Algorithmic Generation of Chinese Lattice Designs

S. Y. Lee and K. M. Tiong

***Abstract*—Chinese lattice designs commonly appear on traditional Chinese windows, doors and decorative ornaments.The underlying structure of Chinese latticedesigns are mathematically interesting and present wonderful opportunities for detailed investigation. A closer look reveals that many of these Chinese lattice designs can be algorithmically designed using some simple procedures. In this paper, we look at the algorithmic design procedure of some Chinese lattice designs sourced from Daniel Sheets Dye's definitive book on Chinese lattices using the turtle graphics functionality of MuPAD®, a MATLAB® Symbolic Math Toolbox. We also generate random multi-colored Chinese lattice designs.**

***Index Terms*—Algorithmic, chinese lattice design, turtle graphics, MuPAD®.**

I. Introduction

Chinese lattice designs are mathematically interesting structural art. They can be commonly found on traditional Chinese windows, doors and decorative ornaments. Chinese lattice designs are found in Far East countries like China, Korea, Japan and even in Vietnam.Chinese lattices are made of wood, which is not long-lasting, and thus many Chinese lattices have not survived the passing of time. However, evidence of Chinese lattices is found in pictures or reliefs on old bronzes, ancient ceramics and things found in graves.Some physical examples of Chinese lattice designs can be found in [1] and [2].Interestingly, Chinese lattice designs are now even used in modern architectural designs as a building façade [3].

There are very few publications on Chinese lattice designs. The most important work on Chinese lattice designs is Daniel Sheets Dye's book [4]which contains the most comprehensive description and collection on Chinese lattice designs constructed between 1000BC and 1900AD. Dye classified Chinese lattices into 26 groups according to common visual features.

Investigations on Chinese lattice designs have previously centered on the perspective of shape grammar that describes the composition procedures of lattice designs and generates existing lattice designs as well as new hypothetical designs of similar style [5]-[7]. In several papers, Majewski and Wang investigated Chinese lattice designs from an algorithmic [1], [8] and symmetry groups point of view [2], [9].The algorithmic investigation using MuPAD® turtle graphics in [1] was introductory in nature and limited to only three examples of Chinese lattice designs.

In this paper, we looked at more Chinese lattice designs from the different group classifications of Dye (12 groups were selected with a total of 25 Chinese lattice designs). The Chinese lattice designs were then algorithmically generated using MuPAD® turtle graphics. We added a further element by coloring the Chinese lattices.

II. MuPAD® Turtle Graphics

A. Turtle Graphics

Turtle graphics can be simply described as the mimicking of finite straight line movements by an abstract pen through a sequence of commands.

B. MuPAD® Commands

The following table from [10] shows the turtle graphics command of MuPAD®.

TABLE I: MuPAD® Commands

Command	Method	Description
Left(*angle*)	left(*angle*)	Turn left given angle (in radians)
Right(*angle*)	right(*angle*)	Turn right given angle (in radians)
Forward(*angle*)	forward(*length*)	Draw forward a line of given length
Up	penUp()	Take the pen up
Down	penDown()	Take the pen down
Push	push()	Save the current stage of the turtle
Pop	pop()	Move the turtle to the last stage
LineColor (*color*)	setLineColor(*color*)	Change the current path color

To familiarize with MuPAD®'s turtle graphics, simple examples of implementation can be examined in [11].

III. Algorithmic Process of Chinese Lattice Designs

When representing the Chinese lattice designs algorithmically, the first step is to determine the design's basic building block. The basic building block may consist of more than one simpler building block. Each simpler building block will be drawn using a procedure. The procedures for the simpler building blocks will be combined to form the basic building block. This combined procedure will then be repeated with *for-loop* to build the whole Chinese lattice design structure.

We describe the algorithmic process through the Out-Lock example below.

The basic building block of the Out-Lock pattern in Fig. 1 is shown in Fig. 2. It can be observed that it is made up of many rectangular shapes arranged in a certain order. Table II

Manuscript received June 18, 2013; revised August 28, 2013.

S. Y. Leewas is with the School of Science and Technology, Universiti Malaysia Sabah, 88999 Kota Kinabalu, Sabah, Malaysia.

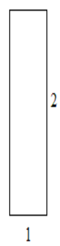
K. M. Tiong is with the Faculty of Arts and Social Sciences, University of Nottingham Malaysia Campus, 43500 Semenyih, Selangor, Malaysia (e-mail: KungMing.Tiong@nottingham.edu.my).

