International Journal of Computer Applications (0975 – 8887)
Volume 65– No.8, March 2013

ON Tiling Patterns Involving Islamic Stars with an Odd Number of Vertices

T.Gangopadhyay
XLRI
C.H.Area(E), Jamshedpur,
India

ABSTRACT

Islamic star patterns have been extensively studied for their symmetry and aesthetic appeal. This paper presents a simple construction method for generating these stars on computers and constructing new tiling patterns that involve stars with an odd number of vertices

General Terms

Islamic, Art, Pattern, Algorithm, Turbo C++, Program.

Keywords

Polygon, Rosette, Star, Symmetry, Trigonometric,

1. INTRODUCTION

Owing to their symmetry and aesthetic appeal, Islamic star and rosette patterns have been studied extensively by several researchers(Gr¨unbaum and Shephard [8], Abas and Salman [1], Bourgoin[ ],Dewdney [5], Castera [3], Dunham[6]). The innate symmetry of these structures have led to geometrical constructions of these as well as tiling designs devised through putting such structures in contact(Castera [4], Gr¨unbaum and Shephard[7], Kaplan[1]. Kaplan[10] has devised an elaborate construction method for stars based on methods described originally by Henkin [9] and Lee [12]. In addition Kaplan[11] has also devised a number of interesting tiling methods by putting stars in contact. Most of these styling styles use stars with an even number of vertices. In this paper, a simpler method to construct a star is described and two new generalized tiling methods which use stars with both even and odd number of vertices. In addition a threefold symmetric tiling is constructed for all stars whose number of vertices are a multiple of three. Using stars with odd number of vertices to create new tiling patterns with them are the main distinguishing features of this paper.

2. HOW TO DRAW A STAR

Let n and d be integers. Let A[i] be the ith vertex of the star, and let x[i] and y[i] denote respectively the abscissa and ordinate of A[i], where $0 \leq i \leq n-1$. Let x0 and y0 respectively denote the abscissa and ordinate of the centre of the star and let its radius be k. Then

$x[i] = k\cos(2\pi i / n)+x0$ and $y[i]=k\sin(2\pi i / n)+y0$.

For each i, $0 \leq i \leq n-1$, let B[i] be the point of intersection of the lines A[i]A[(i+d)mod n] and A[i+1]A[(i+n-d+1)mod n] and C[i] be the point of intersection between the lines A[i]B[i] and A[(i+d-2+n) mod n]B[(i+d-3+n)mod n].Further, let D[i] be the point of intersection between the lines A[i]C[i] and A[(i+2)mod n]C[(i+1) mod n]

Then, for $4 \leq d \leq (n-1) / 2$, star(n, d) is drawn by joining C[i] to both A[i] and A[(i+d-2+n)mod n] for each i, 0<=i<=n-1. Star(n,3) is drawn by joining D[i] to both A[i] and A[(i+2)mod n] for each i, $0 \leq i \leq n-1$. In Figure 1 below, we illustrate the construction of star(11, 4) for k=120, in which A[3], B[3], C[3] and A[7] are respectively highlighted by a cyan, a magenta, a yellow and a blue circle.

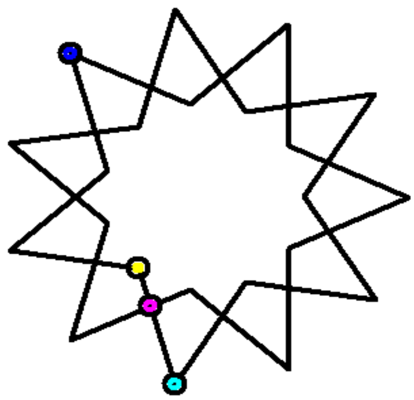

Fig 1 : star(11, 4)

In the code in section 3, the functions interx and intery respectively obtains the abscissa and ordinate of two lines joining two pairs of points. The function star draws star(n,d) when flag variable fl is set to zero and its mirror image when it is set to 1.

