Construction of Islamic Stars and Rosettes using Turtle Geometry and Tiling Patterns based on Them

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

ABSTRACT
Islamic star and rosette patterns have been extensively studied for their symmetry and aesthetic appeal. This paper presents several new construction methods for stars and rosettes using Turtle geometry and also presents new geometric shapes even more complex than stars and rosettes the construction of which is based on Turtle geometry. It also explores new Turtle geometry based tiling patterns that use these shapes.

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

Keywords
Turtle Geometry, 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 [9], Abas and Salman [1], Bourgoin[3],Dewdney [6], Castera [4], Dunham[7]). 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 [5], Gr¨unbaum and Shephard[8], Kaplan[16]. Kaplan[15] has devised an elaborate construction method for stars based on methods described originally by Henkin [14] and Lee [17]. In addition Kaplan[16] has also devised a number of interesting tiling methods by putting stars in contact. Most of these tiling styles use stars and rosettes with an even number of vertices. Tiling with stars having an odd number of vertices have been studied by Gangopadhyay([10],[11]). In yet another paper(Gangopadhyay[12]), a simpler method to construct a rosette is described. Gangopadhyay[13] has also developed several new generalized tiling methods which use rosettes with both even and odd number of vertices. Although simpler than previous construction methods, the construction of stars and rosettes in [10] and [12], are still somewhat involved in the sense that they use diverse trigonometric functions and a number of specialized functions based on co-ordinate geometry. In this paper, a new, much simpler construction method is presented. The new method uses Turtle geometry (Abelson and diSessa[2]) to construct stars and rosettes. In addition, new geometric shapes even more complex than stars and rosettes are constructed using simple methods involving Turtle geometry. Further, new Turtle geometry based tiling patterns using these shapes are also explored. These are the main distinguishing features of this paper.

2. BASIC TURTLE GEOMETRIC FUNCTIONS
Given below are the basic functions of Turtle geometry along with a brief description of the task that each function performs.

Here px and py denote cursor positions and ang denotes the angle at which the cursor will move. Ps is a binary variable denoting whether pen is up(ps=1) or down(ps=0).

```
void fd(float dist)
{float hx=cos(ang*3.1415926536/180);
float hy=sin(ang*3.1415926536/180);
float nx=px-hx*dist;
float ny=py-hy*dist;
if(ps!=1)goto label;
line(px,py,nx,ny);
label:px=nx;py=ny;
}
```

The function ‘fd’ draws a line of length ‘dist’ from the present position of the cursor(px,py) at an angle ‘ang’.

```
float rt(float l)
{ang+=l;
return ang;}
```

The function ‘rt’ increases the angle ‘ang’ by l degrees.

```
float lt(float l)
{ang-=l;
return ang;}
```

The function ‘lt’ decreases the angle ‘ang’ by l degrees.

```
void movexy(int x,int y)
{px=x;py=y;ang=90;}
```

The function movexy psitions the cursor at (x,y) at angle 90 degrees.

```
void pu()
{ps=0;}
```

The function pu disables the line drawing function ‘line’ in the function ‘fd’.

```
void pd()
{ps=1;}
```

The function pd enables the line drawing function ‘line’ in the function ‘fd’.

3. HOW TO DRAW A STAR
The function star given below uses the Turtle geometric functions described in section 2 to construct an Islamic star for n vertices, where n is >7 and is a factor of 360. The

function has three arguments, viz, n – the number of vertices, a – the length of a side in the inner polygon of the star and angy – the angle supplementary to the external angle between two adjacent sides of the inner polygon. The function constructs an Islamic star in two steps. First the inner polygon is constructed; then the outer polygon is constructed after positioning the cursor appropriately through right rotation at an appropriate angle.

```
void star(int n, int a, float angy)
{ float t=180+angy-(2*n-4)*90/n;
        for(int i=0;i<n;i++)
        {fd(a);rt(angy);fd(a);lt(t);}
        rt(t);
        for(int i=0;i<n;i++)
        {fd(a*sin((90-angy/2)*3.14/180)/sin((90-
        t+angy/2)*3.14/180)); lt(2*t-angy);
        fd(a*sin((90-angy/2)*3.14/180)/sin((90-
        t+angy/2)*3.14/180)); rt(t);}
}
```

In figure 1, we illustrate the construction of a star with n=9, a=30 and angy =80. The inner polygon has been drawn in red color to explain the construction.