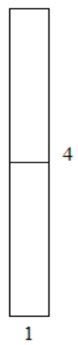
DOI: 10.7763/IJCCE.2013.V2.279

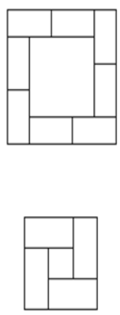
shows the MuPAD® codes for drawing the basic building block for the Out-Lockpattern.

TABLE II: MUPAD® CODE FOR THE BASIC BUILDING BLOCK OF THE OUT-LOCK PATTERN

MuPAD® code	Description	Result

x1:=proc(U)
begin
forward(U,2);
U::right(PI/2);
forward(U,1);
U::right(PI/2);
forward(U,2);
U::right(PI/2);
forward(U,1);
U::right(PI/2);
return(U)
end:
	Produces a rectangle with sides of two and one units	

x2:=proc(U)
begin
U::push():
forward(U,4);
U::right(PI/2);
forward(U,1);
U::right(PI/2);
forward(U,4);
U::right(PI/2);
forward(U,1);
U::right(PI/2);
Upforward(U,2);
U::right(PI/2);
forward(U,1);
U::pop():
return(U)
end:
	Produces a rectangle with sides of two and one units	

y1:=proc(U)
local i,j;
begin
U::push():
for i from 1 to 4 do
x2(U);
Upforward(U,5);
U::right(PI/2);
end_for:
Upforward(U,1);
U::right(PI/2);
Upforward(U,1);
U::left(PI/2);
for j from 1 to 4 do
x1(U);
Upforward(U,3);
U::right(PI/2);
end_for:
U::pop():
return(U)
end:
	In this procedure, two *for-loop* are used. The first *for-loop* produces the pattern on the right (above). The second *for-loop* produces the pattern on right (below). The whole code will then be combined to produce the pattern in Fig. 2.	

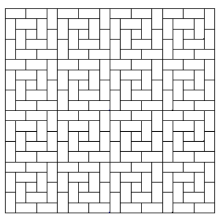


Fig. 1. Out-lock pattern.

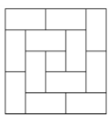
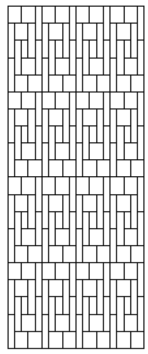


Fig. 2.basic building block of the out-lock pattern.

The next step is to repeatedly produce the basic building block to produce the overall Out-Lock pattern in Fig. 1. Table III shows the MuPAD® codes for replicating the basic building block of the Out-Lock pattern to form the complete pattern.

TABLE III: MUPAD® CODE TO REPLICATE THE BASIC BUILDING BLOCK OF THE OUT-LOCK PATTERN TO FORM A COMPLETE PATTERN

MuPAD® code	Description	Result

T:=plot::Turtle():
T::push():
for i from 0 to 3 do
T::pop():
T::push():
Upforward(T,i*5);
for j from 0 to 3
do
y1(T);
T::right(PI/2);
 	
Upforward(T
,5);
T::left(PI/2);
end_for:
end_for:
 	
plot(T)
	The first for-loop is to repeat the basic building block by row whereas the next *for-loop* is to repeat the basic building block by column. *plot(T)* then produces the final complete pattern.	

In the code above (Table III), we defined a new command "Upforward" where the abstract pen moves a certain finite length without drawing the line. This is to simplify the pen up – move – pen down movements which are repeatedly used.The codes are shown in Table IV. To produce a multi-colored effect for the Chinese lattice, we randomized the line color with codes described in Table V.

TABLE IV: MUPAD® CODE TO SIMPLIFY THE PEN UP – MOVE – PEN DOWN MOVEMENTS WITHOUT DRAWING A LINE

MuPAD® code	Description
Upforward:=proc(V,dist) begin V::penUp(); V::forward(dist); V::penDown(); return(V) end:	*Pen comes up, moves a certain length (without drawing the line) and then pen goes down*

TABLE V: MUPAD® CODE TO RANDOMIZE THE LINE COLORS

MuPAD® code	Description

forward:=proc(V,dist)
local a,b,c;
begin
a:=frandom();
b:=frandom();
c:=frandom();
V::setLineColor([a,b,c]);
V::forward(dist);
return(V)
end:
	Produces random colored lines

The complete program codes for this Out-Lock pattern is given in the Appendix. The program generates a multi-colored Chinese lattice as in Fig. 3.

