspect (azimuth angle), signature value and projected area.

2.1 IR model

In the sphere model used for IR a ship is represented by a number of spheres. Depending on azimuth angle the total projected area of the spheres corresponds to the total projected area of the ship and the total intensity. This means that the radius for the spheres describing the ship will vary for different aspects, in some cases even disappear and new spheres will appear. This is described in more detail in [2]. An example is given in figure 2.1. Note that some of the spheres describing the ship could be obscuring others spheres as shown in the figures below. In the model no sphere describing the ship is obscuring another sphere. This means that the total sum of the projected spheres gives the correct projected area (even if some spheres may be overlapping others).

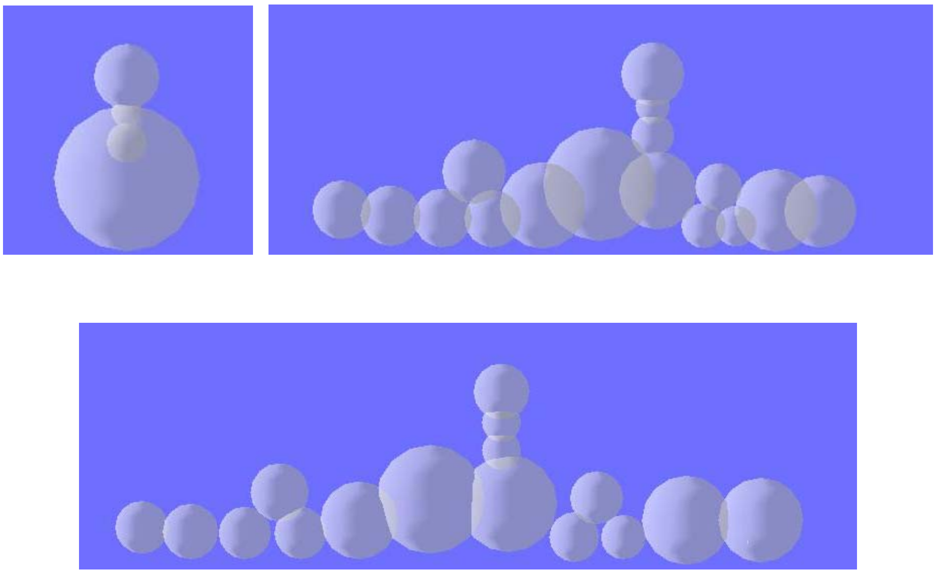


Figure 2.1 A sphere model of a ship seen from the aspect angles 0°, -45° and -90°. Images generated in AFE [5]

The use of spheres in the IR model, gives a more detailed and accurate result without taking to much CPU time when it comes to simulation.

2.2 Radar model

The complexity differs some between the IR and radar case. In the radar case the ship could be regarded as a number of reflectors placed at different locations along the ship. The major problem will be to tell the true distributions of the reflectors. Measurement to get the RCS values as a function of azimuth angle gives the total RCS not the distribution. The issue how to get the distribution is currently under discussion and will eventually be solved in the future.

In the mean time, the sphere model used in the IR is used even in the radar case, because it represents (at least to some extent) the size and geometry of the ship.

A first step would be to use the total RCS for a given azimuth angle and conformably distribute this to the projected circles according to their area. This is of course a coarse or even rude simplification, but it will still be a better approximation than a point source. The first step also prepares the models for step two.

Step two is to modify the distribution of RCS to the different spheres. This step can however not be achieved until the distribution is determined.

3 The sphere model of a decoy

In this report the decoys discussed are either flares or chaff clouds. Whether the decoy is a flare or a chaff cloud they are represented by a sphere.

3.1 Flare model

A flare is described by a sphere with its origin in a given position. The radius and intensity of the sphere will vary as a function of time as shown in the figure below.

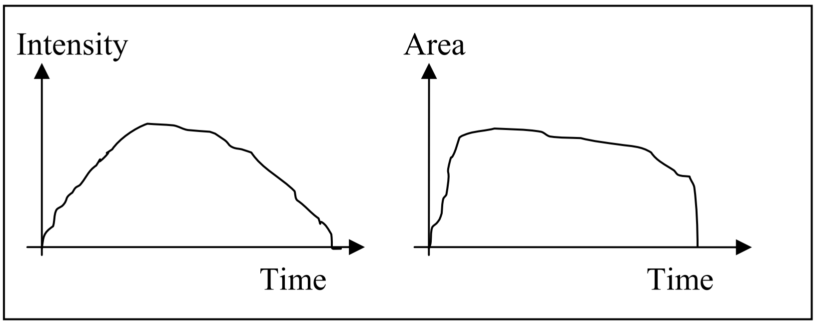


Figure 3.1 Intensity and area for a flare decoy.

The behaviour of a flare is that after burst it is drifting with the wind and descending towards the sea surface. The flare is to be considered as non transparent since it has a screening effect.

3.2 Chaff cloud model

Like the flare a chaff cloud is described with a sphere. The RCS and area will vary as a function of time. But in difference from the flare a chaff cloud has a long lifetime (relative to simulated event of a missile attack). The descending rate for a chaff cloud is compared to the descending rate of a flare very slow (descending rate numbers in cm/s).

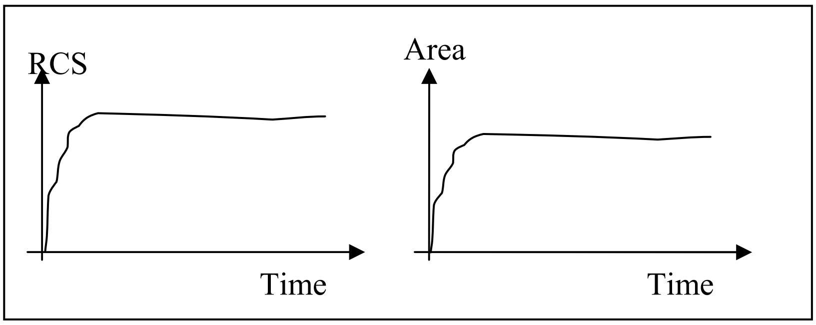


Figure 3.2 RCS and area for a chaff cloud.

The radar cross section of the chaff cloud is depending on number off chaff in the cloud [6]. If a chaff cloud consists of N chaffs, where the cloud has the total projected area A_0 and the RCS of each dipole is σ_{dipole}. Then equation 3.1 could be applied.

$$\sigma = A_0\left(1 - e^{-\frac{N}{A_0}\sigma_{dipole}}\right) \qquad \textit{Eq. 3.1}$$

The equation above gives the chaff clouds total radar cross section. If the cloud has low density of chaffs the expression for the RCS will be:

$$\sigma_{chaffcloud} = N\sigma_{dipole} \qquad \textit{Eq. 3.2}$$

where

$$\sigma_{dipole} = 0.15\lambda^2 \qquad \textit{Eq. 3.3}$$

If the cloud has high density the expression for the RCS will be

$$\sigma_{chaffcloud} = A_0 \qquad \textit{Eq. 3.4}$$

In this model the chaff density is regarded as high and equation 3.4 will be used. This means that the situation is similar to the one illustrated in figure 3.3.

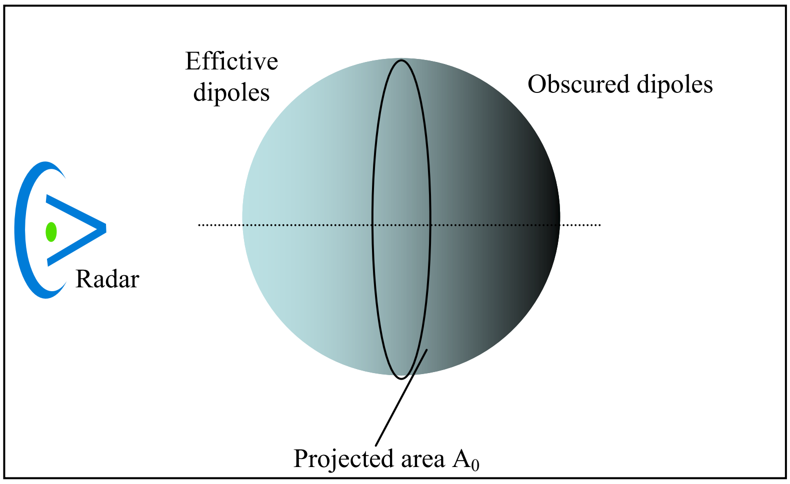


Figure 3.3 High density chaff cloud. Since the chaff density is high, only the chaffs in the half sphere closest to the radar will give a reflection. The other half sphere (dark) does not tribute because the chaff are obscured.

