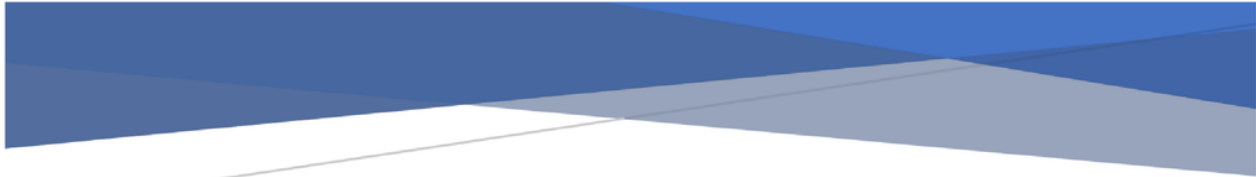


Influence Maps by P148551

by Andrew Ridgeway

FILE	4150_ANDREW_RIDGEWAY_INFLUENCE_MAPS_BY_P148551_177048_1686624689.DOCX (1.3M)		
TIME SUBMITTED	22-MAR-2019 07:01PM (UTC+0000)	WORD COUNT	2115
SUBMISSION ID	103041584	CHARACTER COUNT	11687



Influence Maps

By Andrew Ridgeway
P148551

L6 - Advanced AI & Machine Learning (M6X01080)

Table of Contents

Introduction	2
Influence Maps task 1	2
Calculations	3
Implement Influence Map Task 2.....	9
A.I. Influence Map Task 3	11
Testing Table	11
Source Code	12
References	16

Table of Contents

Figure 1 - A diagram of influence map[3].	3
Figure 2 - Here is a diagram of the gaussian kernel filter[2].....	4
Figure 3 - Here is a diagram of the influence cell[2].	4
Figure 4 - Here is a diagram of the calculations[2].	5
Figure 5 - Here is a diagram on the edge[2].	5
Figure 6 - Here is a diagram on the corner[2].	5
Figure 7 - Here is a diagram of the Gaussian kernel filter[5].....	6
Figure 8 - Here is a diagram of the rows and columns of the Influence map[5].	6
Figure 9 - Here is a diagram of a rows and columns of influence[5].	8
Figure 10 - Diagram of example calculation[5].	8

Introduction

The influence map as be evolved to be able to work out the spatial reasoning within game, such as Go and it a Chinese board game, this method has been used with RTS also known as real-time strategy game and are the know game, such as Age of Empires, StarCraft, Warcraft, and Warhammer 40,000[1]. Influence map can also be known as tactical analyse and this technique is highly used in RTS for the A.I. to keep a track on both sides, this means when the end user is playing a military game, such as Warhammer this technique would be used on the A.I. to keep track of A.I. enemies influence on all sides of the battlefields[2]. The way influenced Maps work is based around the objects that is concept within the game influence that has a strong relation between the influence area and the player or A.I., this means the influence would spread from the current position and throughout the whole area of the map[2][3].

Influence Maps task 1

The influence map algorithm works by using the values within the area of the game world or known as the map of the whole area that is showing both the end-users controlling side and a level of degree of influence, this means it would be the degree of security of the location of the whole area within the game world[2][4].

When it comes to the virtual world in the game influence map can help the A.I. to make a better decision by giving information that is useful about the surrounding areas of the virtual world within their game and the influence maps can provide information but there is three different types of such as[4]:

- **Understanding Surroundings:** - Influence maps has the capacity of summarising all the details within the world and once it looks at the world it's easier to understand, this means influence maps has a great understanding of what's going on in the area within the game, such as how much control enemies have in a certain area and how big is the military might in those areas[4].
- **Historical:** - Influence Maps not only has the capability to store information because influence map has the capability to remember what happened but only to a certain extent of that time, this means influenced Maps has the

capability of when this area has been assaulted and remember from how well the previous attack that just happened[4].

- **Predict Moves:** - Influence Maps has a type of feature but it's not looked at that much such as it has the capability to predict when attack is going to happen for future events, this means influence map will look at the area within the game world and use that information to know how the enemy had attacked previous and predict how to prevent that from happening[4].

