

# Mojette Transform with non classical cubic lattices

Sen LIU

25th August 2013

Master Multimedia and Data Management (MDM)  
Supervisor: Nicolas Normand, Vincent Ricordel

## Abstract

Nowadays projection technologies are more and more frequently used and noticed, especially in medical tomography and network packets transfer. But all these technologies are based on the ordinary square or cubic lattices. In this thesis, we will propose a new way of Mojette transform in which the hexagonal lattice will be used. At the beginning, we will introduce some basic knowledge related to our topic. They are Dirac and Spline 0 Mojette transform and their back projection as well as the fundamental definitions and properties about the lattices in both 2-Dimension and 3-Dimension. And then we will address the method of generating the Farey Series in hexagonal lattice which will be used as the discrete directions of the projections. And we will also represent the algorithm of Dirac and Spline 0 Mojette transform in hexagonal lattice. Eventually, we will show the results of experiments which compare the projected bins in hexagonal lattice with those in square lattice and ground truth.

**Keyword:** Mojette transform, hexagonal lattice, Farey Series.

## Résumé

Aujourd'hui, les technologies de projection sont de plus en plus fréquemment utilisés et remarqué, en particulier dans la tomographie médicale et réseau de transfert de paquets. Mais toutes ces technologies sont basées sur la place ordinaire ou réseaux cubiques. Dans cette thèse, nous allons proposer une nouvelle façon de Mojette transformation dans laquelle le réseau hexagonal sera utiliser. Au début, nous allons introduire quelques notions de base relatives à notre sujet. Ils sont Dirac et Spline 0 Mojette transformation, leur projection et les définitions et propriétés fondamentales sur les réseaux dans tous les deux 2-Dimension et 3-Dimension. Et puis nous aborderons la méthode de génération de la Série Farey en réseau hexagonal qui sont les directions discrètes des projections. Et nous allons aussi représenter l'algorithme de Dirac et Spline 0 Mojette transformée en réseau hexagonal. Finalement, nous allons montrer les résultats des essais qui comparent les bins en réseau hexagonale avec ceux en réseau carré et le données réalité.

**Mots-clés :** Mojette transform, réseau hexagonal, Série de Farey

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	IRCCyn Lab and IVC Team . . . . .	5
1.2	Aspects of the topic . . . . .	5
<b>2</b>	<b>Bibliography</b>	<b>6</b>
2.1	Mojette Transform . . . . .	6
2.1.1	Direct Transform . . . . .	6
2.1.2	Inverse Transform . . . . .	8
2.1.3	Farey Series . . . . .	13
2.2	Lattices . . . . .	13
2.2.1	Definition . . . . .	13
2.2.2	Examples of problems involving lattices . . . . .	16
2.2.3	Several important lattices and their illustrations . . . . .	20
<b>3</b>	<b>Works and Algorithms</b>	<b>28</b>
3.1	Farey Series in Hexagonal Lattice . . . . .	28
3.2	Dirac Mojette Transform in Hexagonal Lattice . . . . .	30
3.3	Spline0 Mojette Transform in Hexagonal Lattice . . . . .	30
3.3.1	Transformation of the Points . . . . .	30
3.3.2	Method of Generating Kernel . . . . .	32
3.3.3	Convolution with the Kernel . . . . .	44
3.4	Interpolation in Voronoi Cells . . . . .	45
3.4.1	Shepp-Logan Phantom . . . . .	45
3.4.2	Shepp-Logan in Square Lattice . . . . .	45
3.4.3	Shepp-Logan in Hexagonal Lattice . . . . .	46
3.5	Conclusion of the Jobs . . . . .	50
<b>4</b>	<b>Experiments</b>	<b>51</b>
4.1	Getting Ground Truth . . . . .	51
4.2	The Experiments in Circles . . . . .	52
4.2.1	The Effect of Valid Points . . . . .	52
4.2.2	The CC on Different Directions . . . . .	54
4.3	The Experiments in One Ellipse . . . . .	55
4.4	The Experiments in Shepp-Logan Phantom . . . . .	56
4.5	Conclusion of Experiments . . . . .	58
<b>5</b>	<b>Conclusion</b>	<b>59</b>

# 1 Introduction

## 1.1 IRCCyn Lab and IVC Team

IRCCyN, which stands for Institut de Recherche en Communications et Cybernétique de Nantes, is a Joint Research Unit, and it has been recognized and granted by the CNRS for more than 40 years. It consists of several research teams. One of them is the Image and video-communication (IVC) research group which focuses on digital multimedia processing and analysis.

Most of the processes of digital multimedia processing are studied in our group, from the data acquisition to its visualization, including data transmission coding or storage.

A wide range of research areas are covered around three main lines of work, namely perception, communication and representation.

## 1.2 Aspects of the topic

The Mojette Transform is defined as a discrete exact Radon transform. The direct and inverse Mojette Transforms have been proposed for the cubic lattice.

The project aims at explore them with non classical cubic lattices, for example with the hexagonal lattices in dimension2, or with orthorhombic lattices in dimension 3.

We all know that the distances between the vertices and gravity centroids in cubic lattices are much greater than the ones between the edges and gravity centroids in the hexagonal lattices. The goal of the topic is to see if there is a significant improvement when we change the lattices of the Mojette Transform in dimension2 and dimension 3. And in this thesis, we will not only introduce the way of computing Farey Series<sup>1</sup> in Hexagonal lattice compared to just in Square lattice, but also introduce the methods of projection in Hexagonal lattice. At the end of the thesis, we will express the results of experiments which are compared with both the ordinary Square lattice and the ground truth.

In order to achieve the mentioned goals, some knowledge is required such as the knowledge about the Mojette Transform and also the information about the lattice. The following sections will focus on the details of the knowledge, and it will be helpful for us to understand the topic.

---

<sup>1</sup>see the book, note for mark

## 2 Bibliography

### 2.1 Mojette Transform

The Mojette transform is an exact, discrete form of the Radon transform defined for specific 'rational' projection angles. Like the classical Radon transform, the Mojette transform represents the image as a set of projections. However, the Mojette transform has an exact inverse from a finite number of discrete projections (as few as 1 depending on the angle set). The rational projection angles  $\theta_i$  are defined by a set of discrete vectors ( $p_i$  and  $q_i$ ) as  $\tan \theta_i = q_i/p_i$ . These vectors must respect the condition that  $p_i$  and  $q_i$  are coprime (i.e.  $\gcd(p_i, q_i) = 1$ ) and since  $\tan$  is  $\pi$ -periodic,  $q_i$  is restricted to be positive except for the case  $(p_i, q_i) = (1, 0)$ .

Another principle difference from the classical Radon transform is the sampling rate on each projection, which is no longer constant but depends on the chosen angle as  $1/\sqrt{p_i^2 + q_i^2}$ . The number of bins  $B(i)$  for each projection depends on the chosen direction vector  $(p_i, q_i)$ , and for a  $P \times Q$  image is found to be

$$B(i) = (Q - 1)|p_i| + (P - 1)|q_i| + 1, \quad (1)$$

The computational complexity of the Mojette transform for a  $P \times Q$  image with  $I$  projections is  $O(PQI)$ .

#### 2.1.1 Direct Transform

2 kinds of Mojette transform are used now: *the Dirac Mojette Transform* and *the Spline Mojette Transform*. In the following paragraphs these 2 kinds of Mojette transform will be discussed.

**Dirac Mojette Transform** The dirac Mojette transform, in general form, can be expressed as follows:

$$[\mathcal{M}f](b, p_i, q_i) = \text{proj}_{\delta_{p_i, q_i}}(b) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f(k, l) \Delta(\underbrace{b + kq_i - p_i l}_n), \quad (2)$$

where  $(k, l)$  defines the location of an image pixel,  $b$  is the index of a bin and  $\Delta(n)$  is the Kronecker delta function equal to 1 when  $n = 0$  and zero otherwise. The equation  $b = -kq_i + p_i l$  represents the line of projection, i.e. the set of projected pixels.  $\Delta(b + kq_i - p_i l)$  is therefore equal to 1 only for those pixels which are on the projection line.

From (2), we can see that the transform domain of an image is a set of projections where each element (called a bin as in tomography) corresponds to the sum of the pixels centered on the line of projection. The details can be seen in Figure 1

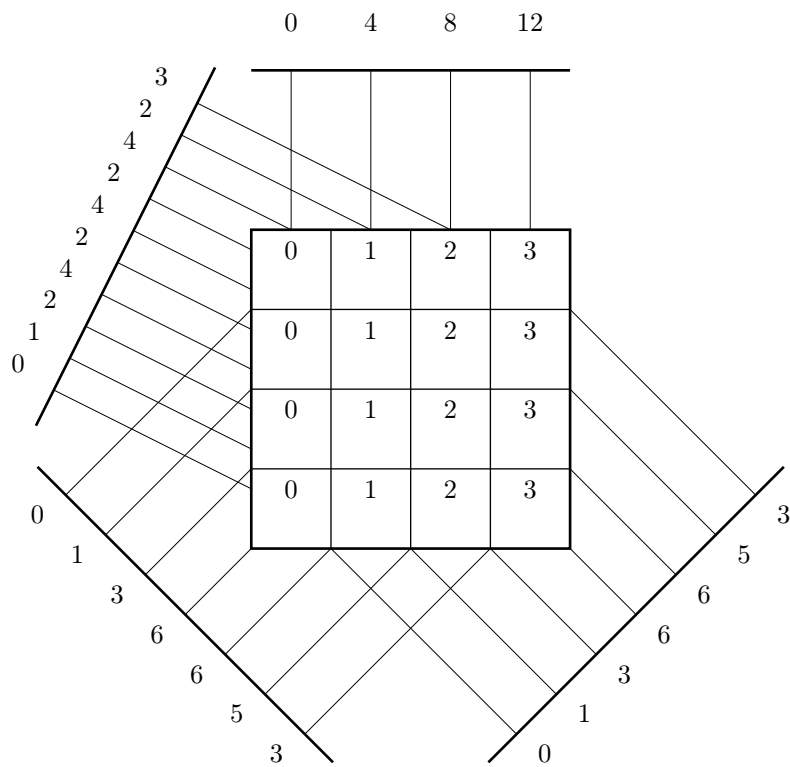


Figure 1 – The Dirac Mojette Transform

**Spline 0 Mojette Transform** In dirac Mojette transform, we can see that the projected bins are only the sum of the pixels that their centers are in the projection lines. The bins only have the information of the center points in the projection line. This is not good for both robustness and performance. But actually the projection goes not only through the centers of the pixels, but also the other parts of the them. Thus, the spline 0 Mojette transform is proposed. It not only consider the center pixels in the projection, but also combine them with the other pixels in the projection line.