3. TILING WITH STARS

The process of tiling is illustrated for star(n,d) with n=7 and d=3. The basic prototype is a double star which is shown in Figure 2 and is drawn as follows. First star(7,3) is drawn with its center coordinates at i and j and the length of its radius as k. Then its mirror image is drawn adjacent to star(7,3) so that their corresponding vertices touch each other. This is achieved by setting the abscissa of the center of the mirror image as i+2k, while keeping the ordinate unchanged. In Figure 2, star(7,3) and its mirror image are drawn in the manner described above.

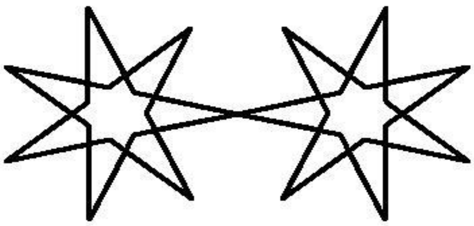

Fig. 2 : The Prototype - double star

The image in Figure 2 is then repeated by increasing the center coordinates of star(7,3) at an interval of 2k(1+ cos((angle / 2) π / 180)) for the abscissa i and 2k sin((180 – ((n+1) / 4)ang) π / 180) for the ordinate j.where angle = 360 / n. In figure 3 we show the output after one vertical and one horizontal rotation. The line in green indicates the distance 2k (1+ cos((angle / 2) π / 180)) between the center of the original star(7,3) and that of the star(7,3) after one horizontal(abscissa-wise) iteration. Similarly the line in cyan indicates the distance 2k sin((180 – ((n+1) / 4)ang) π / 180) between the center of the original star(7,3) and that of the star(7,3) after one vertical(ordinate-wise) iteration. The final output is shown in Figure 4.

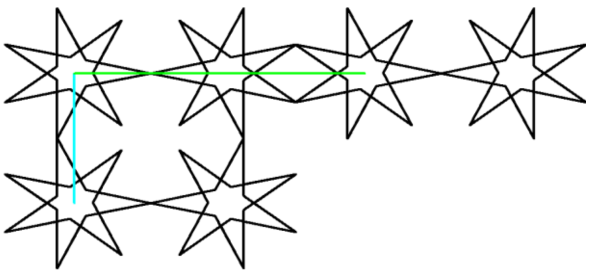

Fig. 3 : The double star after iteration

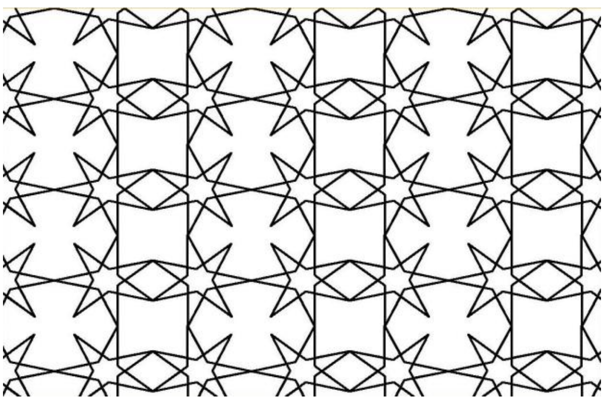

Fig 4 : The final output of tiling

The algorithm described in sections 2 and 3 is presented in the form of the C++ code in section 4.