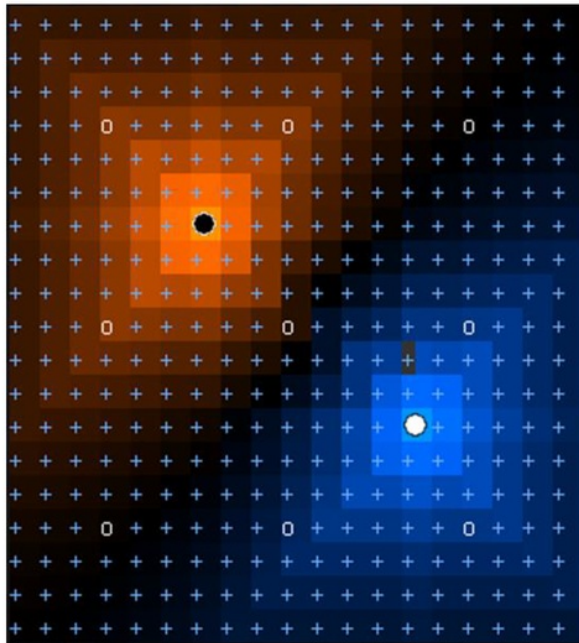


Figure 1 - A diagram of influence map[3].

Here is a simple diagram of how influence map by looking at the two units, such as the White and the black dot that is projecting two different colours, this means if these two different colours would get brighter they will influence the map by changing the black background to change into the colours that is shown on the diagram[3].

Calculations

When it comes to calculate an area with using influence map there is every unit within the game for every location of that area, this means each unit needs to be considered when it within the level of the games location[2]. With these calculations it depends how big the map is such as if it's a huge map the calculations will have a

huge task but if the map is small the calculations won't be as big, an example of this would be if the player was playing RTS with a huge amount of unit and a large location within their game then the calculations would be a lot more bigger to calculate that game[2].

Within an RTS game it would use these calculations such as the Big O notation that is $O(nm)$ known as Nanometre) and the memories would be $O(m)$ known as Metre), nm is used within such as m is the number of locations within the game and the n is the number of units[2].

An influence map would calculate the values of the pixels and then it would overlap the template and then there will be a mathematical towards the pixel and template values, so it becomes computed then the values of the template and pixels are added together, this means it would represent the overlap of both the template and the pixels within the area[2][5].

Here are the calculations of how influence map word that is shown below.

$$M = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Figure 2 - Here is a diagram of the gaussian kernel filter[2].

Here is the gaussian kernel filter that is used with the calculations.

$$\begin{array}{ccc} 5 & 6 & 2 \\ 1 & 4 & 2 \\ 6 & 3 & 3 \end{array}$$

Figure 3 - Here is a diagram of the influence cell[2].

Here is the influence cell that would be used to calculate with the gaussian kernel filter.

$$v = \left(\begin{array}{ccc} 5 \times \frac{1}{16} & + & 6 \times \frac{2}{16} & + & 2 \times \frac{1}{16} & + \\ 1 \times \frac{2}{16} & + & 4 \times \frac{4}{16} & + & 2 \times \frac{2}{16} & + \\ 6 \times \frac{1}{16} & + & 3 \times \frac{2}{16} & + & 3 \times \frac{1}{16} & \end{array} \right) = 3.5.$$

Figure 4 - Here is a diagram of the calculations[2].

Here are calculations of used gaussian kernel filter and influence cell to get the results.

$$\frac{1}{9} \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix}$$

Figure 5 - Here is a diagram on the edge[2].

Here are calculations of when the influence is at the edge.

$$\frac{1}{12} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \end{bmatrix}$$

Figure 6 - Here is a diagram on the corner[2].

Here are calculations of when the influence is at the corner.

For the calculations will be using three of the table such as gaussian kernel filter, influence map and influence cell, the calculations will be shown down below.

1/16 2/16 1/16
 2/16 4/16 2/16
 1/16 2/16 1/16

Figure 7 - Here is a diagram of the Gaussian kernel filter[5].