The Spline 0 Mojette transform is given by:

$$\mathcal{M}_0 f(k, l) = \mathcal{M}_\delta f(k, l) * kernel_0(k, l, b, p, q), \quad (3)$$

with

$$kernel_0(0, 0, b, p, q) = \begin{cases} \underbrace{(111 \dots 1)}_p * \underbrace{(111 \dots)}_q & \text{if } p \text{ and } q \text{ both odd,} \\ \frac{1}{2} \underbrace{(111 \dots 1)}_p * \underbrace{(111 \dots)}_q * (11) & \text{if either } p \text{ or } q \text{ even} \end{cases} \quad (4)$$

where  $\mathcal{M}_\delta f(k, l)$  refer to (2)

This transform projects each flat pixel and generates trapezoidal shapes onto the projection as depicted in Figure 3

The example of spline 0 Mojette transform is shown in Figure 2

### 2.1.2 Inverse Transform

In this section, we will introduce 2 kinds of inverse transform. They are *Iterative Mojette reconstruction* or traditional Mojette inverse tranform and *Exact filtered backprojection reconstruction*.

**Iterative Mojette reconstruction** This reconstruction uses the transform geometry and the fact that the number of pixels that contribute to each bin is not constant. In some cases, a bin might correspond to a unique pixel in the image, so this bin is then can be reconstructible. The reconstruction solves for one pixel at a time and subtracts this value from the bins that include this pixel in each of the projections.

After this bin update, the set of projections is the exact transform of the pixels that are left to be reconstructed. The reconstruction propagates from the image corners (where there is only one pixel value in the bins) to the center.

In order to achieve it, 2 assistance accounting images are projected with the same set of projection angles and reconstructed simultaneously with the unknown image. The pixels of the first image has the same value 1, referred to as



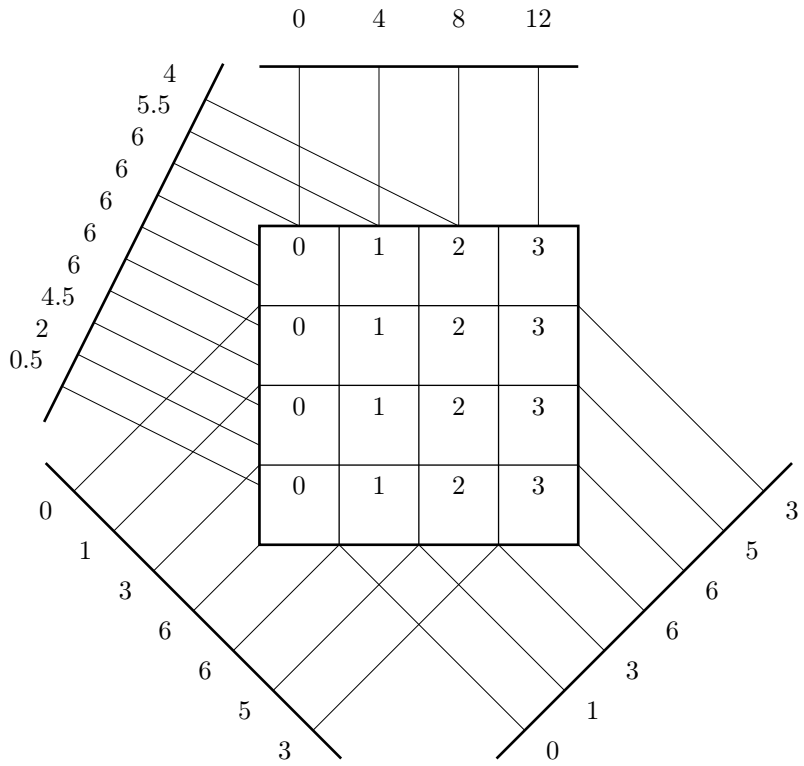


Figure 2 – The Spline 0 Mojette Transform

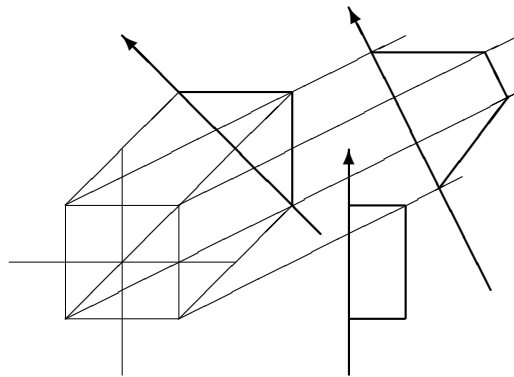


Figure 3 – Projections of a unique pixel with spline 0 Mojette for angles (1,0), (2,1) and (1,1) counter-clockwise

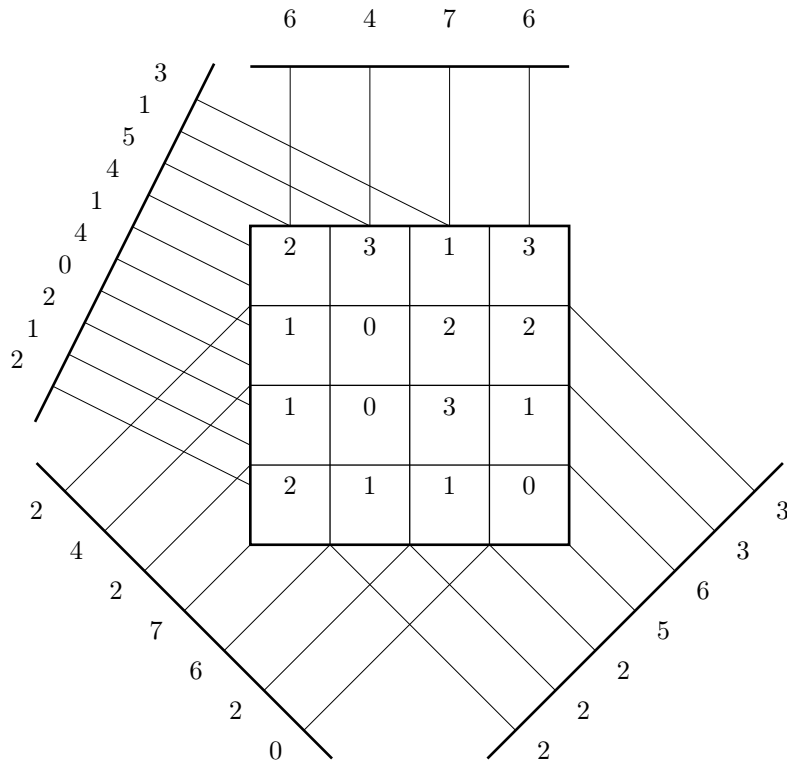
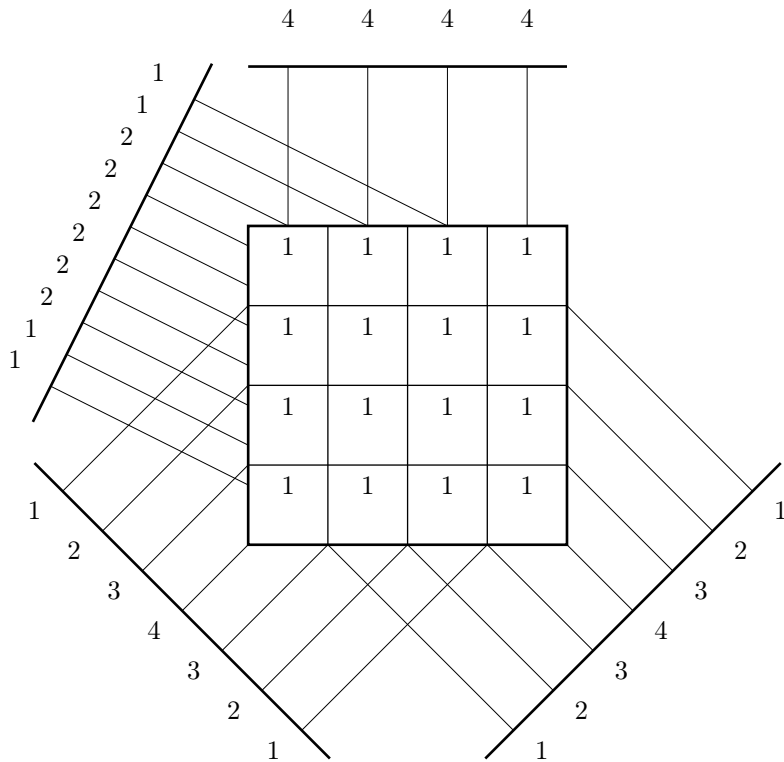


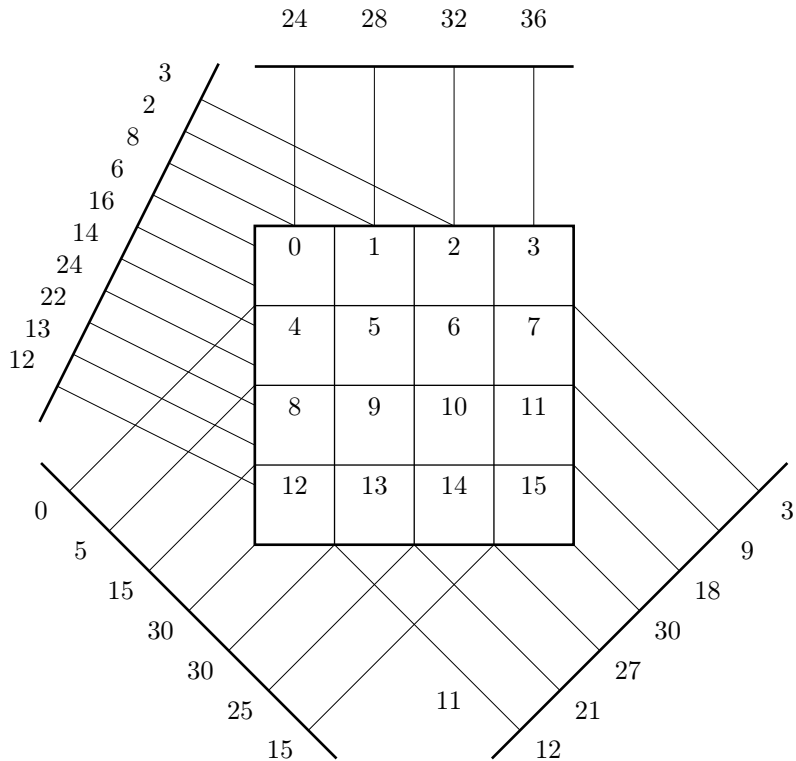
Figure 4 – Mojette transform of the image f

the *unitary image*. The second is an *index image*, whose pixels are labeled in a way that permits a fast access. In Figure 4 and Figure 5, 3 kinds of images are shown.

**Algorithm** The whole algorithm of Iterative Mojette reconstruction is show as follows:



(a) Transform of the *unitary image*  $f_1$



(b) Transform of the *index image*  $f_i$

Figure 5 – The 2 support accounting images

---

**Input:**  $n$ : pixel count,  $I$ : number of projections,  $\mathcal{M}_{p_i, q_i} f, \forall i \in [1 \dots I]$  : projections  
**Output:**  $\{(k, l)$ : pixel value of the reconstructed image for each pixel  $k, l$   
**while**  $\exists b, i$ , a reconstructible bin in  $\mathcal{M}_{p_i, q_i}$  do  
    Find the unique pixel  $(k, l)$  projected onto  $\mathcal{M}_{p_i, q_i} f(b)$ ;  
     $f(k, l) \leftarrow \mathcal{M}_{p_i, q_i} f$ ;  
    Remove the contribution of pixel  $(x, y)$  in all projections  
    **for**  $i \leftarrow 1$  to  $I$  **do**  
         $b \leftarrow q_i \times k - p_i \times l$ ;  
         $\mathcal{M}_{p_i, q_i} f(b) \leftarrow \mathcal{M}_{p_i, q_i} f(b) - f(k, l)$   
    **endfor**  
**endwhile**

---

**Exact filtered backprojection reconstruction** is an exact method which was discovered by Servieres et al.[7]. Given all  $I$  possible projections in the  $P \times Q$  array, backprojection yields  $I - 1$  times the original pixel value plus the sum of the image.