4 Simulation scenario

The simulation scenario contains a ship, a variable number of decoys (either flares or chaff clouds) and a missile, see one example figure 4.1.

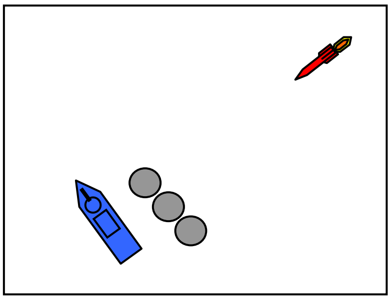


Figure 4.1 Scenario with ship, decoys and missile.

In the beginning of the simulation when the seeker opens up there is only the ship. As mentioned in chapter 2.1, no consideration is taken to the fact that some of the spheres describing the ship may actually be obscuring other spheres. During the simulation the ship will launch a number of decoys, each represented by a sphere.

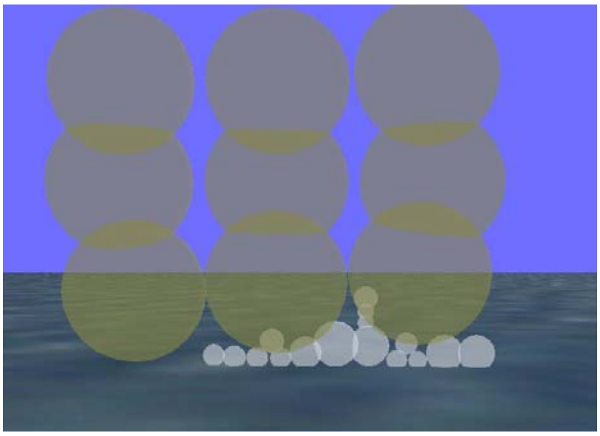


Figure 4.2 Scenario with ship, decoys (chaff clouds) seen from missile, where all sphere are made transparent to illustrate obscuring effects. Note that the chaff clouds obscure each other and that some of spheres describing the ship are completely obscured. Image generated in AFE [5].

Now the effect of obscuring parts has to be taking into account to generate the right input to seeker model. This is done by first sorting all the spheres with consideration to distance to the seeker. One example of quick sorting algorithm to use (written for MatLab) is given in appendix 2. Further information of quick sorting algorithms is given in [6] and [7].

When the sorting has been done, it is time to check and calculate how much of each sphere is visible from the seeker point of view. In this process the spheres are regarded as circles (which are the projected area of the spheres). To be able to do this the following steps needs to be addressed:

- Calculation of overlap with two circles
- Calculations of overlap with several circles
- Representation of the spheres after calculations

The above given steps are discussed in the following chapters.

5 Calculation of overlap with two circles

The basic problem that has to be solved in the model is when there are two circles and one is overlapping the other. Several combinations can be found. In order to simplify the description of the cases the lower circle (marked red in figure 5.1) is partly or completely covered by the upper circle (marked blue in figure 5.1). This also means that when calculating the projection of a sphere on the detector plane or in a range gate, a third coordinate is needed to determine the order which of the circles is the upper respectively the lower circle.

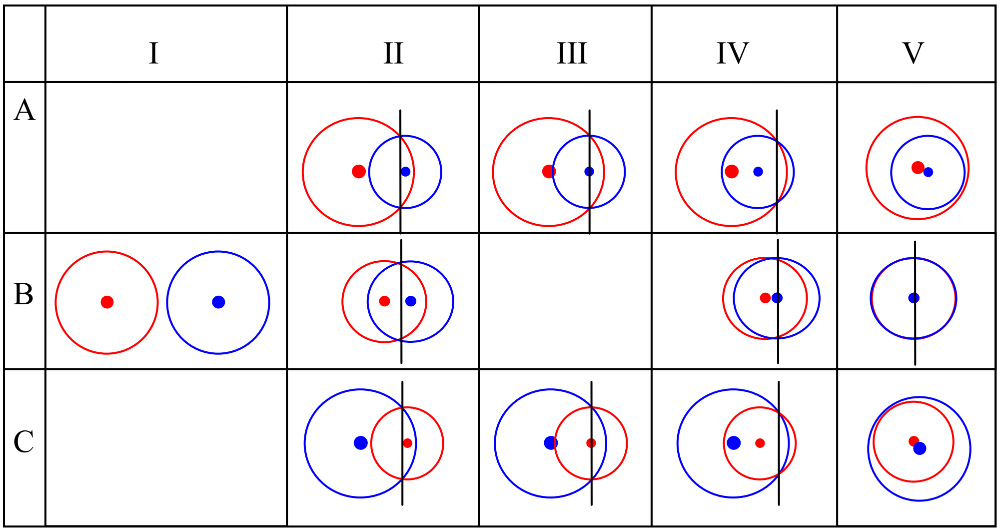


Figure 5.1 *Possible combinations of two circles in regards to overlap, where blue circle is in front of red circle. In row A circle blue is smaller then circle red. In row B the circles have the same size. In row C blue circle is larger then circle red.*

I. The two circles do not overlap each other.

II. Overlap, the centres of the circles are on each side of the intersection line (marked black).

III. Overlap, the centres of one of the circles is on the intersection line.

IV. Overlap, the centres of the circles are on the same side of the intersection line.

V. Complete overlap, the upper circle is completely covering the lower circle or the upper circle covering the lower circle but lower circle is larger then the upper circle.

In order to determine the overlap for two circles following are needed:

- The coordinates for the centres of each circle (2 coordinates)
- One coordinate that can be used to determine their relationship in distance in order to determine which is the upper circle and which is the lower
- The radius of each circle

5.1 Calculation of a circle segment

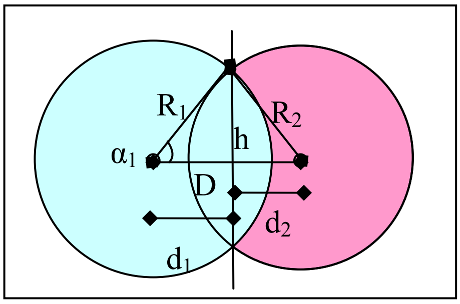


Figure 5.2 Overlap and definition of distance D. The distance D is the distance between the two circles origin. R is the radius for the circles. Index 1 indicates the upper blue circle, index 2 indicates the lower red circle. The distance d is the distance between the centre of a circle and the line that cuts through the intersection points and h is the half the length of the line between the two intersection points.

As can be seen in figure 5.2 there are four variables that have to be determined. The first is the distance D which is the distance between the origins of the circles. The distance d_1 and d_2 representing the distance between the origin of the circles to the intersection line. The height h is half the length of the line between the two interception points. Since the coordinates for the circles origins are given, the distance D can be expressed as:

$$D = \sqrt{(x_2^2 - x_1^2) + (y_2^2 - y_1^2)} \qquad \textit{Eq 5:1}$$

An expression for the distance d_1 can be expressed as:

$$d_1 = R_1 \cos(\alpha_1) \qquad \textit{Eq 5:2}$$

The expression for the law of cosines [7] will give, with parameters in figure 5.2, the following:

$$\cos(\alpha_1) = \frac{R_1^2 + D^2 - R_2^2}{2DR_1} \qquad \textit{Eq 5:3}$$

Using the equation 5:3 into 5:2 then gives:

$$d_1 = R_1 \cos(\alpha_1) = \frac{R_1^2 + D^2 - R_2^2}{2D} \qquad \textit{Eq 5.4}$$

Since d_1 is found is easy to find d_2.

$$d_2 = D - d_1 \qquad \textit{Eq 5:5}$$

For each of the circles the area of the part lying on the other side of the interception line (seen from the origin) needs to be calculated. Since the following equations can be used for both circles the index 1 and 2 is changed to index i. First find height h according to eq. 5.6. Note that the height h will be same for both circles. So it does not need the index.