Student Number	Cell (Row,Col)			
1451588	(2,2)			
1650628	(3,2)			
1805212	(4,2)			
1550157	(5,2)			
1550637	(3,2)			
1550750	(3,4)			
1551077	(3,5)			
1650650	(4,5)	example	example	example
1650717	(6,5)			
1650732	(6,4)			
1650734	(6,3)			
1651487	(6,2)			

	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	32	32	32	34	34	36
Row 2	26	26	26	26	30	36
Row 3	18	18	18	24	24	24
Row 4	14	16	16	22	22	24
Row 5	14	14	14	20	20	18
Row 6	16	20	20	22	24	26
Row 7	16	22	22	24	26	30
Row 8	18	24	24	26	28	32

Figure 8 - Here is a diagram of the rows and columns of the Influence map[5].

With the top of the image is showing the rows and columns the one with example and marked in yellow on is the one will be using, and it shows on the bottom of the image the squares that will be used.

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

X

24	24	24
22	22	24
20	20	18

=

1/16 * 24	1/16 * 24	1/16 * 24
1/16 * 22	1/16 * 22	1/16 * 24
1/16 * 20	1/16 * 20	1/16 * 18

1.5	3	1.5
2.75	5.5	3
1.25	2.5	1.125

Here is the mathematical equation that is used to show how it's done such as the right is 1 between 4 / 16 x with the grid shown on figure 2 and would equal that is shown in this equation.

$$\frac{1.5 + 3 + 1.5 + 2.75 + 5.5 + 3 + 1.25 + 2.5 + 1.125}{9} = \frac{22.125}{9} = 2.4583$$

Here is the next step shown above of the 9 numbers within the grid is to add all up and divided by the same number in the grid such as dividing it by 9.

Student Number	Cell (Row,Col)	r
1451588	(2,2)	5
1650628	(3,3)	6
1805212	(4,4)	7
1550157	(5,4)	6
1550637	(3,4)	7
1550750	(3,2)	5
1551077	(3,5)	7
1650650	(3,6)	9
717	(4,5)	9
1650732	(3,4)	8
1650734	(6,3)	8
1651487	(6,2)	8

Figure 9 - Here is a diagram of a rows and columns of influence[5].

In this diagram it shows the location of the rows and column in yellow that is going to be used for the calculations.

$$Id = \frac{Io}{(e^{(d-r)} + 1)} \quad \text{equation 1}$$

Figure 10 - Diagram of example calculation[5].

Here is an example of how the calculations will be used shown in the diagram such as there will be two location of influence sources, such as here is location of influence source 1 and the second one will be location of influence source 2.

Influence source 1
$3.4 = \frac{3.5}{(e^{(\sqrt{(3-1)^2 + (6-1)^2} - 9)} + 1)}$

Influence source 2
$4.18 = \frac{4.3}{(e^{(\sqrt{(3-1)^2 + (6-1)^2} - 9)} + 1)}$

By looking at both of the table shown above the equation with equal 3.4 and the second one would equal 4.18, once both are calculated the next step would be adding both together.

$$3.4 + 4.18 = 7.58$$

By looking at the table above the calculations are added together to make 7.58.

Implement Influence Map Task 2

For this appropriate task will be the gaussian kernel influence because it's the easiest calculations to understand and by looking at these calculations that was created in code to show how easy it is to implement in Visual Studio as a mathematical equation. These mathematical equations were taking from task 1 and turning into a code function.

Influence calculations

This table shows the influence cells.

```

11 // this line of code is use to do the equation of the influence
12 void mathInfluence(void) {
13     float sum = 0, driv = 0;
14     // this is use to set up the table influence map
15     cout << "Influence map" << endl;
16     float inf[MAXX][MAXX]{
17         32, 32, 32, 34, 34, 36,
18         26, 26, 26, 26, 30, 36,
19         18, 18, 18, 24, 24, 24,
20         14, 16, 16, 22, 22, 22,
21         14, 14, 14, 20, 20, 20,
22         16, 20, 20, 22, 24, 26,
23         16, 22, 22, 24, 26, 30,
24         18, 24, 24, 26, 28, 32
25     };
26
27     // this line of code is use to show the influence map
28     for (int i = 0; i < MAXX; i++) {
29         for (int j = 0; j < MAXX; j++)
30             cout << "[" << inf[i][j] << "]" << " ";
31         cout << endl << endl;
32     }
33 }