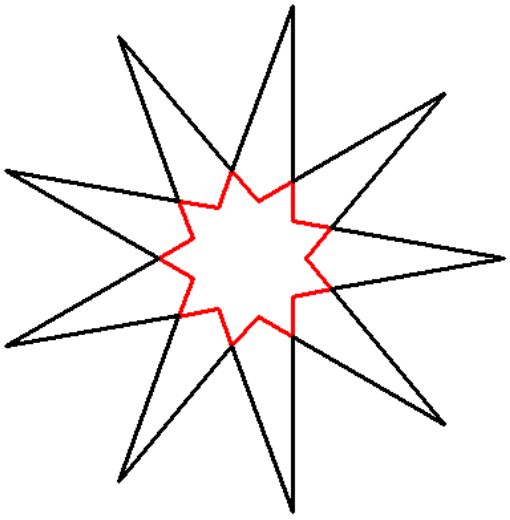

Fig 1 : The output of star(9, 30,80)

4. HOW TO DRAW A ROSETTE

To draw a rosette, first a star is drawn following the method described in section 3. Every rosette has a star inscribed inside. In fact, it is simply a symmetric outer polygon circumscribed on the star it inscribes. So the cursor is first moved to the nearest tip(a vertex in the outer polygon) of the star. Then the circumscribing poly gon is drawn. The entire method is codified below in the function rose. The function rose has five arguments, the first three of which are the three arguments –n, a and angy of the inscribed star. The remaining two arguments of the function rose refer to the circumscribing polygon. They are d – the length of the shorter side of the polygon and s, where 2s is the angle between two adjacent longer sides of the polygon. Here n is >7 and is a factor of 360.

Below the code for the function rose is presented.

```
void rose(int n, int a, float angy, int d, float s)
{ star(n, a, angy);
float t=180+angy-(2*n-4)*90/n;
fd(a*sin((90-angy/2)*3.14/180)/sin((90-t+angy/2)*3.14/180));
setcolor(3);
float b=a*sin((90-angy/2)*3.14/180)/
sin((90-t+angy/2)*3.14/180);
for(int i=1;i<=n;i++)
        {lt(3*t/2-angy-90);fd(d);lt(s);fd(b*sin((180-
t)/2*3.14/180)/sin(s*3.14/180));lt(180-2*s);
        fd(b*sin((180-
t)/2*3.14/180)/sin(s*3.14/180));lt(s);fd(d);
        rt(90+t/2);}
}
```

In figure 2, we illustrate the construction of a rosetter with n=9, a=30, angy =60, d=30 and s=40. The circumscribing polygon has been drawn in cyan to explain the construction.

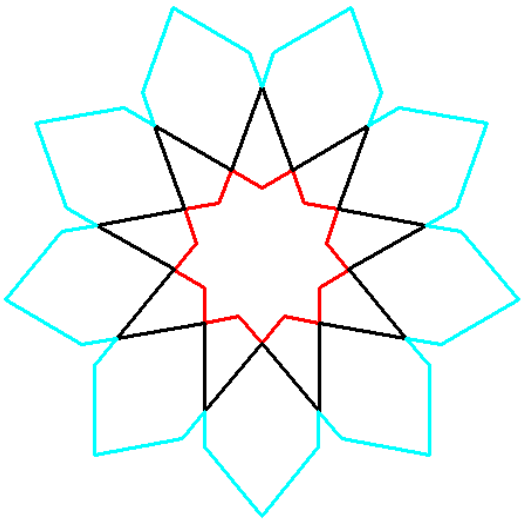

Fig 2 : The output of rose(9, 30,60,30,40)

5. MORE COMPLEX CONSTRUCTIONS

The function polygram, presented below, constructs symmetric shapes more involved than stars or rosettes. The function has two arguments, n – the number of vertices of the shape and a – the length covered by the function fd in every step. Here n is always greater than 7, but need not be a divisor of 360.