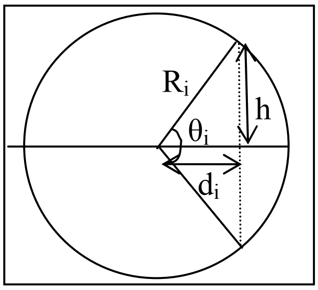


Figure 5.3 Definitions of parameters.

$$h = \sqrt{R_i^2 - d_i^2} \qquad \text{Eq 5:6}$$

Now the cone (the triangular part in figure 5.3) can be calculated as:

$$A_{cone\_i} = d_i \cdot h \qquad \text{Eq 5:7}$$

Find the angle θ, according to:

$$\theta_i = 2\arcsin\left(\frac{h}{R_i}\right) \qquad \text{Eq 5:8}$$

With the angle given the circle sector can be found as:

$$A_{\sec tor\_i} = \frac{R_i^2 \theta_i}{2} \qquad \text{Eq 5:9}$$

Finally, the area of the searched segment can be found as:

$$A_{segment\_i} = A_{\sec tor\_i} - A_{cone\_i} \qquad \text{Eq 5:10}$$

The above shown equations are what are needed to calculate the hidden area. However, consideration has to be taken into account on which situation there is (see figure 5.1).

5.2 Determine the situation

The first check that has to be done is to check if the two circles are overlapping each other. This is achieved by using following relationship.

$$D \geq R_1 + R_2 \qquad \text{Eq. 5:11}$$

If the condition in eq. 5:11 is not met the circles are overlapping somehow. Next step will be to determine if one of the circles is completely inside the other circle (case V in figure 5.1). This can be done by checking the conditions given in equations 5:12 and 5:13.

$$R_1 \geq D + R_2 \qquad \text{Eq. 5:12}$$

If true the lower circle is completely covered by the upper circle, which yields that the contribution from the lower circle will be zero.

$$R_2 \geq D + R_1 \qquad \text{Eq. 5:13}$$

If true the upper circle is completely inside the lower circle, which yields that the contribution from the lower circle will be:

$$f = \frac{\pi(R_2^2 - R_1^2)}{\pi R_2^2} \qquad \text{Eq. 5:14}$$

In equation 5:14 the variable f is the factor which tells how much of the lower circle that is visible. If none of the three conditions above is true the two circles is overlapping according to figure 5.1 combination II, III and IV.

5.2.1 Origins of the circles are on opposite sides of the intersection line

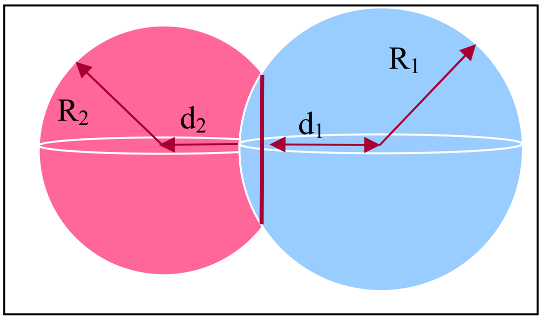


Figure 5.4 Example of situation, compare with figure 5.1, column II.

In this situation, it does not matter which of the circles is the largest the hidden area of the red circle in figure 5.4 can be calculated as:

$$A_{hidden} = A_{segment\_1} + A_{segment\_2} \quad \text{Eq. 5:15}$$

Note that the both of the parameters d_1 and d_2 have positive values greater than zero.

5.2.2 Origin of one of the circles is on the intersection line

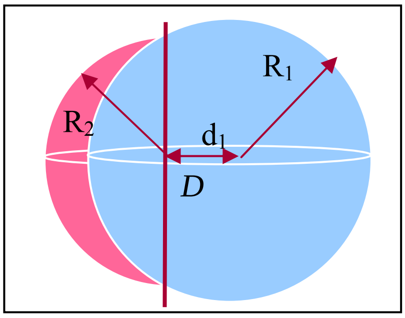


Figure 5.5 Example of situation, compare with figure 5.1, column III.

In this situation, it does not matter which of the circles is the largest. The hidden area of the red circle in figure 5.4 can be calculated as given in eq. 5:15. Note that one of the parameters d_1 or d_2 has the value zero while the other has a positive value greater then zero.

5.2.3 Origins of the circles are on the same side of the intersection line

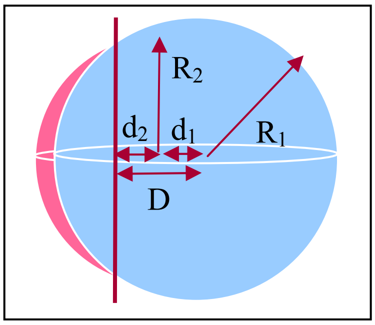


Figure 5.6 Example of situation, compare with figure 5.1, column IV.

In this situation, it does however matter which of the circles is the largest. If the upper circle (blue in figure, $R_1 \geq R_2$) has a larger or equal area to that of the lower circle (red in figure) then the hidden area of the red circle in figure 5.6 can be calculated as given in eq. 5:16.

$$A_{hidden} = \pi R_2^2 - (A_{segment\_2} - A_{segment\_1}) \qquad \textit{Eq. 5:16}$$

If the lower circle has a larger radius ($R_2 > R_1$) the hidden area will instead be given by eq 5:17.

$$A_{hidden} = \pi R_1^2 - (A_{segment\_1} - A_{segment\_2}) \qquad \textit{Eq. 5:17}$$

Not that one of the parameters d_1 or d_2 will have a value that is negative while the other will have a positive value greater then zero.

5.2.4 Rules for calculations of overlap

To sum this chapter up, the following rules and consequences can be set:

$D > R_1 + R_2$ No overlap and thus

$A_{hidden} = 0$

$R_1 > D + R_2$ Upper circle overlaps the other circle completely and thus

$A_{hidden} = \pi R_2^2$

$R_2 > R_1 + D$ Upper circle is inside the lower circle completely and thus

$A_{hidden} = \pi R_1^2$

If there is an overlap and the above relations are false, then the equation to be used is depending on the sign of the expression d_1 times d_2 and the relation between R_1 and R_2, according to table 5.1.

Sign($d_1 \cdot d_2$)	**$R_1 \geq R_2$**	**$R_1 < R_2$**
1, 0	$A_{hidden} = A_{seqment\_1} + A_{seqment\_2}$	$A_{hidden} = A_{seqment\_1} + A_{seqment\_2}$
-1	$A_{hidden} = \pi R_2^2 + A_{seqment\_1} - A_{seqment\_2}$	$A_{hidden} = \pi R_1^2 + A_{seqment\_2} - A_{seqment\_1}$ -

Table 5.1 Equation to use for calculation of hidden area depending on the relations between the radius and sign of distance to intersection line.

What is needed is the visible area, which can be calculated as follows.

$$A_{visible} = \pi R_2^2 - A_{hidden} \qquad \textit{Eq. 5:18}$$

Furthermore, an expression for the normalized area will be needed.

$$f = \frac{A_{visible}}{\pi R_2^2} \qquad \textit{Eq. 5:19}$$

6 Calculation of overlap with several circles

If there are only two spheres the problem can easily be solved as proven in previous chapter. This is not likely to be the case. The situation could contain several spheres which will overlap each other. Two examples are shown in figure 6.1.