4. THE CODE

The code uses three functions interx, intery and star which are declared first..

```
float interx(float x1,float y1,float x2,float y2,float x3,float
y3,float x4,float y4)
{return (y3-y1-x3*(y4-y3)/(x4-x3)
+x1*(y2-y1)/(x2-x1))/((y2-y1)/(x2-x1)-(y4-y3)/(x4-x3));}

float intery(float x1,float y1,float x2,float y2,float x3,float
y3,float x4,float y4)
{return (x3-x1-y3*(x4-x3)/(y4-y3)
+y1*(x2-x1)/(y2-y1))/((x2-x1)/(y2-y1)-(x4-x3)/(y4-y3));}

void star(int fl,int k,int n,
int d,int x0,int y0)
{float x[100],y[100],xx[100],yy[100],
xz[100],yz[100],u[100],v[100];

for(int i=0;i<n;i++)
{x[i]=k*cos(2*3.14*i/n)+x0;
y[i]=k*sin(2*3.14*i/n)+y0;
}
for(int i=0;i<=n;i++)
{xx[i]= interx(x[i],y[i],x[(i+d)%n],
y[(i+d)%n],x[(i+1)%n],y[(i+1)%n],
x[(i+n-d+1)%n],y[(i+n-d+1)%n]);

yy[i]= intery(x[i],y[i],x[(i+d)%n],
y[(i+d)%n],x[(i+1)%n],y[(i+1)%n],
x[(i+n-d+1)%n],y[(i+n-d+1)%n]);}

for(int i=0;i<n;i++)
{xz[i]= interx(x[i],y[i],xx[i],yy[i],
x[(i+d-2+n)%n],y[(i+d-2+n)%n],
xx[(i+d-3+n)%n],yy[(i+d-3+n)%n]);

yz[i]= intery(x[i],y[i],xx[i],yy[i],
x[(i+d-2+n)%n],y[(i+d-2+n)%n],
xx[(i+d-3+n)%n],yy[(i+d-3+n)%n]);
}

setlinestyle(SOLID_LINE,0,4);

for(int i=0;i<n;i++)
{if(fl==0){line(x[i],y[i],xz[i],yz[i]);
line(x[(i+d-2+n)%n],y[(i+d-2+n)%n],xz[i],yz[i]);}

else{line(2*x0-x[i],y[i],2*x0-xz[i],yz[i]);
line(2*x0-x[(i+d-2+n)%n],y[(i+d-2+n)%n],2*x0-xz[i],yz[i]);
}
}

for(int i=0;i<n;i++)
{
u[i]=interx(x[i],y[i],xz[i],yz[i],
x[(i+2)%n],y[(i+2)%n],xz[(i+1)%n],yz[(i+1)%n]);
v[i]=intery(x[i],y[i],xz[i],yz[i],
x[(i+2)%n],y[(i+2)%n],xz[(i+1)%n],yz[(i+1)%n]);
if(d==3){if(fl==0){line(x[i],y[i],u[i],v[i]);
line(x[(i+2)%n],y[(i+2)%n],u[i],v[i]);}
else {line(2*x0-x[i],y[i],2*x0-u[i],v[i]);
```

```
line(2*x0-x[(i+2)%n],y[(i+2)%n],2*x0-u[i],v[i]);}}
}

}

void main()
{
initwindow(1040,760);

int n=7;int k=90;int d=3;
float ang=360./n;

for(int i=0;i<1040;i+=2*k*(1+cos(ang/2.*3.14/180.)))
for(int j=0;j<800;j+=2*k*sin((180.-(n+1.)/4.*ang)*3.14/180))
{star(0,k,n,d,i,j);
star(1,k,n,d,i+2*k,j);}

getch();
closegraph();
}
```

5. VARIATIONS ON THE SAME THEME

By changing the values of n and d in the main program we may generate a number of interesting star patterns.,

VARIATION 1.

Let n=11 and d=3.Then the resulting output is illustrated in Figure 3.

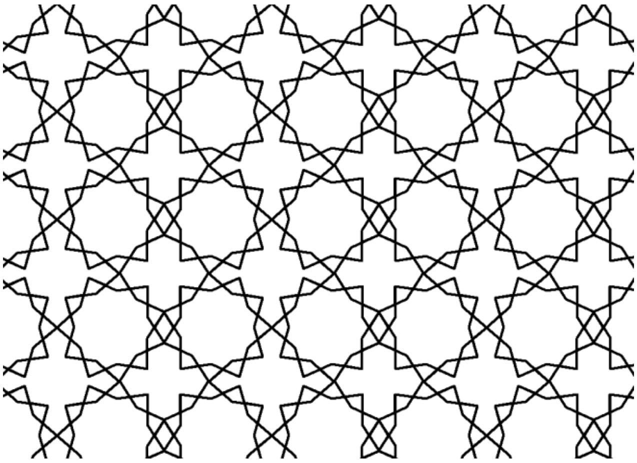

Fig. 5 : Variation 1 -

VARIATION 2.