```

These calculations are shown the gaussian kernel and influence cell from task 1.

```

34 // this is use to set up the table of the kernel filter
35 cout << "3x3 Gaussian kernel filter" << endl;
36 float a[MIN][MIN]{
37     0.0625, 0.125, 0.0625,
38     0.125, 0.25, 0.125,
39     0.0625, 0.125, 0.0625
40 };
41
42 // this line of code is use to show the kernel filter
43 for (int i = 0; i < MIN; i++) {
44     for (int j = 0; j < MIN; j++)
45         cout << "[" << a[i][j] << "]" << " ";
46     cout << endl << endl;
47 }
48
49 // this is use to set up the table of the student number
50 cout << "Student number" << endl;
51 float b[MIN][MIN]{
52     24, 24, 24,
53     22, 22, 24,
54     20, 20, 20
55 };
56
57 // this line of code is use to show the student number
58 for (int i = 0; i < MIN; i++) {
59     for (int j = 0; j < MIN; j++)
60         cout << "[" << b[i][j] << "]" << " ";
61     cout << endl << endl;
62 }
63 }

```

This table shows the calculations of x in the table together and then adding them up. then divided them by 9.

```

64 // this line of code is use to show the calculations
65 // by times together
66 cout << "Time kernel function and student number" << endl;
67 float c[N][N];
68 for (int i = 0; i < N; i++) {
69     for (int j = 0; j < N; j++) {
70         c[i][j] = a[i][j] * b[i][j];
71         cout << "[" << c[i][j] << "]" << " ";
72     }
73     sum = sum + c[i][j];
74     cout << endl << endl;
75 }
76
77 // this line of code is use to show the calculations
78 // by adding all the cell together
79 cout << "Add all the cell together" << endl;
80 cout << "[" << sum << "]" << " ";
81 cout << endl << endl;
82
83 // this line of code is use to show the calculations
84 // by divided by 9
85 cout << "Divided by 9" << endl;
86 div = sum / 9;
87 cout << "[" << div << "]" << " ";
88 cout << endl << endl;
89

```

Here is the table which shows the results of the code working in that the console screen.

```

▶ D:\Python\Computer Game Development\Level 8 Year 7 Addressed AI & Machine Learning\MOVIE1905\Assignment\influence
[32] [32] [35] [34] [34] [36]

Influence map
[26] [26] [26] [26] [30] [36]

[18] [18] [18] [24] [24] [24]

[14] [16] [16] [22] [22] [22]

[14] [14] [14] [20] [20] [20]

[16] [20] [20] [22] [24] [26]

[16] [22] [22] [24] [26] [30]

[18] [24] [24] [26] [28] [32]

3x3 Gaussian kernel filter
[0.0025] [0.125] [0.0025]

[0.125] [0.25] [0.125]

[0.0025] [0.125] [0.0025]

Student number
[24] [24] [24]

[22] [22] [24]

[20] [20] [20]

Time kernel filter and student number
[1.5] [3] [1.5]

[2.75] [5.5] [3]

[1.25] [2.5] [1.25]

Add all the cell together
[22.25]

Divided by 9
[2.47222]

Press any key to continue . . .

```

here is the influence map those shown in the console screen that it's running but this is not created in OpenGL but it's more created with symbols and characters instead, the characters are symbolised as follow such as g is symbolise as the enemy A.I. while p is symbolised as the player A.I. and e is the symbolise of one of the points that player A.I. have reached.

```
#E      .....G#
#####
####           #
##          #####
###         ##
##        ####
##       #
##     #####
##    #####
##   #####
##  #####
## #####
## #####
## #####
## #####
## #####
## #####
## #####
## #####
## #####
## #####
```

A.I. Influence Map Task 3

for task 3 is the player A.I. would go through the map to find to find the endpoint while getting chased by enemy A.I. and other enemy A.I. just wandering around. For example, one of the enemy A.I. would try pursuing the player A.I. to prevent it to reach the endpoint, while the other enemies A.I. will have random movements so if the player A.I. would accidentally touching it would prevent the player A.I. from reaching the endpoint.