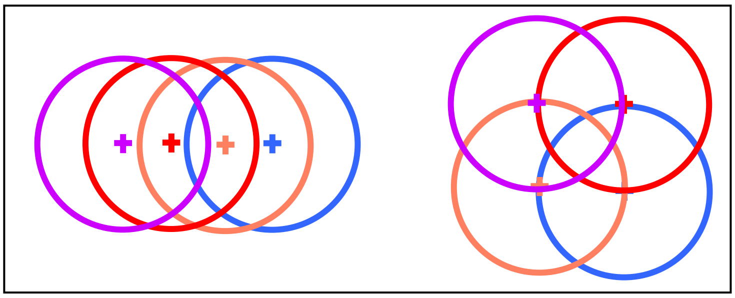


Figure 6.1 Examples with four circles overlapping each other. The order from top is purple, red orange and blue.

There ought to be an analytical solution but the calculations to achieve the solution will increase with number of circles. So another way has to be taken in order to keep the simulation time as short as possible.

The principal solution to solve this problem will be to use the result from previous chapter. Treat the situation by looking at the spheres two and two. This means that the spheres have to be sorted according to distance as step one.

For step two there are two possible alternatives, top down or bottom up. But before going through step two, step three has to be explained. In step three two spheres are compared according to previous chapter. If as shown in figure below blue circle overlaps the red, the visible area of the red circle is calculated.

A new radius is calculated for the red circle according to equation 6:1.

$$R_{2\_new} = \sqrt{fR_2^2} \qquad \textit{Eq. 6:1}$$

The question now becomes where to position the resized circle, considering visible area and to some extent the form (of two circles). Since the resized circle no longer should be obscured (by the other circle) the circle is placed where the circle is covered as little as possible. This means that the reduced circle should be moved along the line, going through the origins of the two circles, according to figure 6.1.

Since a right orthogonal coordinate system is used the angle (φ) for the vector from centre of blue circle to the centre of the red circle can be calculated. New coordinates are calculated for the red circle as described in equation 6:2.

$$\begin{cases} x_{2\_new} = x_2 + (R_2 - R_{2\_new})\cos(\varphi) \\ y_{2\_new} = y_2 + (R_2 - R_{2\_new})\sin(\varphi) \end{cases} \qquad \textit{Eq. 6:2}$$

The results of these operations are shown in figure 6.1.

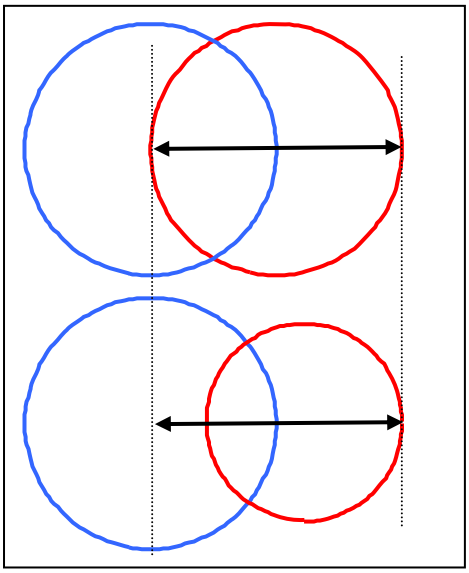


Figure 6.1 Illustration of reducing area (radius) and change position of partly covered circle.

6.1 Top down approach

In the top down approach, the closest circle (to the seeker) is tested against the second circle. If the circles overlap, the second circle gets a new radius and a new position. Then the first circle is tested against the third circle, which gets a new radius and a new position if there is an overlap. The first circle is tested against all the circles further down the list. In figure 6.2 this step is illustrated by going from A to B. In the same manner the new circle 2 is tested against those behind. In figure 6.2, circle 2 only effects circle 4 (step B to C). Finally, circle 3 is tested against circle 4, which will get a new radius and a new position (step C to D in figure). This means that all circles are tested against all the other circles that are further away from the seeker.

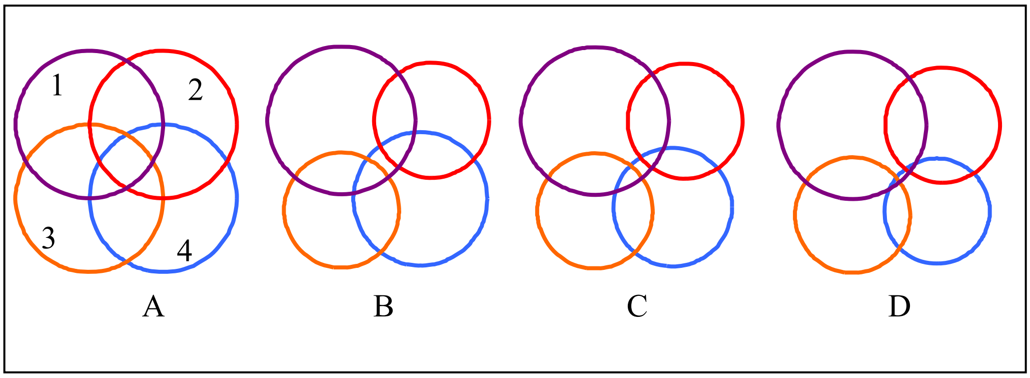


Figure 6.2 Example of top down approach with four circles.

Step A to B. Circle 1 is tested against all the others, which all gets new radius and new positions.
Step B to C. Circle 2 is tested against all others, but only circle 4 is affected.
Step C to D. Circle 3 is tested against circle 4, which is affected.

6.2 Bottom up approach

In the bottom up approach, the circle furthest away is tested against the circle closest in distance. If the circles overlap, the start circle gets a new radius and a new position. Then it is tested against the third circle (note that the first circle now has new radius and position). The start circle is tested against all the other circles in the list.

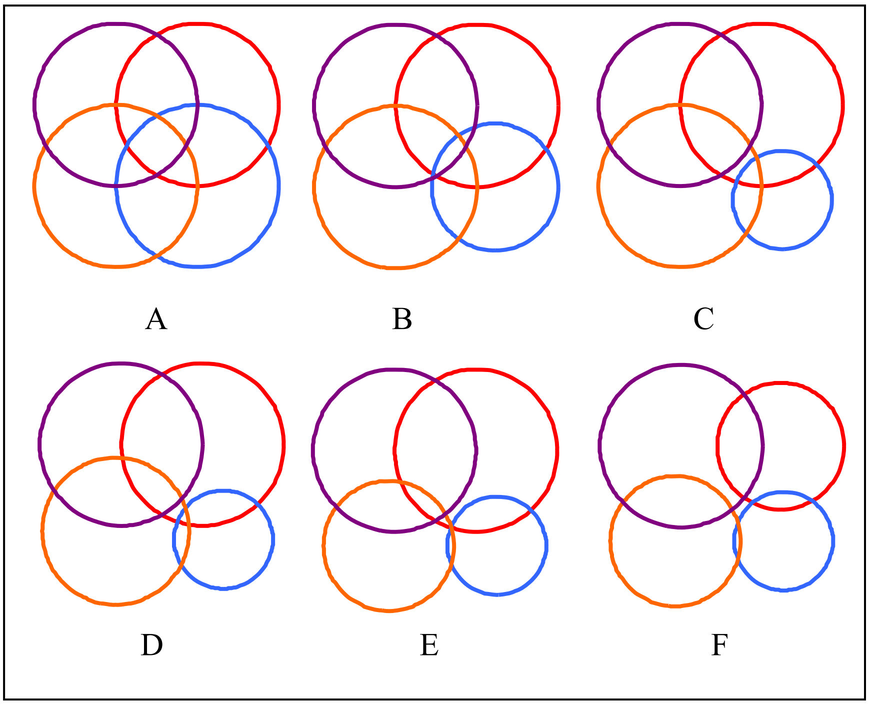


Figure 6.3 Example of bottom up approach with four circles.

Step A to B. Circle 4 is tested against circle3, circle 4 gets new radius and position.
Step B to C. Circle 4 is tested against circle2 circle 4 gets new radius and position. Circle 4 is not effected by circle 1.
Step C to D. Circle 3 is tested against circle 2, circle 3 gets new radius and position.
Step D to E. Circle 3 is tested against circle 1, circle 3 gets new radius and position.
Step E to F. Circle 2 is tested against circle 1, circle 2 gets new radius and position.