**Algorithm** Here, an exact discrete backprojection filtering algorithm is described for the case of the Dirac pixel model from a global backprojection point of view[6]:

$$[\mathcal{M}_{\delta}^* \text{proj}_{(p, q)}](k, l) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} \sum_{b \in B} \delta(k-i)\delta(l-j) \times \text{proj}(b, p, q) \Delta(b+qi-pj), \quad (5)$$

From the Dirac-Mojette projection (equation (2)) and backprojection operators (equation (5)), we derive the value of the pixel  $(k, l)$  in the image  $(\tilde{f})_i$  obtained by backprojecting the projection with angle  $(p_i, q_i)$ :

$$\tilde{f}_i(k, l) = \sum_{k'} \sum_{l'} f(k', l') \Delta(q_i k - p_i l - q_i k' + p_i l'), \quad (6)$$

This expression can be understood as the sum of pixels aligned with  $(k, l)$  in the direction  $(p_i, q_i)$ . Furthermore, considering a set  $S_I$  of  $I$  projections  $S_I = \{(p_i, q_i), 1 \leq i \leq I\}$ , the backprojected value of pixel  $(k, l)$  becomes

$$\tilde{f}_{S_I}(k, l) = I f(k, l) \sum_{k' \neq k} \sum_{l' \neq l} f(k', l') \Delta(q_i k - p_i l - q_i k' + p_i l'), \quad (7)$$

However, only a finite (although large) number of projections can be considered due to the finite image size. If  $S_I$  contains all the possible discrete

angles on the compact support,  $\tilde{f}_{S_I}(k, l)$  becomes

$$\tilde{f}_{S_I}(k, l) = (I - 1)f(k, l) + \underbrace{\sum_{i=1}^N \sum_{j=1}^N f(i, j)}_S, \quad (8)$$

where  $S$  is the sum of all pixel values, also equal to the sum of each projection bin values. The exact reconstruction scheme is

$$f(k, l) = \frac{1}{I - 1}[\tilde{f}_{S_I}(k, l) - S], \quad (9)$$

### 2.1.3 Farey Series

In the previous paragraphs, we talked about the discrete angles many time. Here let us explain how to get these discrete angles. All the possible discrete angles on a compact space are given by the Farey series corresponding to the size of the reconstruction square. A Farey series of order  $N$  is the set of all ratios in lowest terms between 0 and 1 whose denominators do not exceed  $N$ , arranged in order of magnitude. For instance,  $F_4$  is the series  $\{\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1}\}$ . With such a series, we have a set of discrete angles in the  $\frac{q}{p}$  form arranged in order of magnitude between  $[0, \frac{\pi}{2}]$ . Angles over  $[0, \pi]$  are obtained by symmetry. To obtain all the discrete directions in an  $N \times N$  image, a Farey series of order  $(N - 1)$  is used with its symmetric values towards the first bisector and the  $y$ -axis. The ratio's denominator gives the  $p$  direction and the numerator gives the  $q$  direction. The Figure 6 shows the angle in both  $0 - \frac{\pi}{4}$  and  $0 - \pi$ . The rest angles can be got from symmetry.

## 2.2 Lattices

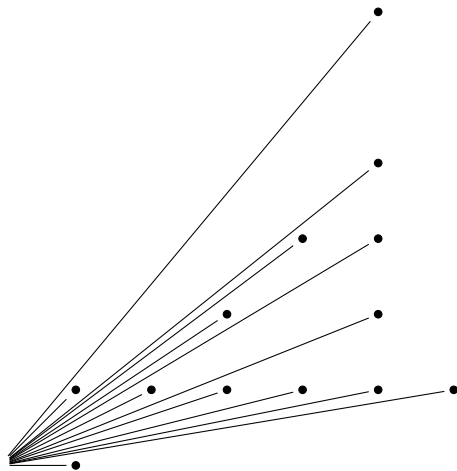
### 2.2.1 Definition

First of all, let us give a definition of the lattice. Lattice is a regular repeating arrangement of points in a certain dimension space. In crystallography these lattices are usually called *Bravais lattices*[5][1][8][2]. In three dimensions, for example, we can find three centers  $\nu_1, \nu_2, \nu_3$ , or generally  $n$  centers  $\nu_1, \nu_2, \dots, \nu_n$  for an  $n$ -dimensional lattice. The set of all these centers consists of the sums  $\sum k_i \nu_i$  where the  $k_i$  are integers.

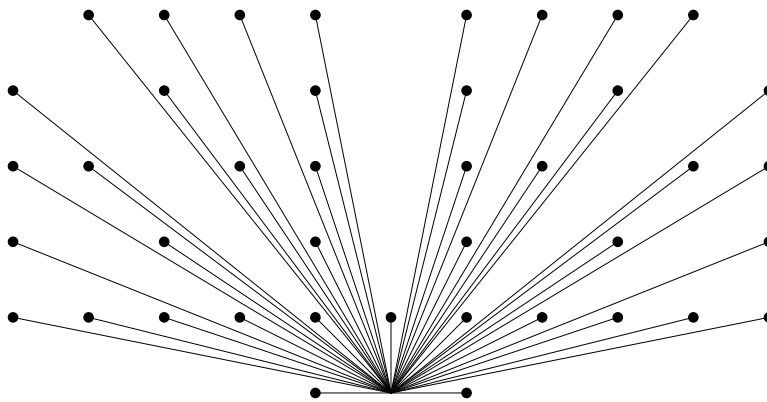
The vectors  $\nu_1, \dots, \nu_n$  are then called a *basis* for the lattice, The paralleloptope consisting of the points

$$\theta_1 \nu_1 + \dots + \theta_n \nu_n \quad (0 \leq \theta_i < 1)$$

is a *fundamental paralleloptope*. Figure 7 shows a two-dimensional lattice and the fundamental paralleloptope determined by the basis  $\nu_1, \nu_2$ . A fundamental paralleloptope is an example of a *fundamental region* for the lattice, that is, the structure in the whole space will be built just by repeating many times of one



(a) The Farey Series,  $0 - \frac{\pi}{4}$



(b) The Farey Series,  $0 - \pi$

Figure 6 – The Farey Series

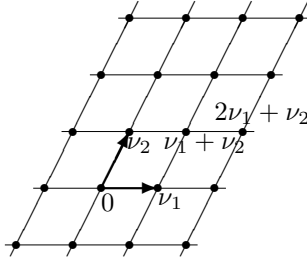


Figure 7 – The plane divided into fundamental parallelotopes of a 2-dimension lattice.

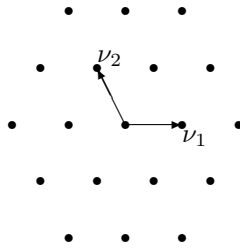


Figure 8 – The hexagonal lattice.

lattice point in each copy. Figure 8 shows a hexagonal fundamental region for a two-dimensional lattice.

There are many different ways of choosing a basis and a fundamental region for a lattice  $\Lambda$ . But *the volume of the fundamental region* is uniquely determined by  $\Lambda$ , and the square of this volume is called the *determinant* or *discriminant* of the lattice. The simple formula for the determinant is as follows:

Let the coordinates of the basis vectors be:

$$\nu_1 = (\nu_{11}, \nu_{12}, \dots, \nu_{1m}),$$

$$\nu_2 = (\nu_{21}, \nu_{22}, \dots, \nu_{2m}),$$

...

$$\nu_n = (\nu_{n1}, \nu_{n2}, \dots, \nu_{nm}),$$

where  $m \geq n$  (sometimes it is convenient to use  $m > n$  coordinates to describe an  $n$ -dimensional lattice). The matrix

$$M = \begin{pmatrix} \nu_{11} & \nu_{12} & \dots & \nu_{1m} \\ \nu_{21} & \nu_{22} & \dots & \nu_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{n1} & \nu_{n2} & \dots & \nu_{nm} \end{pmatrix}, \quad (10)$$

is called a *generator matrix* for the lattice, and the lattice vectors consist of all the vectors:

$$\xi \mathbf{M}, \quad (11)$$

where  $\xi = (\xi_1, \dots, \xi_n)$  is an arbitrary vector with integer components  $\xi_i$ . The matrix

$$\mathbf{A} = \mathbf{M} \mathbf{M}^{tr}, \quad (12)$$

where  $tr$  denote transpose, is called a *Gram matrix* for the lattice. The  $(i, j)$ th entry of  $\mathbf{A}$  is the inner product  $\nu_i \cdot \nu_j$ . The determinant of  $\Lambda$  is then the determinant of the matrix  $\mathbf{A}$ ,

$$\det \Lambda = \det \mathbf{A}, \quad (13)$$

if  $\mathbf{M}$  is a square matrix this formula can be written as:

$$\det \Lambda = (\det \mathbf{M})^2. \quad (14)$$

Since we have the basic ideas about the lattice, now we can go to its applications or the problems that the lattices can solve. In the next section, we will address not only the problems that involved the lattice but also the key parameters that characterize the lattice.

### 2.2.2 Examples of problems involving lattices

#### Packing problem

The classical sphere problem is to find out how densely a large number of identical spheres can be packed together. And what the lattice packing is whether it has the properties that 0 is a center and that if there are spheres with centers  $\mu$  and  $\nu$  then there are also spheres with centers  $\mu + \nu$  and  $\mu - \nu$ . In other words the set of centers forms an additive group. The Figure 9 shows the different packing ways due to the different lattices.

Since we represent the mathematical form of lattice in the previous section, now we can give a precise definition of the density  $\Delta$  of a lattice packing:

$$\begin{aligned} \Delta &= \text{percentage of the space that is occupied by the spheres} \quad (15) \\ &= \frac{\text{volume of one sphere}}{\text{volume of fundamental region}} \\ &= \frac{\text{volume of one sphere}}{(\det \Lambda)^{\frac{1}{2}}} \quad (16) \end{aligned}$$

And here there is a very important parameter— $\rho$ , which is the radius of the inscribed sphere. From the equation (16), we can see that *The volume of one sphere* is the numerator of the equation. it is constrained by the value of  $\rho$ . So finally,  $\rho$  will effect the result of the density  $\Delta$ .