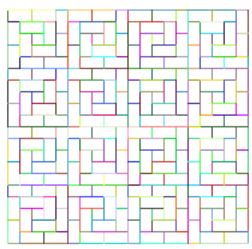


Fig. 3. Multi-colored out-lock pattern.

IV. CHINESE LATTICES DESIGNS GENERATED

Following the principles and steps in Section III, we generated the following Chinese lattice designs.

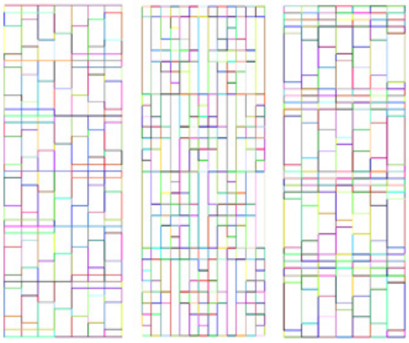


Fig. 4. Parallelogram.

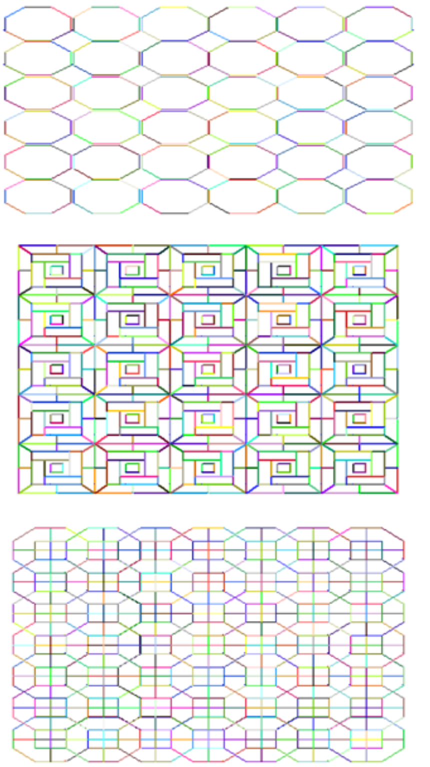


Fig. 5. Octagon or octagon square.

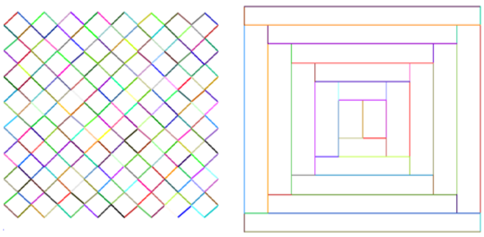


Fig. 6. Single focus frames.

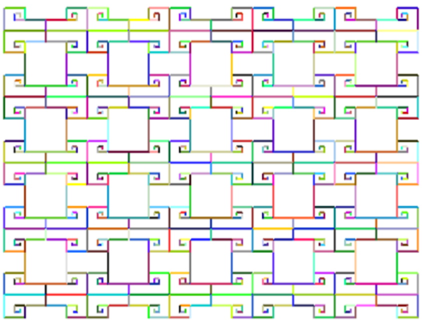


Fig. 7. Wedge-*lock*.

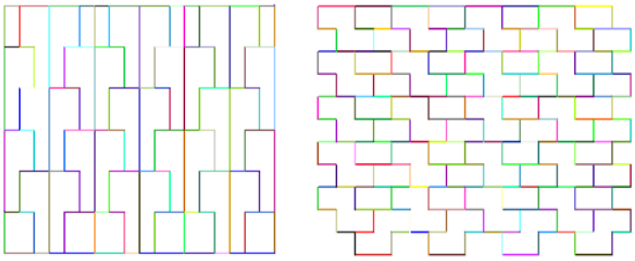


Fig. 8. Presentation.

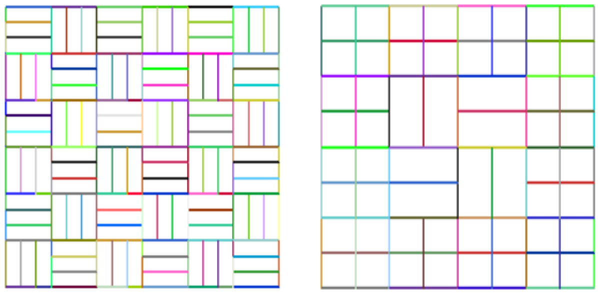


Fig. 9. Out-lock.