6.3 Determination of method for optimal result

Two methods similar to each other, which one is to be used? In order to answer that question, a couple of situations have to be tested. The situations that were tested are illustrated in figure 6.4.

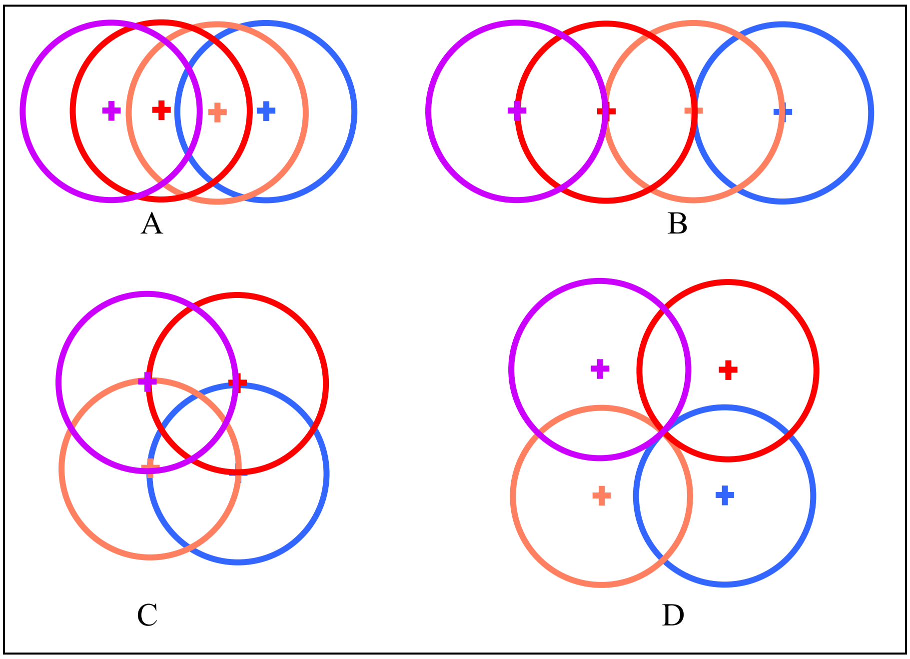


Figure 6.4 Test situations.

During the test the total visible area where calculated for the two methods top-down or bottom up. The situations were also generated in an image. This means that first the circle furthest away was drawn with the value 1 for each picture element, the next circle was drawn with value 2 and added to the image. Circle 2 was drawn with value 4 and added to the image. Finally the circle 1 (closest) was drawn with value 8 and added to the image. Resulting image is shown I figure 6.5.

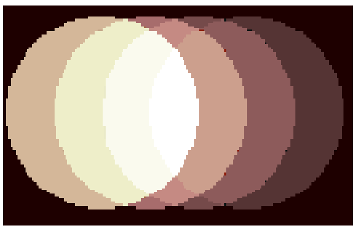


Figure 6.5 Example of image for test calculations.

In the resulting image each picture element has a value ranging from 0 to 15, where 0 indicates background and values greater then 0 indicates that one, two, three or four circles cover the picture element accordingly to table 6.1.

Pixel-value	Contents in fig
0	Background
1	Circle 4
2	Circle 3
3	Circle 3 & 4
4	Circle 2
5	Circle 2 & 4
6	Circle 2 & 3
7	Circle 2 & 3 & 4
8	Circle 1
9	Circle 1 & 4
10	Circle 1 & 3
11	Circle 1 & 3& 4
12	Circle 1 & 2
13	Circle 1 & 2 & 4
14	Circle 1 & 2 & 3
15	Circle 1 & 2 & 3 & 4

Table 6.1 *Pixel values in image and what the pixel values represent. Visible area for circle 4 is represented by pixel value 1. Visible area of circle 3 is represented by pixel values 2 and 3. Visible area of circle 2 is represented by pixel values 4 to 7. Visible area of circle 1 is represented by pixel values greater than 7.*

By running a histogram on the image a resulting vector h is given where the number of pixels is given as a function of the values in the images. Since the interest is how much of the different circles are visible it is rather easily achieved. For instance, the value f for circle 4 is

$$f_{circle4} = \frac{h(1)}{h(1)+h(3)+h(5)+h(7)+h(9)+h(11)+h(13)+h(15)} \qquad \textit{Eq. 6.3}$$

By comparing the figures from the different methods with those from received from the images some conclusions can be drawn. In table 6.2 the result from the four tests are presented.

		Calculated circles				Circle area from image				
Test	Method	1	2	3	4	1	2	3	4	Deviation [%]
A	TD	1.0	0.36	0.36	0.37	1.0	0.36	0.36	0.360	0.6
	BU	1.0	0.36	0.36	0.36	1.0	0.36	0.36	0.36	-0.2
B	TD	1.0	0.61	0.64	0.33	1.0	0.61	0.61	0.31	1.9
	BU	1.0	0.61	0.61	0.32	1.0	0.61	0.61	0.31	0.1
C	TD	1.0	0.61	0.61	0.51	1.0	0.61	0.57	0.36	7.5
	BU	1.0	0.61	0.52	0.37	1.0	0.61	0.57	0.36	-1.3
D	TD	1.0	8.2	8.2	0.76	1.0	0.82	0.82	0.64	3.6
	BU	1.0	0.82	0.82	0.68	1.0	0.82	0.82	0.64	1.3

Table 6.2 *Results, the result from the images is treated as the true result. For the two methods tested TD (top down) and BU (bottom up) the visible area is divided by that from the true result. This means that a minus result indicates that the calculated method yields a total area which is less than the true area.*

The total visible area as a function of method is shown below in the table 6.3.

Test situation	**Top-down**	**Bottom-Up**
A	0.6 %	-0.2 %
B	1.9 %	0.1 %
C	7,5 %	-1, 3 %
D	3,6 %	1,3 %

Table 6.3 *Results, copied from table 6.2.*

This means that the bottom-up method should be used, since it gives the least deviation from the true result. The error when using the bottom up method is less than 2%, which is good enough. Note that this applies for the area, if consideration has to be taken to different intensity or RCS for the spheres the error might reach higher values.

7 Representation of the spheres after calculations

When the visible area has been calculated for each circle, using the method bottom-up, the question to be answered is how a sphere should be represented. There are a number of possible ways, all with there plus and minus. The three different methods are:

- Change radius and position
- Change radius but keep “old” position
- Change intensity but keep old position and radius

In order to determine the “best” representation, the different methods were compared with each other. Two circles were given coordinates and radius. The distance along the x-axis between the two origins was also given. The calculation of visible area of the partly obscured circle was calculated according to chapter 4. Then the different methods were tested by drawing the circles in an image according to the calculated result from the different methods.

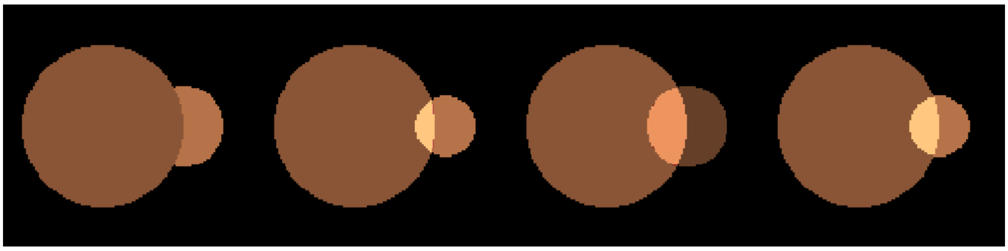


Figure 7.1 Example of image for two circles, with radius 40 and 20 units. Distance between the origins is 40 units. Intensity levels used are 75 for the larger circle and 100 for the smaller circle. From left to right in the figure, true, new radius and new position, new intensity value and finally new radius.

Using the image above the total intensity for one column of picture elements is plotted against its position. Examples from this are illustrated in figure 7.2. Since it can be hard to see the differences in figure 7.2 the difference signals for the different methods and the true representation are plotted in figure 7.3. In figure 7.4 the difference signal are plotted but here the total intensity along a line is plotted against the position. More examples are shown in appendix 2.