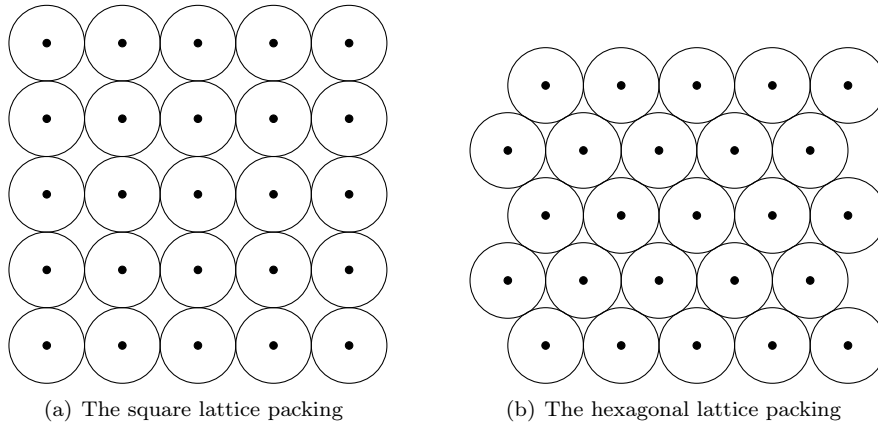


Figure 9 – The packing problems of different lattices

### Kissing number problem

Another problem which is closely related to the packing problem is that *kissing number* problem. An example for this problem is that how many billiard balls can be arranged so that they all just touch, or "kiss", another billiard ball of the same size in three dimensions.

More generally we may define the *kissing number* (usually denoted by  $\tau$ ) of a sphere packing in any dimension to be the number of spheres that touch one sphere. For a certain lattice packing  $\tau$  is the same for every sphere, but for an arbitrary packing  $\tau$  may vary from one sphere to another. The  $n$ -dimensional version of the kissing number problem asks for the greatest value of  $\tau$  attained by any packing of  $n$ -dimensional spheres. We now have the answers to the first 3 kinds of dimensions. In one-dimension the answer is 2. And the answers are 6 and 12 in two dimensions and three dimensions respectively.

Let us consider the kissing number problem in another way: how many points can be placed on the surface of a sphere in  $n$ -dimensional space  $\mathbb{R}^n$  so that the angular separation between any two points is at least  $60^\circ$  shown as Figure 10.

Thus the kissing number problem can be treated as a packing problem. Rather than in  $\mathbb{R}^n$  itself, the packing points are on the surface of a sphere in  $\mathbb{R}^n$ . So we can write down the following formula:

$$\Omega_n = \{(x_1, \dots, x_n) \in \mathbb{R}^n : \sum x_i^2 = 1\}, \quad (17)$$

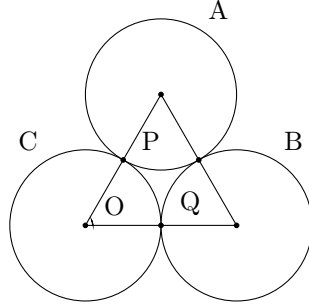


Figure 10 – The kissing number problem. If spheres  $A$  and  $B$  touch sphere  $C$ , then the angular separation between the contact points  $P$  and  $Q$ , the angle  $POQ$ , is at least  $60^\circ$ . For if  $A$  touches  $B$ , the three centers form an equilateral triangle.

We call a finite subset  $\mathbf{X}$  of  $\Omega_n$  a *spherical code*, and say that  $\mathbf{X}$  has *minimal angle*  $\phi$  if  $\phi$  is the largest number for which

$$x \cdot y \leq \cos\phi \quad \text{for any } x, y \in \mathbf{X}, \quad x \neq y. \quad (18)$$

We now generalize the kissing number problem by get , for given  $n$  and  $\phi$ , the maximal number  $A(n, \phi)$  of points in a spherical code in  $\Omega_n$  of minimal angle  $\phi$ . For example,  $A(n, \pi/3)$  is the kissing number problem, and we have seen that  $A(2, \pi/3) = 6$ ,  $A(3, \pi/3) = 12$ ,  $A(4, \pi/3) = 24$  or 25, etc.

### Covering problem

The third problem is a kind of dual to the packing problem, and asks for the most economical way to *cover*  $n$ -dimensional Euclidean space with equal overlapping spheres. Figure 11 shows two different ways to cover the plane with overlapping circles. In (a) the centers of the circles belong to the square lattice  $\mathbf{Z}^2$ , and in (b) to the hexagonal lattice. It is clear that (b) is a more efficient covering than (a), since there is less overlap among the circles.

To make this precise we define *the thickness*  $\Theta$  of a covering in the same way as the density  $\Delta$  of a packing. Suppose an arrangement of spheres of radius  $\mathbf{R}$  covers  $\mathbf{R}^n$ . If the centers form a lattice  $\Lambda$  then the *thickness* is defined by following formular:

$$\begin{aligned} \Theta &= \frac{\text{average numbers of spheres that contain a point of the space}}{\text{volume of one sphere}} \\ &= \frac{V_n R^n}{(\det \Lambda)^{1/2}} = \frac{V_n R^n}{(\det \Lambda)^{1/2}}, \end{aligned} \quad (19)$$

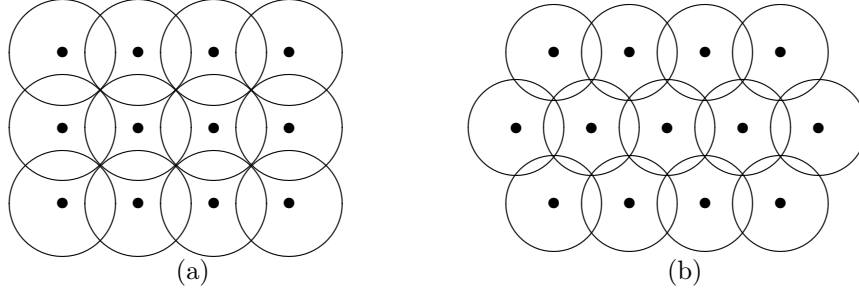


Figure 11 – Covering the plane with circles. In (a) the centers belong to the square lattice  $\mathbf{Z}^2$ , in (b) they belong to the hexagonal lattice. (b) is a more efficient or *thinner* covering.

$\Theta$  is also called the *density* or the *sparsity* of the covering. The thickness of an arbitrary covering is defined in the same way as the density of an arbitrary packing. Notice that we always have  $\Delta \leq 1 \leq \Theta$ .

Sometimes we use *normalized thickness* to reduce the influence of different volumes of spheres. The *normalized thickness* (or *center density*)  $\theta$  is given by

$$\theta = \frac{\Theta}{V_n} \quad (20)$$

The lattice coverings in Figure 11 have thickness  $\Theta = \pi/2 = 1.5708\dots$  and  $\Theta = 2\pi/3\sqrt{3} = 1.2092\dots$  respectively.

Then the *covering problem* asks for the thinnest covering of  $n$ -dimensional space by spheres, i.e. for the covering with minimal thickness.

### Conclusion of the section

In this section we have discussed 3 kinds of lattice problems– *packing*, *kissing number* and *covering problem*. Here we do not give all the possible problems of lattice (the other problems such as quantization problem, shortest vector problem and so on). And also we know several parameters that can characterize a lattice. As a conclusion, we show these parameters as follows:

*minimal vectors*: They are the minimal vectors to form the basis lattice. Details and illustrations can be seen in Figure 7. They also form the *generator matrix*, which is the mathematical representation of the lattice.

Table 1 – The categories of the used lattices

Cubic Lattice	$Z^n$ ( $n \leq 3$ )
	$A_3$ or $D_3$ $A_3^*$ or $D_3^*$
Diamond Lattice	$D_2$
Hexagonal Lattice	$A_2$

$\det \Lambda$ : It determine the volume of the fundamental region. According to equation (16) and equation (19),  $\det \Lambda$  can be used not only to solve the packing problem, but also to solve the covering problem. It can be got from the *Gram Matrix*.

$\Delta$ : The density of the packing. The direct representation of the packing ability of a lattice.

$\rho$ : The radius of the inscribed sphere, which effects the solution to the packing problem. It is the key parameter to represent the packing problem according to (16).

$\tau$ : It means a sphere packing in any dimension to be the number of spheres that touch one sphere.

$R$ : The radius of the circumscribed sphere, which effects the solution to the covering problem according to (19). A good packing is to maximize  $\rho$  and minimize  $R$ [3].

### 2.2.3 Several important lattices and their illustrations

Since we know the problems and the key parameters of lattices, now we are going to introduce the different kinds of lattices and the values of their key parameters. There are several high space dimensional lattices mentioned in the following content, but due to the domain of the topic, our space dimensions are limited to the dimension  $\leq 3$ .

The categories of the lattices that we are going to use are shown in TABLE 1. The meaning of symbols  $Z_n$ ,  $A_n$ ,  $A_n^*$  and  $D_n$  will be described in the following paragraphs.

**The  $n$ -dimensional cubic lattice  $Z^n$**  The set of intergers,  $\dots, -2, -1, 0, 1, 2, 3, \dots$  is denoted by  $\mathbf{Z}$ , and

$$\mathbf{Z}^n = \{(x_1, x_2, \dots, x_n) : x_i \in \mathbf{Z}\}$$

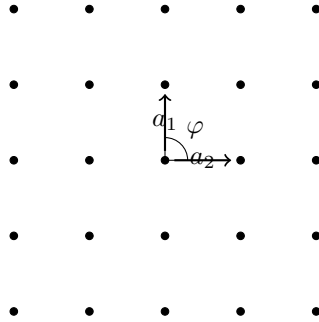


Figure 12 – The Square Lattice.  $|a_1|=|a_2|^2$ ,  $\varphi=90^\circ$ . The square lattice is a  $\mathbb{Z}^2$  lattice.

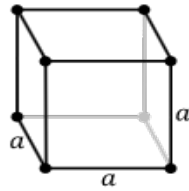


Figure 13 – The Cubic Lattice.  $|a|$  is all the same. The cubic lattice is a  $\mathbb{Z}^3$  lattice.

is the  $n$ -dimensional cubic or integer lattice. For example, the  $\mathbb{Z}^2$  lattice is the square lattice. Its generator matrix is the identity matrix.

Key attributes:

Attribute	Value
$\det \Lambda$	1.
$\tau$	$\tau = 2n$ .
minimal vectors	$(0, \dots, \pm 1, \dots, 0)$ .
$\rho$	$1/2$ .
$R$	$\sqrt{n}/2$ .
$\Delta$	$V_n 2^{-n}$ .

**Illustrations** Here we take  $\mathbb{Z}^2$  and  $\mathbb{Z}^3$  as an example. The  $\mathbb{Z}^2$  lattice is a square lattice shown as Figure 12. And the  $\mathbb{Z}^3$  lattice is a cubic lattice shown as Figure 13

<sup>2</sup>Here,  $a_i \equiv v_i$ ,  $v_i$  is the minimal vectors that we have shown in the previous section

**The  $n$ -dimensional lattice  $A_n$**  The definition of lattice  $A_n$  is: For  $n \geq 1$ ,

$$A_n = \{(x_0, x_1, \dots, x_n) \in \mathbf{Z}^{n+1} : x_0 + x_1 + \dots + x_n = 0\}$$

which uses  $n + 1$  coordinates to define an  $n$ -dimensional lattice:  $A_n$  lies in the hyperplane  $\sum x_i = 0$  in  $\mathbf{R}^{n+1}$ .