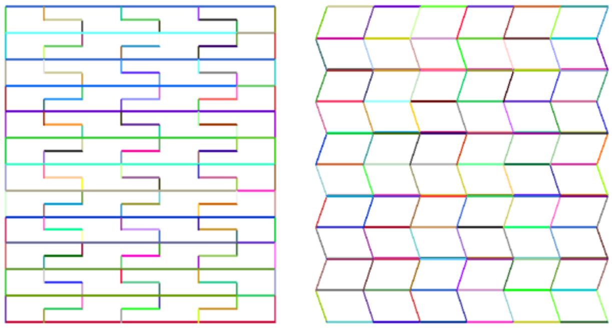


Fig. 10. Parallel waves.

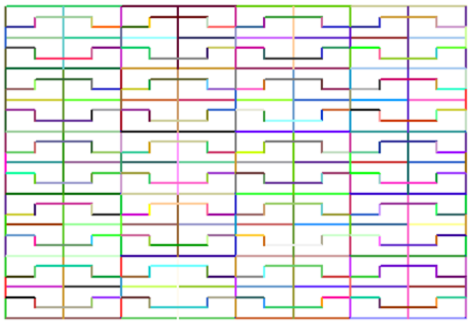


Fig. 11. Opposed waves.

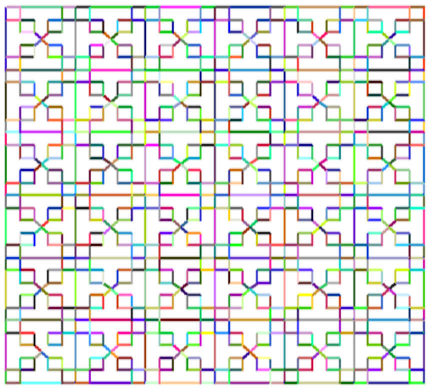


Fig.12. Recurving Wave.

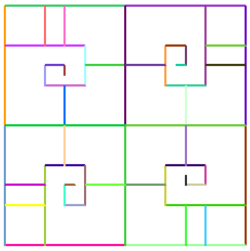


Fig. 13. S-scroll.

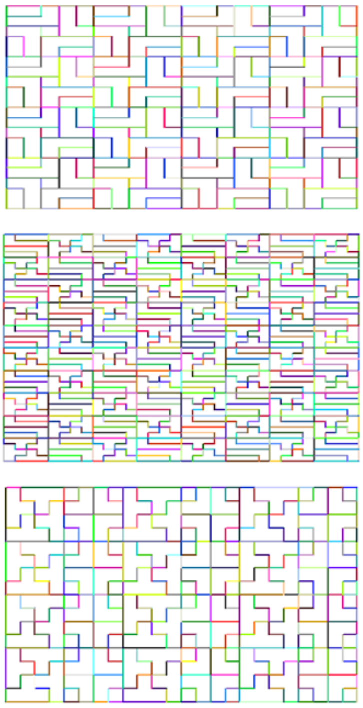


Fig. 14. Like swastikas.

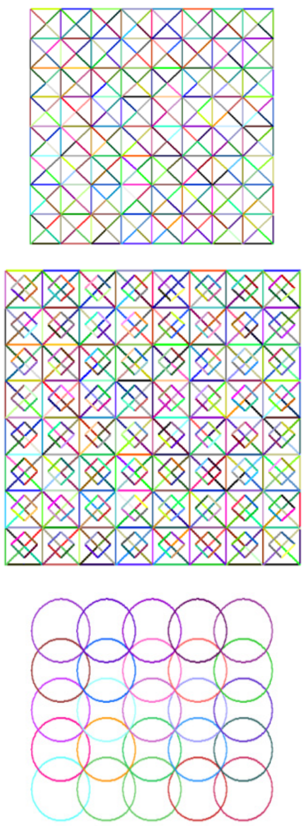


Fig. 15. Square and Round.

V. Conclusion

Many Chinese lattice designs can be successfully generated through an appropriate algorithmic process. For this type of Chinese lattices, the lattice must be broken into smaller basic building blocks. This requires a sharp observation of the Chinese lattice design's underlying structure. It should be noted however that not all Chinese lattices can be generated using this algorithmic process. Rustic ice-ray and symmetrical ice-ray lattices are two examples where MuPAD®'s turtle graphics may not be able to construct successfully. A discussion of these is beyond the scope of this paper.