Below the code for the function polygram is presented.

```
void polygram(int n, int a)
{
for(int i=0;i<n;i++)
{
fd(a);lt(360./n);fd(a);rt(360./n);
```

```
for(int j=0;j<=n-2;j++)
{fd(a);lt(360./n);fd(a);rt(360./n);fd(a);lt(360./n);}
}
}
```

The output of the function for n=9 and a=30 is given in figure 3.

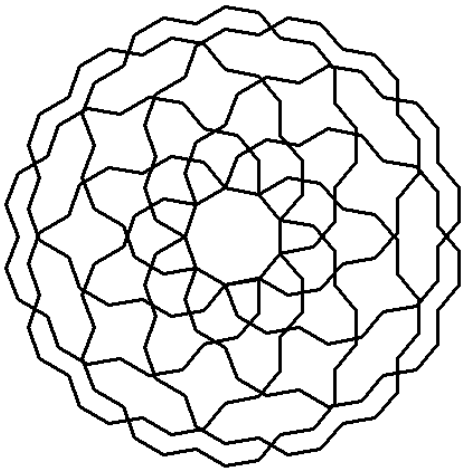

Fig 3 : The output of polygram(9, 30)

Another such symmetric construction is presented in the function polygram1, which is similar to the earlier function polygram, but has an additional argument, angy – which is an angle of deviation.

Below the code for the function polygram1 is presented. Here n is greater than 7 and is not a multiple of 3.

```
void polygram1(int n,int a,float angy)
{float b=a*sin(360./n*3.14/180);
for(int i=0;i<n;i++)
{
{fd(a);lt(360./n-angy);fd(b);lt(720./n+angy);fd(b);lt(360./n-angy);fd(a);
}
rt(360./n-angy);
}
}
```

The output of the function polygram1 for n=11, a=70 and angy =-30 is given in figure 4 Figure 5 depicts the output of polygram1(8, 70, 20);

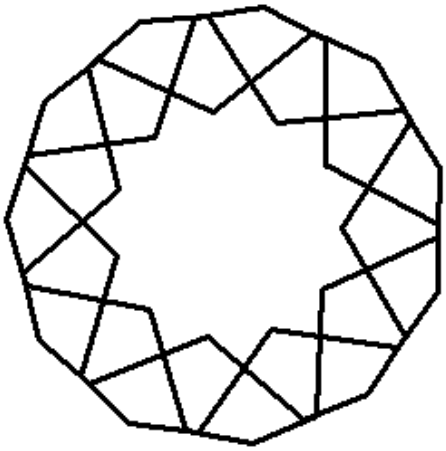

Fig 4 : The output of polygram(11, 70, -30)

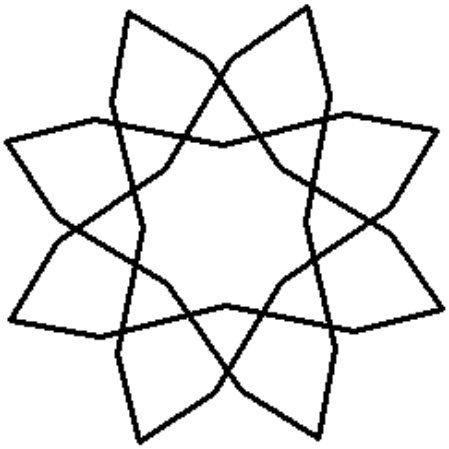

Fig 5 : The output of polygram(8, 70, 20)

6. TILING WITH STARS, ROSETTES AND OTHER SYMMETRIC STRUCTURES USING TURTLE GEOMETRY

The main building block of the first tiling algorithm is the output of the simple function rose8 which constructs an 8-vertex rosette. The function rose8 is presented below:

```
void rose8(int a)
{for(int j=0;j<8;j++)
{for(int i=0;i<6;i++)
{fd(a);rt(90);}
rt(135);}
}
```

The output of rose8 for a= 70 is shown in figure 6.