Testing Table

No.	Testing	Expected Results	Actual Results	Pass/Fail
	Reading text file for the map	would be able to show the map	while trying to show the map it breaks	fail
	hard code in the map	displays the map on the console screen	map is shown	pass
	enemy A.I. goes after a symbol	seeing if the enemy A.I. will move to the symbol	The enemy A.I. is moving to the symbol	pass
	enemy A.I. complete the task	to see if the enemy A.I. touches the simple the console screem will pause	console screen has pause	pass
	player A.I. moving	to see if the player A.I. is moving	this causes the player two move in One Direction and not move in different directions	fail

Source Code

Here is the table of the Source Code that contains the source and the header.

Source	
These lines of code are used for the position of the coordinates for the space and Walls and coordinates for the A.I.	<pre>GoGhost gh; Ghost gh2; // this is used for the coordinates int pX = 20, pY = 14; int gX = 14, gY = 1; int eX = 1, eY = 1; // this line of code is used for the coordinates for the wall and space of the map void arrayAI(int x, int y, int go, int back) { if (theMap[y][x] == space theMap[y][x] == space) { theMap[y][x] = theWall; gh.x = x; gh.y = y; gh.go = go; gh.back = back; theGoGh.push_back(gh); } }</pre>
Here is the enemy A.I. that would set up the coordinates for it to pursue the enemy A.I. and its setup the space and wall, to make sure it does not go through the wall.	<pre>// this line of code is used for the enemy AI to look at all coordinates void theAI(int sx, int sy, int x, int y) { int i = 0; gh.x = sx; gh.y = sy; gh.go = 0; gh.back = -1; theGoGh.push_back(gh); // this is used to for the enemy AI to find the player AI while (i < theGoGh.size()) { if (theGoGh[i].x == x && theGoGh[i].y == y) { while (theGoGh[i].go != 0) { gh2.x = theGoGh[i].x; gh2.y = theGoGh[i].y; theGhost.push_back(gh2); i = theGoGh[i].back; } break; } // this is used to for the enemy AI not to go through the wall arrayAI(theGoGh[i].x + 1, theGoGh[i].y, theGoGh[i].go + 1, i); arrayAI(theGoGh[i].x - 1, theGoGh[i].y, theGoGh[i].go + 1, i); arrayAI(theGoGh[i].x, theGoGh[i].y + 1, theGoGh[i].go + 1, i); arrayAI(theGoGh[i].x, theGoGh[i].y - 1, theGoGh[i].go + 1, i); i++; } }</pre>

Here is the player A.I. that would set up the coordinates and this is used for the AI to check spaces next to it to see if it can move or not to that space.

```
// this line of code is used for the player AI to look at all coordinates
// but the player AI breaks the code
void theGoodAI(int x, int y) {
    int choice = 0, moveUp = 0, moveDown = 2, moveRight = 1, moveLeft = 3;

    int xSpeed = 1;
    int ySpeed = 1;

    // this is used to check if there is a wall
    if (theMap[x + 1][y] == theWall || theMap[x - 1][y] == theWall)
        xSpeed *= -1;
    if (theMap[x][y + 1] == theWall || theMap[x][y - 1] == theWall)
        ySpeed *= -1;

    // this is used to check if there is a space go
    if (theMap[x + 1][y] == space)
        choice = moveRight;
    if (theMap[x - 1][y] == space)
        choice = moveLeft;
    if (theMap[x][y + 1] == space)
        choice = moveUp;
    if (theMap[x][y - 1] == space)
        choice = moveDown;

    // this is used to check coordinates from the player AI
    if (choice == moveUp || choice == moveDown)
        y += ySpeed;
    else if (choice == moveLeft || choice == moveRight)
        x += xSpeed;
    printf("%c", goodGuy);
}
```