And its generator matrix is:

$$M = \begin{pmatrix} -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix}. \quad (21)$$

And its two possible Gram matrix are:

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 \\ 0 & -1 & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & 0 & \dots & -1 & 2 \end{pmatrix}, \quad (22)$$

or

$$A = \begin{pmatrix} 2 & 1 & 1 & \dots & 1 & 1 \\ 1 & 2 & 1 & \dots & 1 & 1 \\ 1 & 1 & 2 & \dots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 2 & 1 \\ 1 & 1 & 1 & \dots & 1 & 2 \end{pmatrix}. \quad (23)$$

Key attributes<sup>3</sup>:

Attribute	Value
$\det \Lambda$	$n+1$ .
$\tau$	$\tau = n(n+1)$ .
minimal vectors	$(1, -1, 0, \dots, 0)$ .
$\rho$	$1/\sqrt{2}$ .
$R$	$\rho\{\frac{2a(n+1-a)}{n+1}\}^{\frac{1}{2}}$ .
$\Delta$	unknown.

**Illustrations** For  $A_n$  lattices, there are two special cases in our topic. That is – *hexagonal lattice* in dimension 2 and *face-centered cubic* in dimension 3. These two kinds of lattices will be introduced in the following paragraphs.

<sup>3</sup> The value 'a' in R is the integer part of  $(n+1)/2$

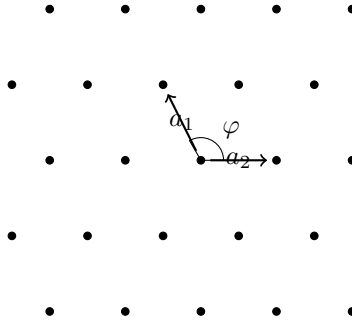


Figure 14 – The Hexagonal Lattice.  $|a_1|=|a_2|$ ,  $\varphi=120^\circ$ . The hexagonal lattice is a  $A_2$  lattice.

**The hexagonal lattice**  $A_2^4$  is the hexagonal lattice as shown in Figure 8. The hexagonal lattice may be spanned by the vectors  $(1, 0)$  and  $(-\frac{1}{2}, \frac{\sqrt{3}}{2})$ , so the generator which is an alternative to (21) is:

$$M = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}. \tag{24}$$

Key attributes:

Attribute	Value
$\det \Lambda$	$\frac{3}{4}$ .
$\tau$	$\tau = 6$ .
minimal vectors	$(\pm 1, 0)$ and $(\pm \frac{1}{2}, \pm \frac{\sqrt{3}}{2})$ .
$\rho$	$\frac{1}{2}$ .
$R$	$\frac{2\rho}{\sqrt{3}}$ .
$\Delta$	$\frac{\pi}{\sqrt{12}} = 0.9069\dots$

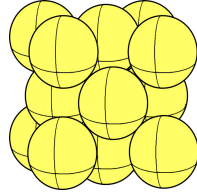
**Illustrations** The  $A_2$  lattice is a hexagonal lattice shown as Figure 14.

**The face-centered cubic lattice**  $A_3$  is a special case of  $A_n$ . Because  $A_3$  and  $D_3$  are equivalent, and called the *face-centered cubic lattice* (or fcc). The simplest definition of  $D_3$  is: the fcc consists of the points  $(x, y, z)$ , where  $x, y$  and  $z$  are integers with an even sum. The generator matrix is:

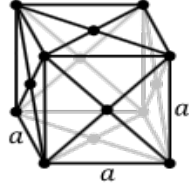
$$M = \begin{pmatrix} -1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}. \tag{25}$$

Key attributes:

<sup>4</sup>Here is the  $A_2$  lattice, for  $A_1$ , there is the equation  $A_1 \cong \mathbf{Z}^1$ .



(a) The common example of fcc



(b) The abstract show of fcc

Figure 15 – The Face-centered Lattice.  $|a_i|$  is all the same. The face-centered cubic lattice is not only a  $\mathbf{A}_3$  lattice, but also a  $\mathbf{D}_3$  lattice.

Attribute	Value
$\det \Lambda$	4.
$\tau$	$\tau = 12.$
minimal vectors	all permutations of $(\pm 1, \pm 1, 0).$
$\rho$	$\frac{1}{\sqrt{2}}.$
$R$	$\rho\sqrt{2} = 1.$
$\Delta$	$\frac{\pi}{\sqrt{18}} = 0.7405\dots$

**Illustrations** The  $A_3$  lattice is a face-centered lattice shown as Figure 15.

**The dual lattice  $\mathbf{A}_n^*$**  The lattice dual to  $\mathbf{A}_n$  is

$$\mathbf{A}_n^* = \bigcup_{i=0}^n ([i] + \mathbf{A}_n), \quad (26)$$

with generator matrix:

$$M = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & -1 & 0 \\ \frac{-n}{n+1} & \frac{1}{n+1} & \frac{1}{n+1} & \dots & \frac{1}{n+1} & \frac{1}{n+1} \end{pmatrix}. \quad (27)$$



An equivalent definition uses either the Gram matrix:

$$\mathbf{A} = \begin{pmatrix} n & -1 & -1 & \dots & -1 \\ -1 & n & -1 & \dots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \dots & n \end{pmatrix}. \quad (28)$$

Key attributes:

Attribute	Value
$\det \Lambda$	$\frac{1}{n+1}$ .
$\tau$	2 (n=1) or 2n+2 (n ≥ 2).
minimal vectors	unknown.
$\rho$	$\frac{1}{2} \sqrt{\frac{n}{n+1}}$ .
$R$	$\rho \sqrt{\frac{n+2}{3}} = \sqrt{\frac{n(n+2)}{12(n+1)}}$ .
$\delta$	$\frac{\frac{n}{2}}{2^n (n+1)^{\frac{n-1}{2}}}$ .

**Illustrations** For  $A_n^*$  lattices, there is one special case in our topic. That is – *body-centered lattice*. This kind of lattice will be introduced in the following paragraph.

**The body-centered cubic lattice** As a special case of  $A_n^*$ ,  $A_3^*$  and  $D_3^*$  are equivalent to the *body-centered cubic lattice* ( or *bbc*). The definition is that : the bcc consists of the points  $(x, y, z)$  where  $x, y$  and  $z$  are all even or all odd integers. A generator matrix is:

$$\mathbf{M} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \quad (29)$$

Key attributes:

Attribute	Value
$\det \Lambda$	16.
$\tau$	8.
minimal vectors	$(\pm 1, \pm 1, \pm 1)$ .
$\rho$	$\frac{\sqrt{3}}{2}$ .
$R$	$\rho \sqrt{\frac{5}{3}} = \frac{\sqrt{5}}{2}$ .
$\Delta$	$\frac{\pi\sqrt{3}}{8} = 0.6802\dots$

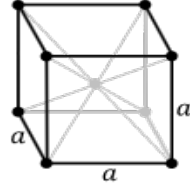


Figure 16 – The Body-centered Lattice.  $|a|$  is all the same. The body-centered cubic lattice is not only a  $A_3^*$  lattice, but also a  $D_3^*$  lattice.

**Illustrations** The  $A_3^*$  lattice is a body-centered cubic lattice shown as Figure 16.

**The  $n$ -dimensional lattices  $D_n$**  The definition of  $D_n$  is: For  $n \geq 3$ ,

$$D_n = \{(x_1, \dots, x_n) \in \mathbf{Z}^n : x_1 + \dots + x_n \text{ even}\},$$

or in other words  $D_n$  is obtained by coloring the points of  $\mathbf{Z}^n$  alternately red and white with a checkerboard coloring, and taking the red points. So it is also called *checkerboard lattice*. Its generator matrix is

$$M = \begin{pmatrix} -1 & -1 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix} \quad (30)$$

Key attributes

Attribute	Value
$\det \Lambda$	4.
$\tau$	$2n(n-1)$ .
minimal vectors	all permutations of $(\pm 1, \pm 1, 0, \dots, 0)$ .
$\rho$	$\frac{1}{\sqrt{2}}$ .
$R$	$\rho\sqrt{2}(n=3)$ or $\rho\sqrt{\frac{n}{2}}$ .
$\delta$	$2^{\frac{-(n+2)}{2}}$ .

**Illustrations** For  $D_n$  lattices, there are two special cases in our topic. That is – *diamond lattice* in dimension 2 and *face-centered cubic* in dimension 3. The diamond lattice is shown as Figure 17, and the face-centered cubic is shown as Figure 15.

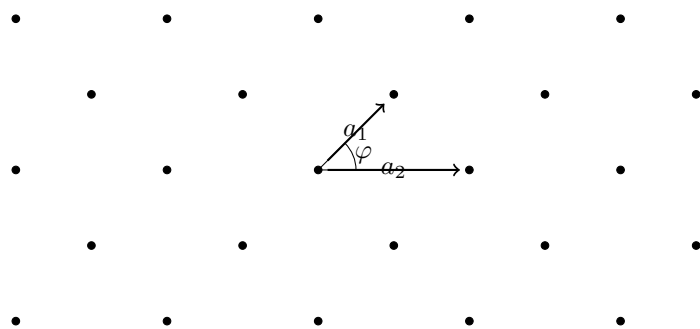


Figure 17 – The Diamond Lattice.  $|a_1| \neq |a_2|$ ,  $\varphi=45^\circ$ . The diamond lattice is a  $D_2$  lattice.

## 3 Works and Algorithms

### 3.1 Farey Series in Hexagonal Lattice

In bibliography, we have introduced the ordinary method of generating the Farey Series. Since we have changed the lattice from square lattice to hexagonal lattice, we also need to change the way of generating Farey Series. In general, the Farey Series can be produced by the normal way but with the transform from square lattice to hexagonal lattice. Let us start from the algorithm of generating Farey Series.

Assume that  $\frac{p}{q}$  are the neighbor of  $\frac{a}{b}$  and  $\frac{c}{d}$  in the Farey Series  $F_N$ . And from the definition of Farey Series, we can get the equation:  $\frac{p}{q} = \frac{a+c}{b+d}$ . We can see that  $\frac{p}{q}$  is in lowest term. So there is an integer  $k$  that makes  $kp = a + c$  and  $kq = b + d$ . Then we can get the value of  $c$  and  $d$  from  $c = kp - a$  and  $d = kq - b$  respectively. Now we should get the value of  $k$ . In fact the value of  $k$  must give a value of  $\frac{c}{d}$  which is as much close as possible to  $\frac{p}{q}$ . That means  $k$  must be as large as possible subject to  $kq - b \leq N$ . So  $k$  is the greatest integer which is no greater than  $\frac{N+b}{q}$ . Then we get the equation (31). If we have given the values of  $p, q, a, b$  in this subpart of Farey Series  $\{\frac{a}{b}, \frac{p}{q}, \frac{c}{d}\}$ , we can get the next sequence  $\frac{c}{d}$ .

$$\begin{aligned} c &= \lfloor \frac{N+b}{q} \rfloor p - a, \\ d &= \lfloor \frac{N+b}{q} \rfloor q - b \end{aligned} \quad (31)$$

So the whole algorithm of generating Farey Series is as show in Table 2.