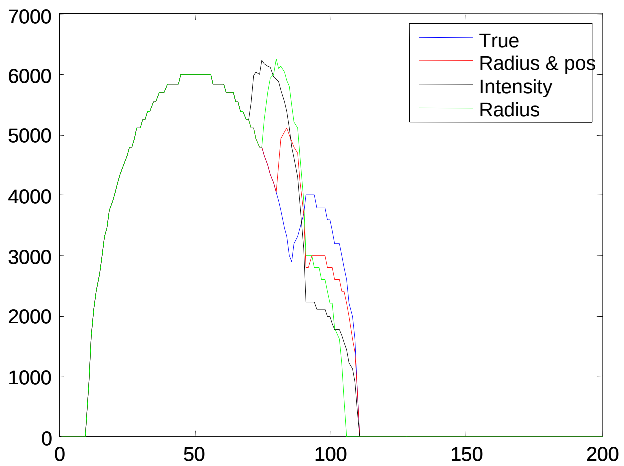


Figure 7.2 The intensity signals, when scanning from left to right (along the x-axis) for the different methods.

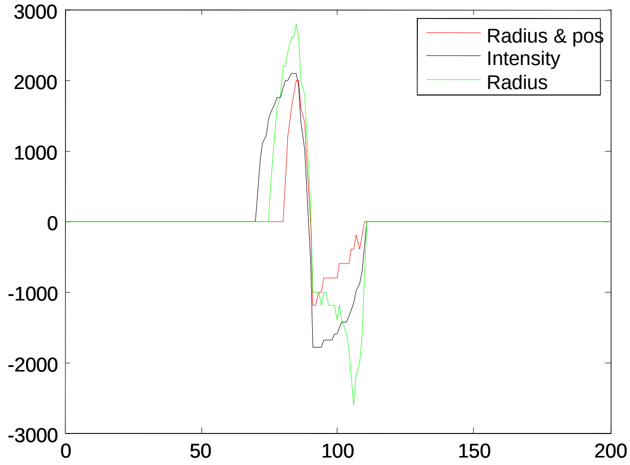


Figure 7.3 *The differences signals (when the true signal is subtracted from the others), when scanning from left to right (along the x-axis) for the different methods. The different methods are: change radius and position, change radius and finally change intensity.*

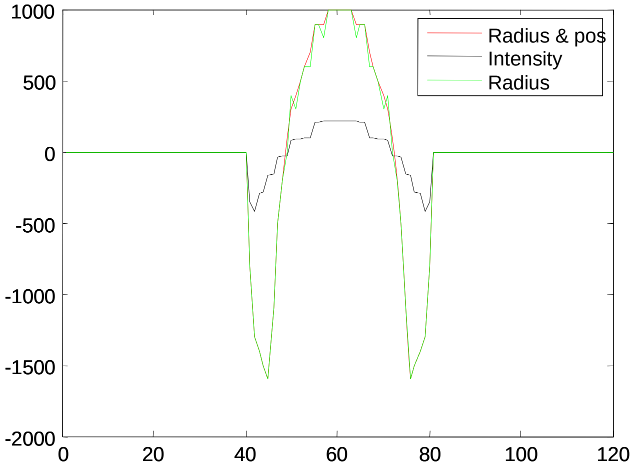


Figure 7.4 *The differences signals (when the true signal is subtracted from the others), when scanning from left to right (along the y-axis) for the different methods. The different methods are: change radius and position, change radius and finally change intensity.*

The chosen solution is to use the intensity/RCS change, in other words, keep the original position and radius just change the intensity or RCS value with the factor *f*.

8 Implementation of the algorithm

The above method is based on the simple fact that there is a sorted list with spheres sorted after their distance to the observer/seeker. The inputs to the sorting routine have to be described at some detail.

The input will be a one parameter and one two-dimensional matrix. The parameter is the number of spheres that are to be sorted in the matrix. Note that the matrix has fixed dimension, so in the majority of the cases it will not be filled. The two dimensional matrix, consists of the following for each sphere.

1. Distance
2. Position in horizontal aspect, (distance from centre line of observation).
3. Position in vertical aspect, (distance from centre line of observation).
4. Radius
5. RCS or Intensity
6. Identification – 1 indicates a ship, otherwise decoy
7. The visible factor – indicates how much is visible of a sphere (to be calculated)

The input matrix can be regarded as a number of rows each containing seven columns. Each row is describing one sphere. The output from the algorithm ought to be exactly like the input, except that the visibility factor is calculated. In order to manage that, the input matrix is copied to a new matrix with eight columns. The last column should contain the index order of the spheres from one up to the number of spheres. The sorting algorithm should manage to sort the rows in the matrix according to the distances.

In order to find a sorting algorithm, the internet was used. Searching with words quicksort and Matlab, gave a number of interesting results, some of the results are shown as references [8, 9]. None of the algorithms found, did exactly what the author wanted it to of course. So modifying one of them originally written in C, did the trick. The resulting code in Matlab is shown in appendix 3.

An overview of the final program is illustrated in figure 7.1 as a flow chart. The routine Circle_calc_list is the routine that calculates if circles overlap and the visibility for each circle and is shown in figure 7.2.

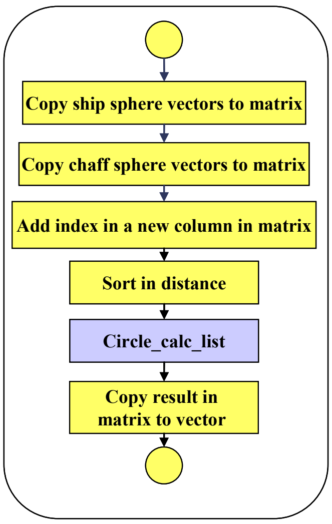


Figure 7.1 Flowchart of program.

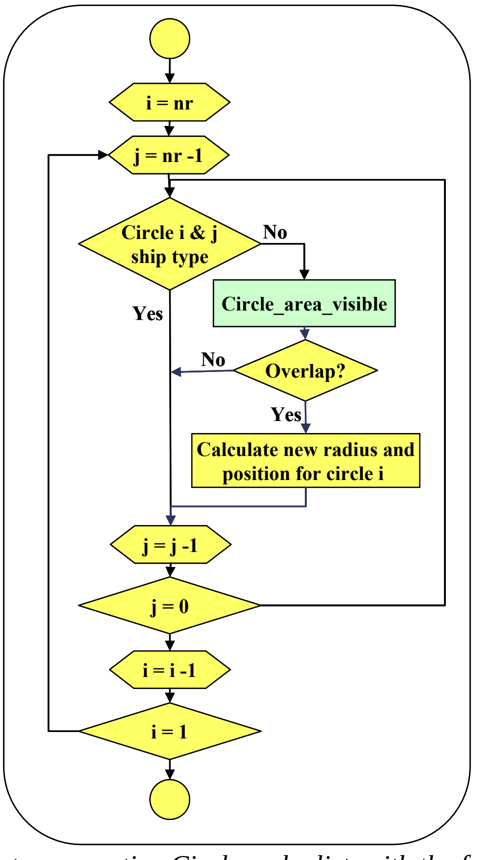


Figure 7.2 *Flowchart over routine Circle_calc_list, with the functionality described in chapter 6. The routine Circle_area_visible calculates the visible area for a circle overlapped by another circle, according to chapter 5.*

The code was originally written in Matlab, but has later been transformed to C. The routines have been integrated with the simulation environment ACSL [10].