A randomized coloring of the lines in a Chinese lattice's design produces appealing multi-colored Chinese lattices.

Appendix

Complete MuPAD® codes to generate the Out-Lock pattern in Fig. 3

```
forward:=proc(V,dist)
local a,b,c;
begin
a:=frandom();
b:=frandom();
c:=frandom();
  V::setLineColor([0,0,0]);
  V::forward(dist);
return(V)
end:


Upforward:=proc(V,dist)
begin
  V::penUp();
  V::forward(dist);
  V::penDown();
return(V)
end:

x1:=proc(U)
begin
  forward(U,2);
    U::right(PI/2);
  forward(U,1);
    U::right(PI/2);
  forward(U,2);
    U::right(PI/2);
  forward(U,1);
    U::right(PI/2);
return(U)
end:

x2:=proc(U)
begin
U::push():
  forward(U,4);
    U::right(PI/2);
  forward(U,1);
    U::right(PI/2);
  forward(U,4);
    U::right(PI/2);
  forward(U,1);
    U::right(PI/2);
```

```
  Upforward(U,2);
    U::right(PI/2);
  forward(U,1);
U::pop():
return(U)
end:

y1:=proc(U)
local i,j;
begin
U::push():
for i from 1 to 4 do
  x2(U);
  Upforward(U,5);
    U::right(PI/2);
end_for:
Upforward(U,1);
U::right(PI/2);
Upforward(U,1);
U::left(PI/2);
  for j from 1 to 4 do
    x1(U);
    Upforward(U,3);
      U::right(PI/2);
  end_for:
U::pop():
return(U)
end:

T:=plot::Turtle():
T::push():
for i from 0 to 3 do
T::pop():
T::push():
Upforward(T,i*5);
  for j from 0 to 3 do
    y1(T);
      T::right(PI/2);
    Upforward(T,5);
      T::left(PI/2);
  end_for:
end_for:
  plot(T)
```

The program codes for the other Chinese lattice designs produced in this paper can be furnished upon request from the authors.

References

[1] M. Majewski and J. Y. Wang, “An algorithmic approach to Chinese lattice design,” *Electronic Journal of Mathematics and Technology*, vol. 3, no. 1, pp. 83-101, February 2009.

[2] M. Majewski and J. Y. Wang, “Chinese lattice design from symmetry groups point of view,” *Electronic Journal of Mathematics and Technology*, vol. 4, no. 2, pp. 101-115, June 2010.

[3] Architecture – European China Centre. [Online]. Available: http://www.europeanchinacentre.com/pages/architecture.aspx

[4] D. S. Dye, *Chinese Lattice Designs*, New York: Dover Publications Inc., 1974.

[5] T. Knight, “Applications in architectural design, and education and practice,” report of the NSF/MIT Workshop on Shape Computation, April, 1999

[6] G. Stiny, “Ice-ray: a note on the generation of Chinese lattice designs,” *Environment and Planning B*, vol. 4, pp. 89-98, 1977.

[7] M. Tapia, “Chinese lattice designs and parametric shape grammars,” *The Visual Computer*, vol. 9, pp. 47-56, 1992.

[8] M. Majewski and J. Y. Wang, “Deconstructing Chinese lattices with MuPAD,” presented at ATCM conference, 2008.

[9] M. Majewski and J. Y. Wang, “A journey through Chinese windows and doors – an introduction to Chinese mathematical art,” presented at ATCM conference, 2009.

[10] M. Majewski, *MuPAD Pro Computing Essentials*, 2nd ed, Berlin: Springer- Verlag, 2004, pp. 300.

[11] MATLAB® turtle graphics. [Online]. Available: http://www.mathworks.com/help/symbolic/mupad_ref/plot-turtle.html

S. Y. Lee was formerly an undergraduate student at the School of science and technology, Universiti Malaysia Sabah. His undergraduate dissertation in 2008 was an investigation on Chinese lattice designs using MuPAD® and he graduated with a B.Sc. in Mathematics with Computer Graphics under the supervision of the second author. He now works as a programmer.

K. M. Tiong is currently an assistant professor at the faculty of arts and social sciences, The University of Nottingham Malaysia Campus at Semenyih, Selangor, Malaysia. He holds a B.Sc.Ed. (Hons) (Mathematics and Physics) from University of Malaya and M.Sc. (Mathematics) from Universiti Teknologi Malaysia. His current research interests involve mathematical patterns, mathematical games and statistics education.