The previous method can generate an ordinary Farey Series. But here what we need is to generate an hexagon based Farey Series. The key is the transformation of the coordinate using *Transform matrix* (32), where  $p'$  and  $q'$  mean the Farey Series in square lattice,  $p$  and  $q$  mean the Farey Series in hexagonal lattice.

$$\begin{pmatrix} p \\ q \end{pmatrix} = \mathbf{H} \times \begin{pmatrix} p' \\ q' \end{pmatrix}, \text{ where } \mathbf{H} = \begin{pmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix}. \quad (32)$$

From the algorithm that we gave in Table 2 and the equation (32), we can get the Farey Series in hexagonal lattice, which can be seen in Fig 18.

The previously generated Farey Series is more like a parallelogram than an hexagon. What will change if we insist to form an strict hexagonal Farey Series? The idea is the same but the area should be also be modified. As shown in Fig 19, The whole area can be divide to the triangle area  $B$  and the square area  $A$  and their symmetries.

Table 2 – Algorithm of generating Farey Series

---

**Input:**  $N$ :the order of the Farey Series  
**Output:**  $F_n$  Farey Series

$a \leftarrow 0$ ;  
 $b \leftarrow 1$ ;  
 $p \leftarrow 1$ ;  
 $q \leftarrow N$ ;  
**while**  $p \leq N$  **do**  
     $F_n \leftarrow (a, b, p, q)$   
     $k \leftarrow \lfloor \frac{N+b}{q} \rfloor$   
     $(a, b) \leftarrow (p, q)$   
     $(p, q) \leftarrow (kp - a, kq - b)$   
**endwhile**

---

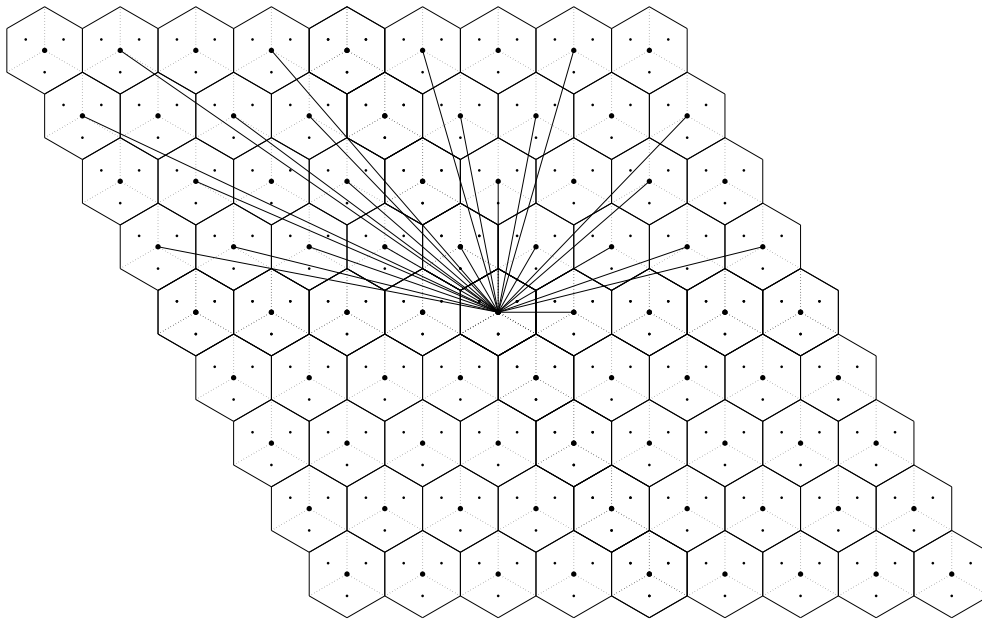


Figure 18 – The Farey Series  $F_4$  in hexagonal lattice

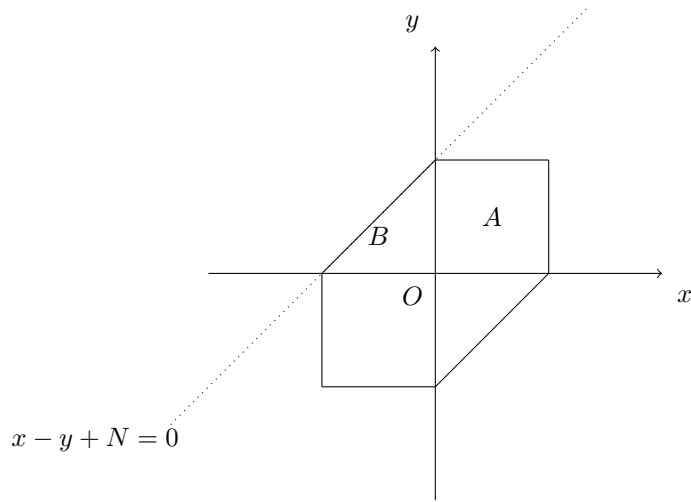


Figure 19 – The illustration of area which form the hexagon after transform

It is not enough for us that just modified the area. The equation of the value of  $k$  is also need to be changed. The value of  $k$  should be less or equal to the line  $x - y + N = 0$ . So the value of  $k$  should be  $\lfloor \frac{a-b-N}{p-q} \rfloor$ . So the algorithm should also be changed to Table 3.

If we have finished all previously mentioned modification. The result should be the same as shown in Fig 20 but with different order  $n$ .

### 3.2 Dirac Mojette Transform in Hexagonal Lattice

Since we have get the Farey Series in hexagonal lattice, we can use the knowledge that mentioned in Bibliography part to do the Dirac Mojette transform[4]. It is the same idea, but the points and directions are all in hexagon lattice. And the example is shown as Fig 21. Here we do not address too much. The details are written in the reference book. And the transformations of directions can be seen in previous part as well as the transformations of points are in the following chapter.

### 3.3 Spline0 Mojette Transform in Hexagonal Lattice

#### 3.3.1 Transformation of the Points

From square lattice to hexagonal lattice, what we first need to do is change the coordinates of points. A point in a plane has different coordinates according to different lattices we used. Because all the computation that we use is based

Table 3 – Algorithm of generating Farey Series (exactly hexagon)

---

**Input:**  $N$ :the order of the Farey Series  
**Output:**  $F_n$  Farey Series  
 $a \leftarrow 0$ ;  
 $b \leftarrow 1$ ;  
 $p \leftarrow 1$ ;  
 $q \leftarrow N - 1$ ;  
**while**  $q - p \leq N$  **do**  
     $F_n \leftarrow (a, b, p, q)$   
     $k \leftarrow \lfloor \frac{a-b-N}{p-q} \rfloor$   
     $(a, b) \leftarrow (p, q)$   
     $(p, q) \leftarrow (kp - a, kq - b)$   
**endwhile**

---

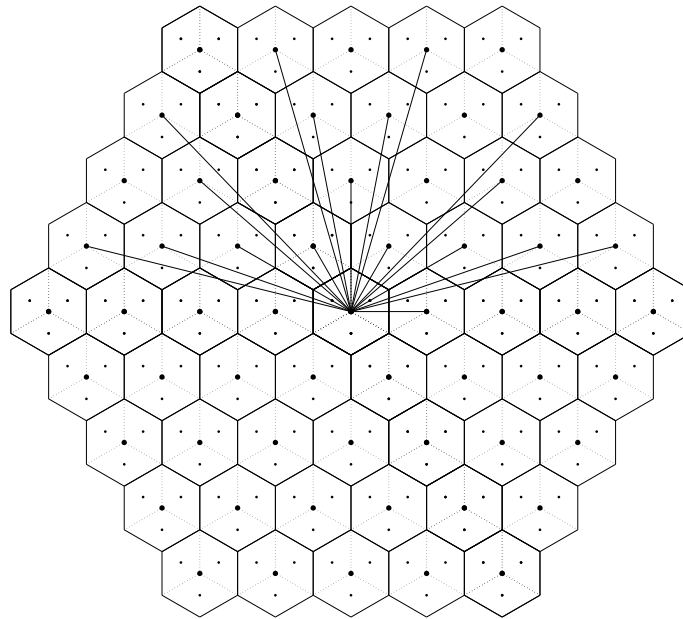


Figure 20 – The Farey Series  $F_4$  in hexagonal lattice (exactly hexagon)

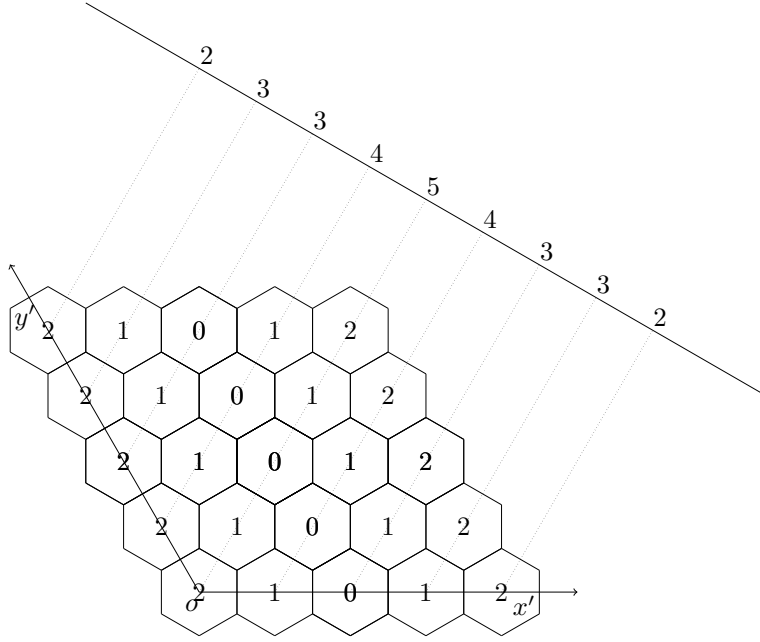


Figure 21 – The illustration of Dirac Mojette transform in hexagonal lattice

on square lattice while the expressions of parameters are in hexagonal lattice, we need a way to transfer the values between them.