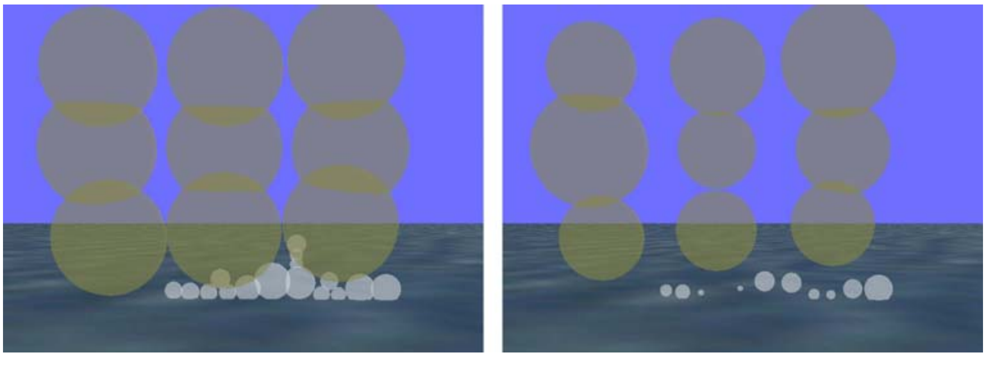


Figure 7.3 *Scenario, same as in figure 4.2. Here the spheres that are obscured have been reduced in size to illustrate the effect of obscured surfaces. Images generated in AFE [5].*

9 Final remarks

As stated in the introduction, handling ship and decoys as point sources gives a rather digitally appearance. Furthermore, if only point sources are used, the obscuring effects will not have to be handled.

By introducing extended targets as described in this report, simulations could be used to answer question of effects when chaff clouds are completely or partly obscuring the ship. It will also enable possibilities for closer studies of naval seduction techniques, for example by studying how to best deploy a number of decoys in order to get the maximum RCS for the chaff clouds. This really means that, the decoys are not obscuring each other (so much, at least) from the view of the missile seeker (IR or radar).

By letting a number of spheres describe the ship and one sphere a decoy, the rather simple calculations described in this report gives a better (more realistic) result in the outcome of the simulation.

This is achieved without adding too much of CPU time to the simulations.

10 References

[1] *Mikael Hansson, Slutrapport projekt Teknisk Hotsystemvärdering, FOI-R--1650--SE, December 2006*

[2] *Lars Berglund, Extended Targets in a Point Source Model, FOI-R--1312--SE, September 2004*

[3] *Lars Berglund, Carl Hedberg, CADIR – A Model For Generating Infrared Images, FOA-R-94-00059-3.6--SE, November 1994*

[4] *Annelie Tonnvik, Skymda mål I ACSL-modell, FOI-R--1367--SE, November 2004*

[5] *Johnny Eckerland, AFE 3.3 Reference Manual, FOI-R--1254--SE, May 2004*

[6] *D. Curtis Schleher, Introduction to Electronic Warfare, p 188, Artech House, US, 1986*

[7] *Ingelstam, Rönngren och Sjöberg, Tefyma, Sjöbergs förlag Stockholm/Bromma, 1977*

[8] *http://en.wikipedia.org/wiki/Quicksort, 2008-12-08*

[9] *http://www.kirupa.com/developer/actionscript/quickSort.htm, 2008-12-08*

[10] *Advanced Continuous Simulation Language (ACSL) Reference Manual, AEgis Simulation, Inc. September 1999*

11 Appendix 1. Graphs of differences signals for different cases.

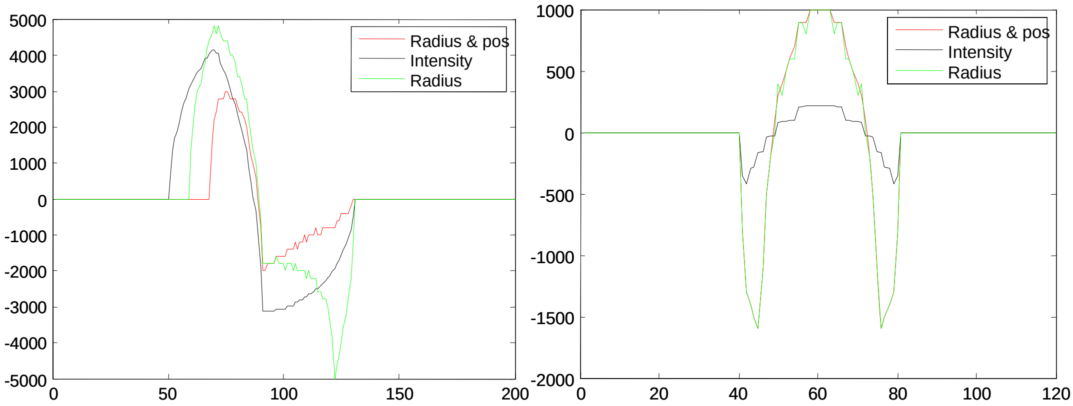


Figure A1.1 Radius of upper circle is 40. Radius of lower circle is 40. Distance between centres of circles is 40. Horizontal alignment.

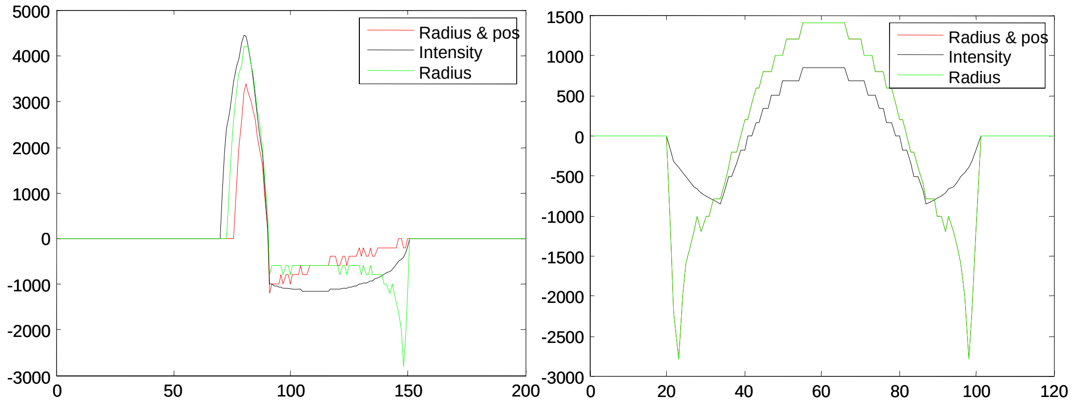


Figure A1.2 Radius of upper circle is 40. Radius of lower circle is 40. Distance between centres of circles is 60. Horizontal alignment.

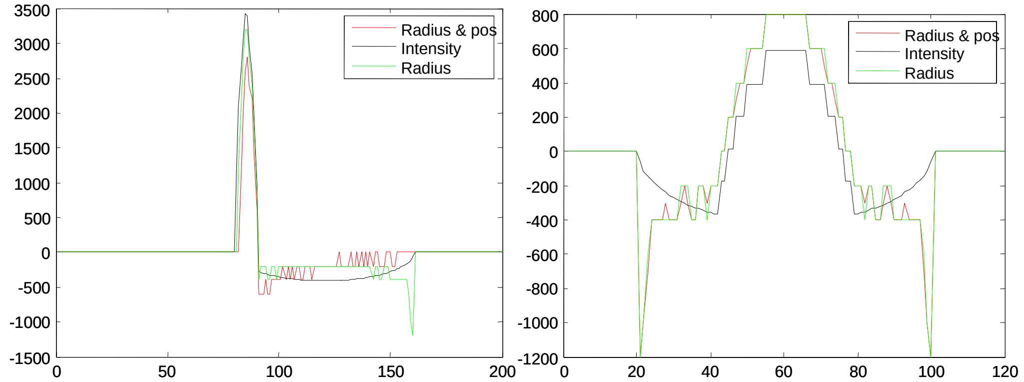


Figure A1.3 Radius of upper circle is 40. Radius of lower circle is 40. Distance between centres of circles is 70. Horizontal alignment.