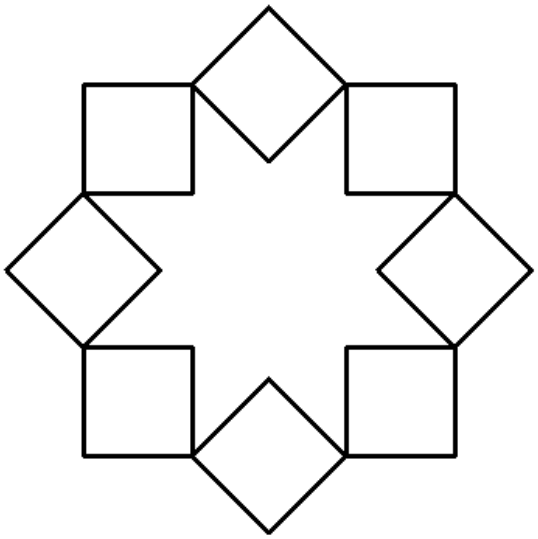

Fig 6: The output of rose8(70)

Next, the function rose8 is embedded in a triple loop to generate the tiling. The code segment for this is given below and the output is shown in figure 7.

```
for(int k=0;k<6;k++)
for(int l=0;l<6;l++){
movexy(k*234,l*234);
for(int i=0;i<8;i++)
{star8(20);pu();fd(40);lt(45);fd(40);pd();}}
```

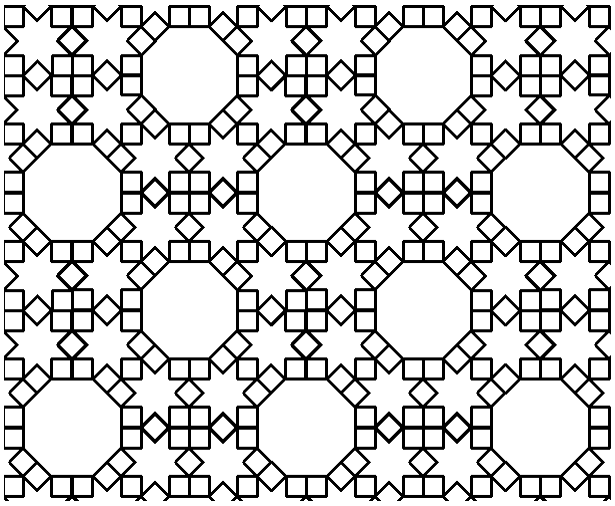

Fig 7: Tiling using the output of rose8(20)

Another tiling algorithm that uses a more involved symmetric structure calls the simple function octagram that draws an 8-vertex symmetric shape. The function octagram has a single argument a, which is the distance traversed every time the function fd is called.

```
void octagram(int a)
{for(int i=0;i<8;i++)
{fd(a);rt(45);
for(int j=0;j<6;j++)
{fd(a);lt(45);
}
}
}
```

The output of the function octagram for a= 50 is given in figure 8.

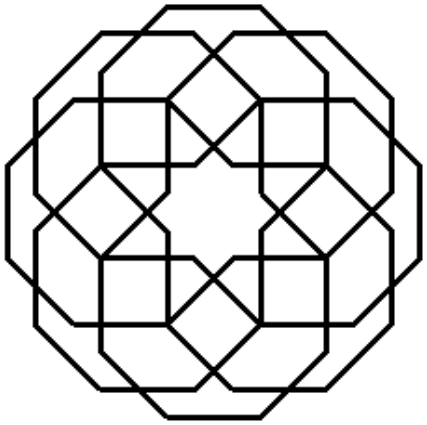

Fig 8 : The output of octagram(50)

Next, the function octagram is embedded in a double loop to generate the tiling. The code segment for this is given below and the output is shown in figure 9.

```
for(int i=0;i<8;i++)
for(int j=0;j<8;j++)
{movexy(i*190,j*190);
octagram(50);
}
```

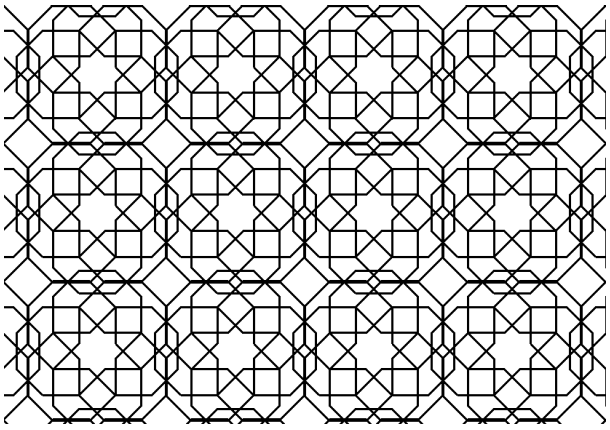

Fig 9: Tiling using the output of octagram(50)