Here we use the same idea as we have used in previous sections—the transformation matrix. The equation 33 is used for transform the hexagonal coordinate to the square coordinate. And its reverse is equation 34 which used for transform from square coordinate back to hexagonal coordinate. Note that  $\begin{pmatrix} x \\ y \end{pmatrix}$  means the coordinate in square lattice as well as  $\begin{pmatrix} x' \\ y' \end{pmatrix}$  in hexagonal lattice. And in following sections we will continue using this representation.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{A} \times \begin{pmatrix} x' \\ y' \end{pmatrix}, \text{ where } \mathbf{A} = \begin{pmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix}. \quad (33)$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \mathbf{A}^{-1} \times \begin{pmatrix} x \\ y \end{pmatrix}, \text{ where } \mathbf{A}^{-1} = \begin{pmatrix} 1 & -\frac{\sqrt{3}}{3} \\ 0 & \frac{2\sqrt{3}}{3} \end{pmatrix}. \quad (34)$$

### 3.3.2 Method of Generating Kernel

As we have expressed in the Bibliography part, Spline 0 Mojette transform need a kernel to convolution. In square lattice we can use the the projection direction  $(p, q)$  to generate the convolution kernel. But we can not do it in



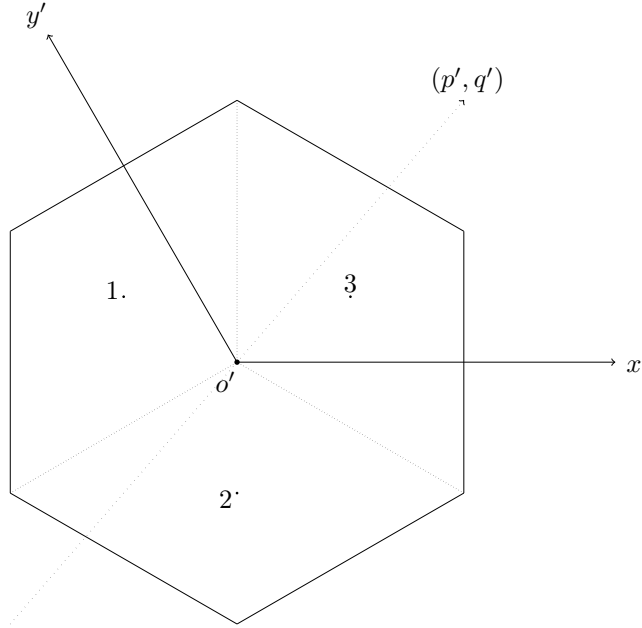


Figure 22 – The decomposition of an hexagon

the same way in hexagonal lattice cause the distance of Voronoi cell center has changed.

**Decomposition of Hexagons and Their Own Direction** Here we use the geometry method to compute the kernel. The elements of kernel is the length in hexagon. So the goal is to get the length in an hexagon. But it is quite complex to compute the length in an hexagon with a given direction  $(p, q)$ . So we decide to decompose an hexagon into 3 parts– the same 3 parallelograms but with different orientations shown as in Fig 22. So we can compute the length in parallelogram label 1, 2 and 3 respectively. And finally we sum these lengths up to get the total length in hexagon.

But the question is how we can get the different lengths in different parallelogram using just the parameter of direction  $(p', q')$ ? Notice that for parallelogram 2, it is the rotation of  $\frac{2\pi}{3}$  at the center  $o'$  of parallelogram 1. It is the same for parallelogram 3, which is also the rotation of parallelogram 2. So we can use the idea of coordinate rotation to compute the length in any direction while considering only one parallelogram case.

Let start with the second parallelogram, which is the rotation of  $\frac{2\pi}{3}$  at the center  $o'$  of the first parallelogram. From equation (35), we can see that the

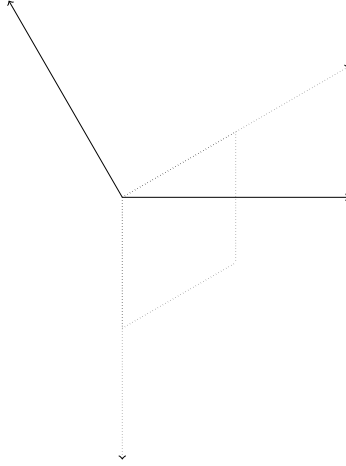


Figure 23 – The rotation of the coordinate system in hexagonal grid

projection direction  $\begin{pmatrix} p' \\ q' \end{pmatrix}$  should right multiply matrix  $T_2$ . That means first transfer from hexagonal lattice to square lattice and then rotate  $\frac{2\pi}{3}$  counterclockwise and finally transfer from square lattice to hexagonal lattice. So we can just use  $T_2$  right multiply with direction  $\begin{pmatrix} p'_1 \\ q'_1 \end{pmatrix}$ , which is the right direction for the first parallelogram, to generate the correct direction  $\begin{pmatrix} p'_2 \\ q'_2 \end{pmatrix}$  for parallelogram 2.

For parallelogram 3, we can repeat the previous operation but use  $\begin{pmatrix} p'_2 \\ q'_2 \end{pmatrix}$  instead of  $\begin{pmatrix} p'_1 \\ q'_1 \end{pmatrix}$ . Then we can get the right direction  $\begin{pmatrix} p'_3 \\ q'_3 \end{pmatrix}$  for parallelogram 3.

$$\begin{aligned}
 \mathbf{T}_2 &= \begin{pmatrix} 1 & \frac{\sqrt{3}}{3} \\ 0 & \frac{2\sqrt{3}}{3} \end{pmatrix} \times \begin{pmatrix} \cos \frac{2\pi}{3} & -\sin \frac{2\pi}{3} \\ \sin \frac{2\pi}{3} & \cos \frac{2\pi}{3} \end{pmatrix} \times \begin{pmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \\
 &= \begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix} \tag{35}
 \end{aligned}$$

So the question left now is how to get the initial value of  $\begin{pmatrix} p'_1 \\ q'_1 \end{pmatrix}$  from the given  $\begin{pmatrix} p' \\ q' \end{pmatrix}$ . The trick is the same. But the rotation angle changed from  $\frac{2\pi}{3}$  to  $\frac{\pi}{2}$  shown in Fig 23. So the rotation matrix should change to  $T_1$ , which is shown in equation (36).

$$\begin{aligned}
\mathbf{T}'_1 &= \begin{pmatrix} 1 & \frac{\sqrt{3}}{3} \\ 0 & \frac{2\sqrt{3}}{3} \end{pmatrix} \times \begin{pmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{pmatrix} \times \begin{pmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \\
&= \frac{\sqrt{3}}{3} \times T_1, \text{ where } T_1 = \begin{pmatrix} 1 & -2 \\ 2 & -1 \end{pmatrix}
\end{aligned} \tag{36}$$

It is ok that we have dropped the value  $\frac{\sqrt{3}}{3}$  which strictly equal to the transformation computation before  $T_1$ , cause that the final goal is to get the direction. We drop the value, but we did not change the direction.

But notice that since we did not follow the computation strictly, it probably will cause the value of  $p'_1$  and  $q'_1$  non co-prime. So before we use  $\begin{pmatrix} p'_1 \\ q'_1 \end{pmatrix}$  to next computation, we need to check if they are co-prime. If not, they need to be divided by their greatest common divisor.

**Example** In order to be more clear about what we have said, we will give an example: the original direction  $(p', q')$  for the hexagon is  $(1, 2)$ . After we right multiply  $T_1$  with its turn  $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ , we can get the direction for the first parallelogram— $\begin{pmatrix} -3 \\ 0 \end{pmatrix}$ , and its co-prime form is  $\begin{pmatrix} p'_1 \\ q'_1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ . And then we use it to right multiply the matrix  $T_2$ , we can get the direction for the second parallelogram:  $\begin{pmatrix} p'_2 \\ q'_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ . Finally we use  $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  repeat the same operation as we did in getting  $\begin{pmatrix} p'_2 \\ q'_2 \end{pmatrix}$ , we get the direction for the third parallelogram:  $\begin{pmatrix} p'_3 \\ q'_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

**Length in Parallelogram** Since we have got the method for changing the direction which adapt to different parallelogram, we can just consider one parallelogram for any direction  $(p'', q'')$  which means the parallelogram's own direction after we use either  $T_1$  or  $T_2$  transformation matrix. Our goal is to get the length in the parallelogram while we only know the information of the parallelogram (the length of edges and its inner angles) and projection angle  $(p'', q'')$ . From Fig 24 we can get the formula (37) which using parameters like  $HA$ ,  $HB$  and  $l$ .

$$P(p'', q'', n, x) = \begin{cases} l, & |x| \leq HA \\ \frac{HB-|x|}{HB-HA}l, & HA < |x| < HB \\ 0, & |x| \geq HB \end{cases} \tag{37}$$

where  $l$  means the longest length in the parallelogram when given a direction  $(p'', q'')$ , which equals to  $CD$  in Fig 24.

Now we have already got the function. The question now is how to get those parameters of the function such as  $HA$ ,  $HB$  and  $l$ . Considering the direction

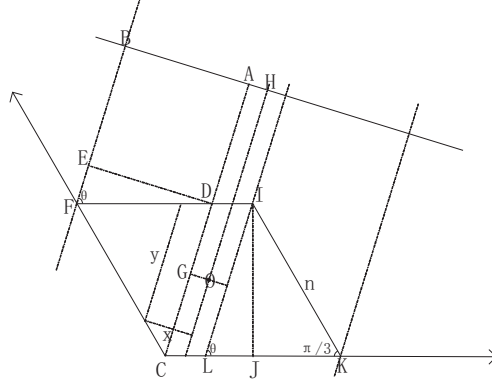


Figure 24 – Case 1.  $\frac{q''}{p''} \geq 1$  and  $p > 0, q > 0$

$(p'', q'')$  varies from 0 to  $\pi$ , we divided those directions into 5 cases to achieve our final goal.

**Case 1:**  $\frac{q''}{p''} \geq 1$  and  $p > 0, q > 0$ <sup>5</sup>

Before we start to calculate the parameters, we first need to know that the direction  $(p'', q'')$  are the direction in hexagonal lattice. The first thing that we need to do is to transfer it into the direction in square lattice using equation (32). And then we can get the direction  $(p, q)$  in square lattice.

From Fig 24, we can see that  $\tan \theta = \frac{q}{p}$ . At first, let us focus on the value of  $l$ . We can suppose the edge  $IK$  equals value  $n$ . From the Trigonometric Function we will know that the length of  $IJ = \frac{\sqrt{3}}{2}n$ . Then we will get the length  $l$  which equals to  $CD$  by equation (38).

$$l = \frac{\sqrt{3}}{2}n \times \sin \theta = \frac{\sqrt{3}}{2}n \times \frac{\sqrt{p^2 + q^2}}{q} \quad (38)$$

Next, we will consider the solution of  $HA$ . We can see that  $HA = OG$  and  $2 \times OG$  is the height of the parallelogram  $DILC$ . So  $2 \times HA \times l$  is the area of  $DILC$ . And the area of  $DILC$  equals to the area of the whole parallelogram minus twice the area of triangle  $IKL$ . The area of whole parallelogram is  $\frac{\sqrt{3}}{2}n \times n$ . The area of triangle  $IKL$  is  $(LJ + JK) \times IJ/2$ . So we can get the

<sup>5</sup>Note:  $p''$  and  $q''$  here means the direction in hexagon lattice.  $p$  means the direction in square lattice. From  $p''$  to  $p$ , we need the previously mentioned transform matrix  $H$  in (32). This kind of representation of directions will still be used in the following 4 cases.