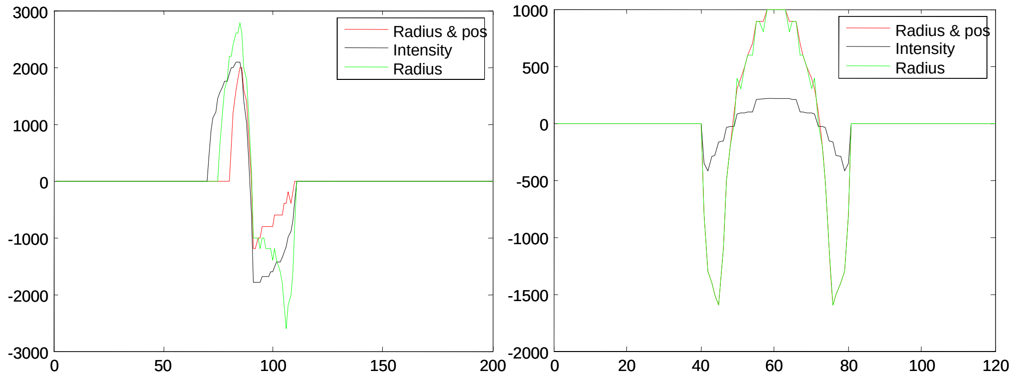


Figure A1.4 Radius of upper circle is 40. Radius of lower circle is 20. Distance between centres of circles is 40. Horizontal alignment.

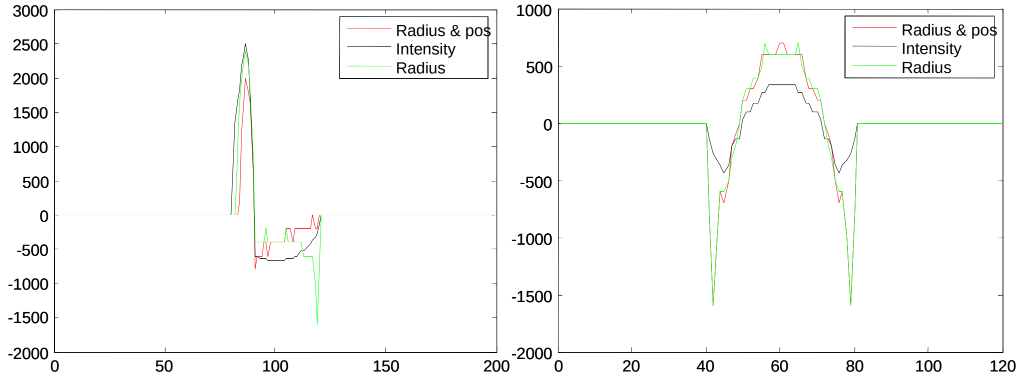


Figure A1.5 Radius of upper circle is 40. Radius of lower circle is 20. Distance between centres of circles is 50. Horizontal alignment.

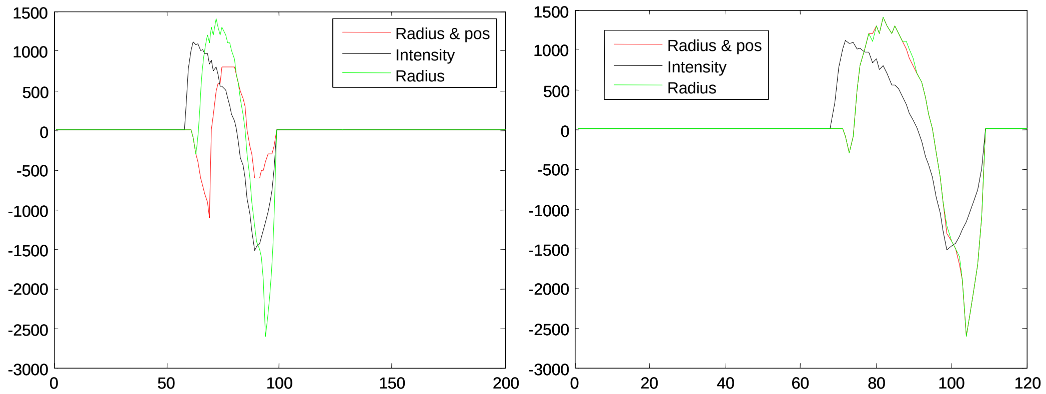


Figure A1.6 Radius of upper circle is 40. Radius of lower circle is 20. Distance between centres of circles is 40. The distance between the two circles is rotated 45 degrees clockwise relative horizon.

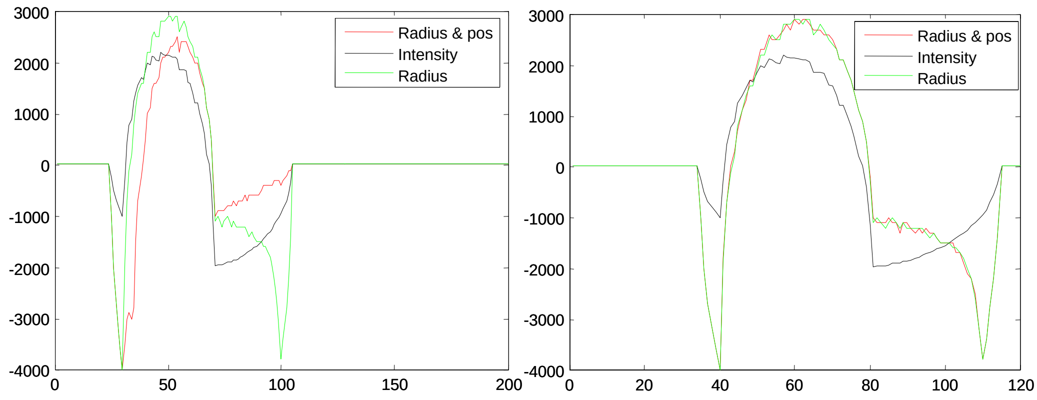


Figure A1.7 Radius of upper circle is 20. Radius of lower circle is 40. Distance between centres of circles is 20. The distance between the two circles is rotated 45 degrees clockwise relative horizon.

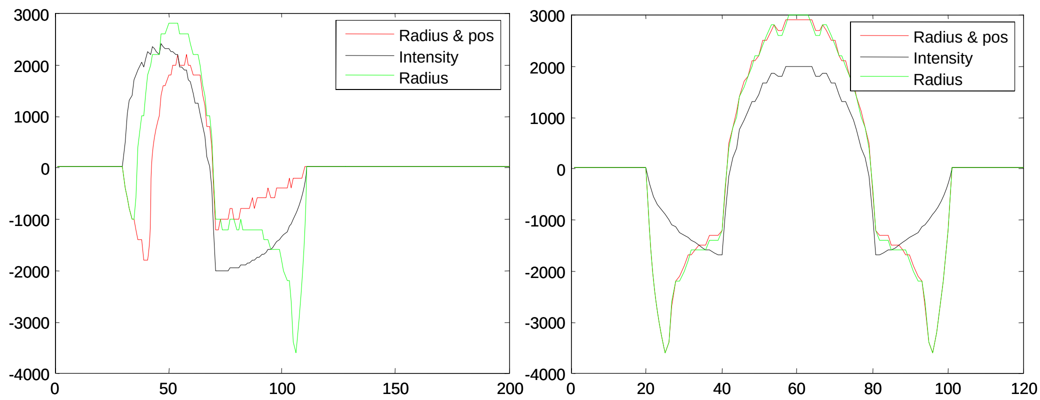


Figure A1.8 Radius of upper circle is 20. Radius of lower circle is 40. Distance between centres of circles is 40. Horizontal alignment.

12 Appendix 2. Quicksort

```
function [y] = quicksort1(x, left, right)
%
% Copied from http://www.kirupa.com/developer/actionscript/quickSort.htm
% Translated into Matlab
%
i = left;
j = right;
pivot = x(i,1);
temp = x(i,:);
while i < j
        while x(j,1) >= pivot && i < j
                j = j - 1;
        end;
        if i ~= j
                x(i,:) = x(j,:);
                i = i + 1;
        end;
        while x(i,1) <= pivot && i < j
                i = i + 1;
        end;
        if i ~= j
                x(j,:) = x(i,:);
                j = j - 1;
        end;
end;
x(i,:) = temp;
pivot = i;
temp = x(pivot,:);
i = left;
j = right;
if i < pivot
        x = [quicksort1(x,i,pivot-1)];
end;
if j > pivot
        x = [quicksort1(x,pivot+1,j)];
end;
y = x;
```