.*ROSETTE WITH SIX VERTICES AND A TILING PATTERNS USING THEM*

The function rose6, presented below, generates a rosette on six vertices which is based on Turtle geometry. The function draws two symmetric polygons,. The inner polygon, drawn in black has 12 sides of length a, and the outer polygon, drawn in

red has 24 sides of length a, where 'a' is the single argument of the function.

```
void rose6(int a)
{for(int i=0;i<6;i++)
{fd(a);lt(120);fd(a);rt(60);
}
fd(a);setcolor(4);
for(int i=0;i<6;i++)
{for(int j=0;j<3;j++)
{fd(a);lt(60);}
fd(a);rt(120);
}
}
```

The output of the function rose6 for a=30 is given in figure 10.

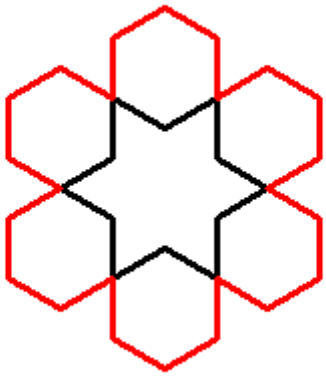

Fig 10 : The output of rose6(30)

Next, the function rose6 is embedded in a double loop to generate the tiling. The code segment for this is given below and the output is shown in figure 11

```
for(int k=0;k<9;k++)
for(int l=0;l<8;l++)
{if(l%2==0)movexy(102*k,92*l);
else movexy(102*k+51,92*l);
rose6(30);
}
```

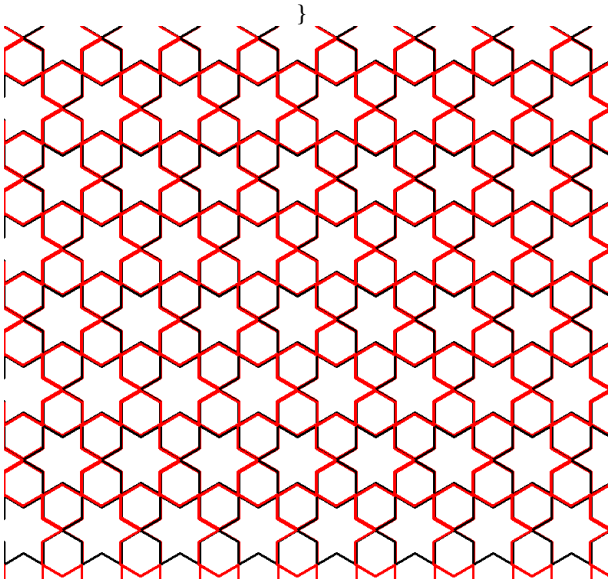

Fig 11 : Tiling using the output of rose6(30)

7. CONCLUSION

Turtle geometry based constructions for stars and rosettes where the number of vertices is not a divisor of 360 are yet to be constructed. These, as well as further new styles of tiling based on turtle geometry 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] Harold Abelson and Andrea diSessa, Turtle Geometry,The MIT press,1986.

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

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

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

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

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

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

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

[10] T. Gangopadhyay, On Tiling Patterns Involving Islamic Stars with an Odd Number of Vertices, International journal of computer applications, Vol. 65, number 8, 39-44.

[11] T. Gangopadhyay, Further Tiling Patterns Involving Islamic Stars with an Odd Number of Vertices, International journal of computer applications, Vol. 67, number 1, 12-16

[12] T. Gangopadhyay, On Tiling Patterns Involving Islamic rosettes with an Odd Number of Vertices, International journal of computer applications, Vol. 69, number 9, 9-14,

[13] T. Gangopadhyay, Further Tiling Patterns Involving Islamic rosettes with an Odd Number of Vertices, International journal of computer applications, Vol. 71, number 6, 36-41.

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

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

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

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

.