Let n=11 and d=4. Then the resulting output is illustrated in Figure 6.

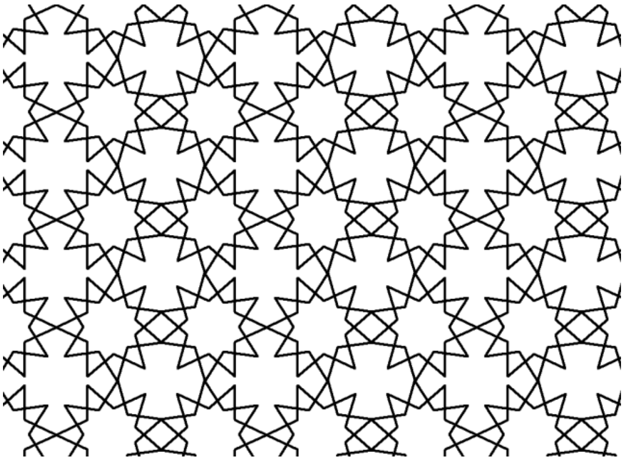

Fig. 6 : Variation 2

VARIATION 3.

Let n=11 and d=5; Then the resulting output is illustrated in Figure 7.

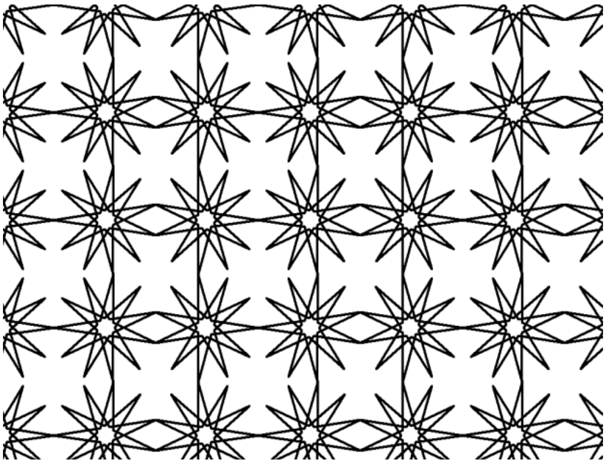

Fig. 7 : Variation 3

VARIATION 4.

Let n=11. By juxtaposing d=3, 4 and 5 we obtain the output illustrated in Figure 8.

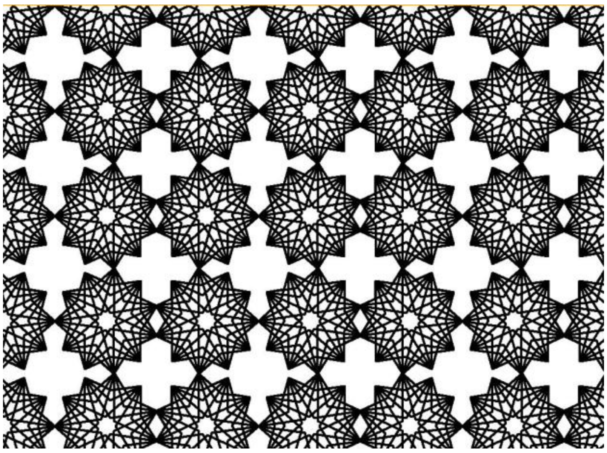

Fig. 8 : Variation 4

VARIATION 5.

Let n=9. In addition to juxtaposing d=3 and 4 if k is allowed to vary in a range and each cooresponding realization is coloured by the colour k, then one obtains the output illustrated in Figure 9.