This line of code is used to call the functions and used to run the both player and enemy A.I.

```
// this line of code is used to call the functions
void playTime() {
    mazeOfMap();
    int xi, yi;

    theXYPos(x, y);
    printf("%c", exits);
    theAI(gX, gY, pX, pY);

    // this is used to run the do loop for player AI and enemy AI
    do {
        theXYPos(pX, pY);
        //printf("%c", goodGuy);
        //theGoodAI(pX, pY);
        //theMap[gY][gX] = pickup;
        theXYPos(gX, gY);
        printf("%c", pickup);

        gX = theGhost.back().x;
        gY = theGhost.back().y;
        theGhost.pop_back();

        theXYPos(gX, gY);
        printf("%c", goGhost);

        if (gX == pX && gY == pY)
            break;

        Sleep(200);
    } while (1);

    printf("\n\n\n\n");
    system("pause");
}
```

This line of code is only used to call the functions.

```
// this line of code is used to run the application
int main() {
    playTime();

    return 0;
}
```

Header

This line of code is used for reading the text file to allow it to be used for the map.

```
// this line of code is used to read a text file
// but line of code breaks the all tof the code
/*void readFile() {
    string line;
    ifstream myfile1;
    myfile1.open("influ.txt");

    if (myfile1.is_open()) {

        // read the line by line of the file
        while (getline(myfile1, line)) {
            line.copy(&theMapId[WIDTH * count], WIDTH, 0);
            count++;
        }
        myfile1.close();
    }

    for (int i = 0; i < WIDTH; i++)
    {
        for (int j = 0; j < HEIGHT; j++)
        {
            theMap[i][j] = theMapId[WIDTH * i + j];
        }
    }
}*/
```

This line of code is used to draw the map and return the X and Y.

```
// this line of code is used to return the x and y
int getXy() {
    return x, y;
}

// this line of code is used to set up the map
void drawOfMap(char p) {
    theMap[x][y] = p;
    theXYPos(x, y);
    printf("%c", p);
}
};
```

References

- [1] C. Su-Hyung, Jang. Sung-Bae, *Evolving Neural NPCs with Layered Influence Map in the Real-time Simulation Game 'Conqueror.'* 2008.
- [2] I. Millington and J. Funge, *Artificial Intelligence for Games, Second Edition.* 2009.
- [3] N. Hansson, "Game School Gems: Influence Maps I," *Game School Gems*, 2010. [Online]. Available: <http://gameschoolgems.blogspot.com/2009/12/influence-maps-i.html>. [Accessed: 11-Mar-2019].
- [4] A. J. Champandard, "The Mechanics of Influence Mapping: Representation, Algorithm & Parameters," *AiGameDev*, 2011. [Online]. Available: <http://aigamedev.com/open/tutorial/influence-map-mechanics/>. [Accessed: 12-Mar-2019].
- [5] I. Gordon, Dickers. Wells, "Course: L6 - Advanced AI & Machine Learning (M6X01080)," *moodle.uwtsd*, 2019. [Online]. Available: <https://moodle.uwtsd.ac.uk/course/view.php?id=2533>. [Accessed: 17-Mar-2019].

Influence Maps by P148551

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

48/100

PAGE 1

PAGE 2

PAGE 3



Comment 1

??? explain

PAGE 4

PAGE 5



Comment 2

You're confused here.

nm might well mean nanometer in a different context but nm means $n \cdot m$ where n and m are variables eg area and number of objects, etc)

PAGE 6

PAGE 7

PAGE 8



Comment 3

no need to divide by 9!

PAGE 9

PAGE 10



Comment 4

some rounding errors here. But close enough.

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

GRADING FORM: SMU GENERIC_GJ

ANDREW RIDGEWAY

45.50

RESEARCH-INFORMED LITERATURE



References generally used appropriately and include standard texts. Could be more varied.

6.5

KNOWLEDGE AND UNDERSTANDING



13/25 Essential ideas understood. Methods used demonstrated by example. Good.

13

ANALYSIS



12/20 Relevant calcs attempted. Beware rounding errors, otherwise fine. No need to divide by 9! Averaging already included in kernel values (they sum to one). Complexity discussion a little confused. Test schedule designed and implemented.

12

PRACTICAL APPLICATION AND DEP.



14/40 Conv algorithm implemented and tested successfully against report calculations. Comment appropriate as is the use of .h files and class definitions. Not much more done beyond this. Testing included- good.

14

SKILLS FOR PROFESSIONAL PRAC.



References relevant. Report structured but content needs further checking for grammatical errors/typos.

2/5