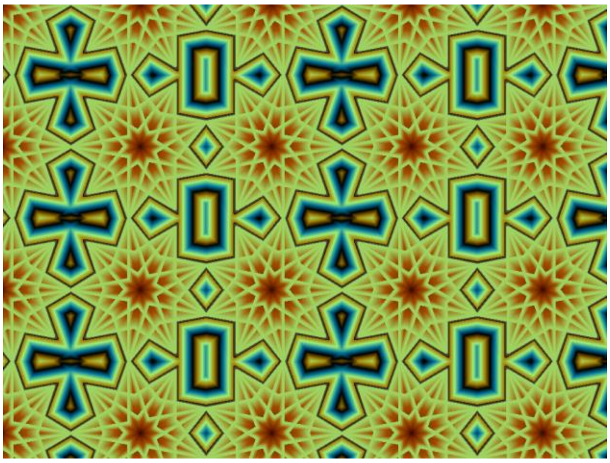

Fig. 9 : Variation 5

6. AN ALTERNATE DESIGN

A more compact style of tiling is achieved by first drawing two copies of star (7,3), both with radius k - one with its center cordinates as (i, j) and the other with its center coordinates at (i+k+k cos((ang) π / 180) +2k cos((ang/2) π / 180), j– k sin((ang) π / 180)) where angle = 360 / n. Next their mirror images are drawn with center coordinats at (i-2k cos((ang/2) π / 180), j) and (i+k+k cos((ang) π / 180), j-k sin((ang) π / 180)). This results in two double stars as shown in figure 10.

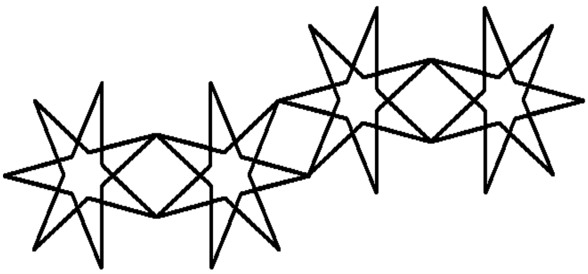

Fig. 10 : Two double stars

The image in Figure 10 is then repeated by increasing the center coordinates (i, j) of star(7,3) in the first double star at an interval of 2k(1+ cos((angle) π / 180)) + 4k cos((angle/2) π / 180)) for the abscissa i and 2k sin((180 – ((n+1) / 4)ang) π / 180) for the ordinate j..
The main program in the code in section 4 above could be changed as follows to accommodate this new style of tiling

```
void main()
{
int n=7;int k=90;float ang=360./n;float sa;
sa=k*cos(ang*3.14/180.);

for(int i=0;i<1440;i+=2*(sa+k)+4*k*(cos(ang/2.*3.14/180.)))
for(int j=-100;j<800;
j+=2*k*sin((180.-(n+1.)/4.
*ang)*3.14/180.))

{star(0,k,n,3,i,j);
Star(1,k,n,3,i-2*k*(cos(ang/2.*3.14/180.)),j);
star(1,k,n,3,i+k+sa, j-k*sin(ang*3.14/180.));
star(0,k,n,3,i+k+sa+2*k*(cos(ang/2.*3.14/180.)),
j-k*sin(ang*3.14/180.));
}

getch();
closegraph();
```

The output of this is illustrated in Figure 11. One may compare Figure 11 to Figure 4.

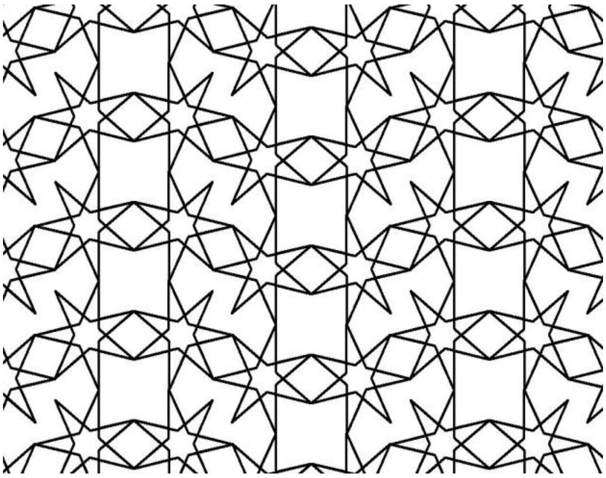

Fig. 11 : An alternate style of tiling

If n is changed to 9, the values of d=3 and 4 are juxtaposed and k is varied in an appropriate range, then this new style of tiling yields the output given in Figure 12. One may compare this to Figure 9.

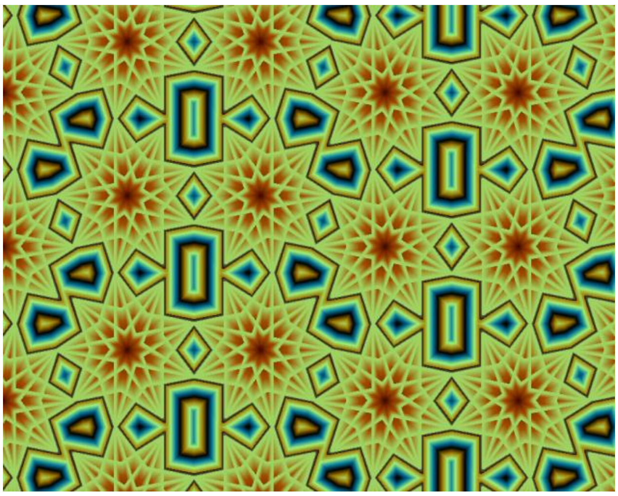

Fig. 12 : The new tiling with juxtaposition and colors

7 . A SPECIAL CASE

When n is a multiple of three, a three-fold symmetric tiling can be constructed as follows: first star(n,d) is constructed with its centre coordinates at (I,j) and radius as k. Then another star(n,d) with the same radius is drawn with its centre coordinates at (i+√(3.)*120*sin((180.-(n+1)/4*ang)*3.14 /180.), j+120*sin((180.-(n+1)/4*ang)*3.14/180.). This creates two touching stars which will be used as a prototype for the tiling. This prototype is displayed in figure 13, for n=9 and d=4.

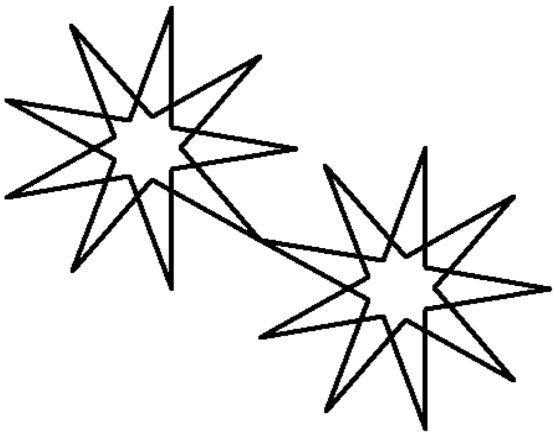

Fig. 13: The prototype for n=9 and d=4

This prototype is then repeated by increasing the center coordinates (i, j) of the first star(n,d) in the prototype by an interval of 2*sqrt(3.)*120*sin((180.-(n+1)/4*ang)*3.14/180.) for the abscissa i and an interval of 2*120*sin((180.-(n+1)/4*ang)*3.14/180.) for the ordinate j. The main program in the code in section 3 above could be changed as follows to accommodate this new style of tiling

```
void main()
{
int n=7;int k=90;float ang=360./n;float sa;
sa=k*cos(ang*3.14/180.);
for(int i=-40;i<1200;

i+=2*sqrt(3.)*120*sin((180.-(n+1)/4*ang)*3.14/180.))

for(int j=-100;j<900;

j+=2*120*sin((180.-(n+1)/4*ang)*3.14/180.))

{star(0,k,n,d,i,j);

star(0,k,n,d,i+sqrt(3.)*120*sin((180.-(n+1)/4*ang)*3.14/180.)

,j+120*sin((180.-(n+1)/4*ang)*3.14/180.));

}
```

The output of this is illustrated in Figure 14. One may compare Figure 14 to Figures 4 and 11

.

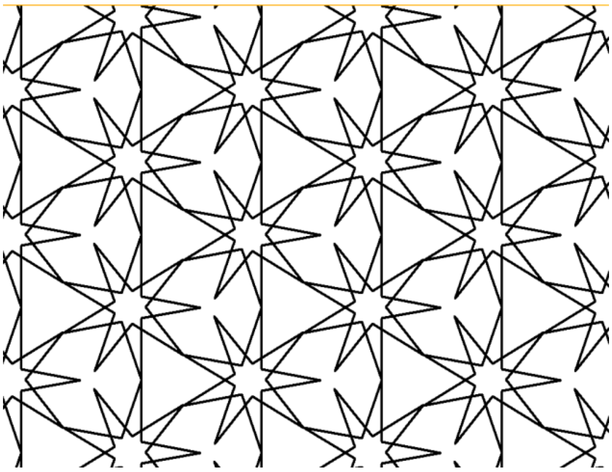

Fig. 14: The tiling for the special case

If n is changed to 9, the values of d=3 and 4 are juxtaposed and k is varied in an appropriate range, then this special style of tiling yields the output given in Figure 15. One may compare this to Figures 9 and 12.

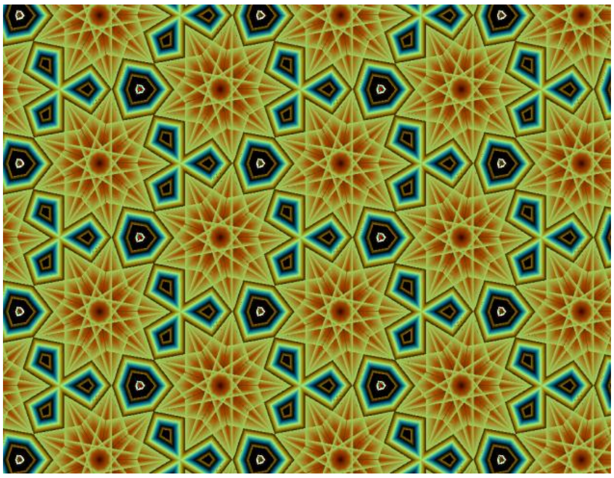

Fig. 14 : The special tiling with juxtaposition and colors

8. CONCLUSION

The tiling with with stars having a nonprime odd number of vertices may yield manifold symmetry.. This, as well as various new styles of tiling with even-vertexed stars would be explored in detail in future work.

9. ACKNOWLEDGMENTS

The author wishes to acknowledge his debt to the referee(s) for their constructive suggestions and encouragement

10. REFERENCES

[1] Syed Jan Abas and Amer Shaker Salman. Symmetries of Islamic Geometrical Patterns. World Scientific, 1995.

[2] J. Bourgoin. Arabic Geometrical Pattern and Design. Dover Publications, 1973. .

[3] Jean-Marc Castera. Zellijs, muqarnas and quasicrystals. In Nathaniel Friedman and Javiar Barrallo, editors,ISAMA 99 Proceedings, pages 99–104, 1999. l.

[4] Jean-Marc Castera et al. Arabesques: Decorative Art in Morocco. ACR Edition, 1999.

[5] A.K. Dewdney. The Tinkertoy Computer and Other Machinations, pages 222–230. W. H. Freeman, 1993.

[6] Douglas Dunham. Artistic patterns in hyperbolic geometry. In Reza Sarhangi, editor, Bridges 1999 Proceedings, pages 139–149, 1999.

[7] Branko Gr¨unbaum and G. C. Shephard. Tilings and Patterns. W. H. Freeman, 1987.

[8] Branko Gr¨unbaum and G. C. Shephard. Interlace patterns in islamic and moorish art. Leonardo, 25:331–339,1992.

[9] E.H. Hankin. Memoirs of the Archaeological Society of India, volume 15. Government of India, 1925.

[10] Craig S. Kaplan. Computer Generated Islamic Star Patterns. In Bridges 2000: Mathematical Connections in Art, Music and Science, 2000.

[11] Craig S. Kaplan and David H. Salesin. Islamic Star Patterns in Absolute Geometry. ACM Transactions on Graphics 23(2):97-119, April 2004.

[12] A.J. Lee. Islamic star patterns. Muqarnas, 4:182–197, 1995.