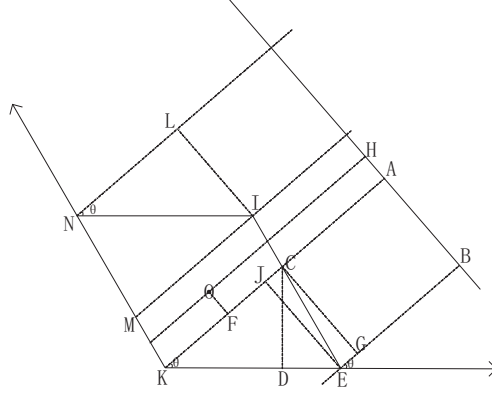


Figure 25 – Case 2.  $\frac{q''}{p''} < 1$  and  $p > 0, q > 0$

equation (39) to get the value of  $HA$ .

$$HA = \frac{q}{2\sqrt{p^2 + q^2}} \left[ n - \left( \frac{\sqrt{3}np}{2q} + \frac{n}{2} \right) \right] \quad (39)$$

At last, we go to the value of  $HB$ . It is clear that  $HB = HA + AB$ . So we only need to get the value of  $AB$ . In triangle  $DEF$ ,  $ED$  equals to  $AB$ , and  $DF$  equals to  $LK$  which we get the value in the solution of  $HA$ . So  $ED/DF = \sin \theta = q/\sqrt{p^2 + q^2}$ . Then we can get the equation (40) to get the value of  $AB$ .

$$AB = \frac{q}{2\sqrt{p^2 + q^2}} \left( \frac{\sqrt{3}np}{2q} + \frac{n}{2} \right) \quad (40)$$

Finally, we can get the value of  $l$ ,  $HA$  and  $HB = HA + AB$  from equation (38), (39) and (40). With these parameters, we can give a solution of function (37) when given a value of  $x$ . This is the end of *Case 1*.

**Case 2:**  $\frac{q''}{p''} < 1$  and  $p > 0, q > 0$

From Fig 25, we can see that  $\tan \theta = \frac{q}{p}$ . And  $\frac{CD}{\tan \theta} + \frac{CD}{\sqrt{3}} = KD + DE = n$ . Then we can get the value of  $CD = n / \left( \frac{p}{q} + \frac{\sqrt{3}}{3} \right)$ . Notice that  $l = CK = \frac{CD}{\sin \theta}$ . So we can get the equation (41) to get the value of  $l$ .

$$l = \frac{n\sqrt{p^2 + q^2}}{p + \frac{\sqrt{3}}{3}q} \quad (41)$$

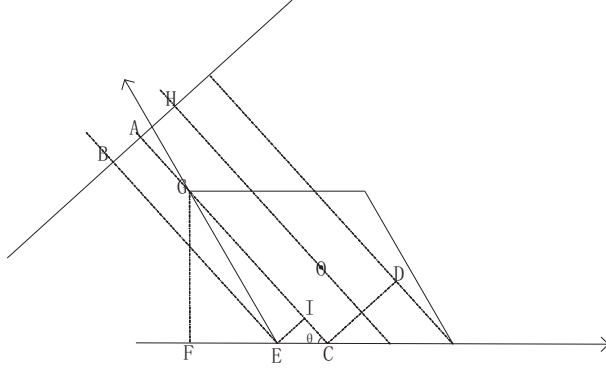


Figure 26 – Case 3.  $\frac{q''}{|p''|} \geq 1$  and  $p < 0, q > 0$

Next, it is  $HA$ . It is the same idea as we used in Case 1. The area of parallelogram  $ICKM$  equals to the area of the whole parallelogram  $IEKN$  subtract twice the area of triangle  $CEK$ . That means  $2 \times HA \times l = \frac{\sqrt{3}}{2}n^2 - n \times CD$ . Then we can get the equation (42).

$$HA = \frac{n(\sqrt{3}p - q)}{4\sqrt{p^2 + q^2}} \quad (42)$$

At last, we go to the value of  $HB$ . It is the same as Case 1. We need to know  $AB$  first. We know that  $AB = CG = JE$ , and  $JE/n = \sin \theta$ . So the equation is (43).

$$AB = \frac{nq}{\sqrt{p^2 + q^2}} \quad (43)$$

Finally, we can get the value of  $l$ ,  $HA$  and  $HB = HA + AB$  from equation (41), (42) and (43). With these parameters, we can give a solution of function (37) when given a value of  $x$ . This is the end of *Case 2*.

**Case 3:**  $\frac{q''}{|p''|} \geq 1$  and  $p < 0, q > 0$

From Fig 26, we can see that  $\tan \theta = \frac{q}{p}$ , and  $\sin \theta = \frac{q}{\sqrt{p^2 + q^2}}$ . And we also know that  $GF = \frac{\sqrt{3}}{2}n$  because the angle  $\angle GEF = \pi/3$  and  $GE = n$ . So

$l = CG = \frac{GF}{\sin \theta}$ . Then we can get the equation (44).

$$l = \frac{\sqrt{3}}{2} n \frac{\sqrt{p^2 + q^2}}{q} \quad (44)$$

Next, it is different from Case 1 and 2. That is we need to get  $AB$  first, and then use  $AB$  to get  $HA$ . Here we can see that  $CE = AB/\sin \theta = AB \frac{\sqrt{p^2 + q^2}}{q}$  and  $CE$  also equals to  $CF - EF$  which equals  $GF \frac{p}{q} - n/2 = \frac{np\sqrt{3}}{2q} - \frac{n}{2}$ . Then we can get the equation (45).

$$AB = \frac{n(\sqrt{3}p - q)}{\sqrt{p^2 + q^2}} \quad (45)$$

At Last, we go to  $HA$ . We still use the relation of areas to get the value. We know that  $2 \times HA \times l = \frac{\sqrt{3}}{2} n \times n - 2 \times \frac{1}{2} l \times AB$ . Then we get the equation (46).

$$HA = \frac{\frac{\sqrt{3}}{2} n^2 - l \times AB}{2l} = \frac{3nq - \sqrt{3}pn}{4\sqrt{p^2 + q^2}} \quad (46)$$

Finally, we can get the value of  $l$ ,  $HA$  and  $HB = HA + AB$  from equation (44), (46) and (45). With these parameters, we can give a solution of function (37) when given a value of  $x$ . This is the end of *Case 3*.

**Case 4:**  $\frac{q''}{|p''|} < 1$  and  $p < 0, q > 0$

From Fig 27, we can see that  $\tan \theta = \frac{q}{p}$ , and  $\sin \theta = \frac{q}{\sqrt{p^2 + q^2}}$ . We at first know that  $CE = CI \times \tan \theta = n \frac{q}{p}$ . Since we already know the value of  $CE$ , we can get the value of  $FG$  from the equation  $\frac{CE}{GF} = \frac{CI}{GI} = \frac{CI}{CI + CG} = \frac{n}{n + \frac{\sqrt{3}}{3} FG}$ . So  $FG = \frac{qn}{(p - q\sqrt{3}/3)}$ . And we also can get the value of  $EI$  which equals  $n/\cos \theta = n\sqrt{p^2 + q^2}/p$ . Since we get the values of  $EI$ ,  $FG$  and  $CE$ . We can get the value  $l$  cause  $l/EI = FG/CE$ . So we get the equation (47) to compute the value of  $l$ .

$$l = \frac{n\sqrt{p^2 + q^2}}{p - \frac{\sqrt{3}}{3}q} \quad (47)$$

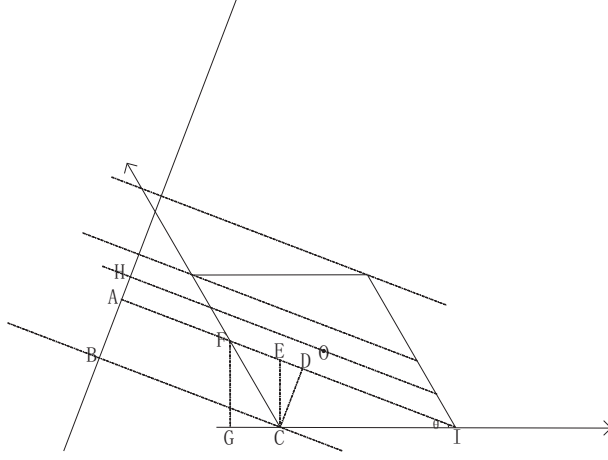


Figure 27 – Case 4.  $\frac{q''}{|p''|} < 1$  and  $p < 0, q > 0$

Next, it is like Case 3. Before we get the value of  $HA$ , we need to know the value of  $AB$  which equals to  $CD$ . And  $CD = CI \times \sin \theta$ . So the equation for  $AB$  is (48).

$$AB = \frac{nq}{\sqrt{p^2 + q^2}} \quad (48)$$

Finally, we can get the value of  $HA$ . The idea of area relationship is still used. From the equation  $\frac{\sqrt{3}}{2}n^2 - l \times AB = 2 \times HA \times l$ , we can get the equation (49) for  $HA$ .

$$HA = \frac{\sqrt{3}np - 3nq}{4\sqrt{p^2 + q^2}} \quad (49)$$

Finally, we can get the value of  $l$ ,  $HA$  and  $HB = HA + AB$  from equation (47), (49) and (48). With these parameters, we can give a solution of function (37) when given a value of  $x$ . This is the end of *Case 4*.

**Case 5:**  $\frac{q''}{p''} \geq 1$  and  $p < 0, q > 0$

From Fig 28, we can see that  $\tan \theta = \frac{q}{p}$ , and  $\sin \theta = \frac{q}{\sqrt{p^2 + q^2}}$ . We also know that  $l = JC = JF / \sin \theta$ . Then we get the equation (50) for  $l$

$$l = \frac{n\sqrt{3}\sqrt{p^2 + q^2}}{2q} \quad (50)$$



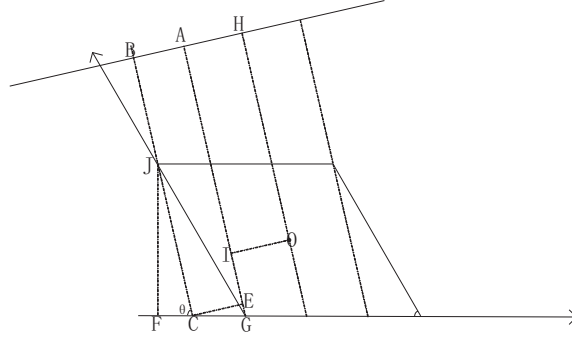


Figure 28 – Case 5.  $\frac{q''}{p''} \geq 1$  and  $p < 0, q > 0$

Next, we need to know the value of  $AB$ . Notice that  $GF = n/2 = FC + CG$ , while  $CG = CE/\sin \theta, CE = AB$  and  $CF = \frac{\sqrt{3}}{2}n \times \tan \theta$ . So we can get the value of  $AB$  by equation (51).

$$AB = \frac{n(q - \sqrt{3}p)}{2\sqrt{p^2 + q^2}} \quad (51)$$

At last, we use the same idea as usual – the relationship of area to get the value of  $HA$ . From the equation  $\frac{\sqrt{3}}{2}n^2 - 2 \times \frac{1}{2} \times l \times AB = 2 \times HA \times l$ . We get the equation (52) for  $HA$ .

$$HA = \frac{(q + \sqrt{3}p)n}{4\sqrt{p^2 + q^2}} \quad (52)$$

Finally, we can get the value of  $l, HA$  and  $HB = HA + AB$  from equation (50), (52) and (51). With these parameters, we can give a solution of function (37) when given a value of  $x$ . This is the end of *Case 5*.

At the end, it is interesting that we only considered the direction with  $q > 0$ . Cause the Farey Series we use only generates the directions in the range  $[0, \pi]$ .

**Length in Hexagon** Now we have the function of the length in parallelogram. Our job becomes how to combine the different lengths in different parallelograms. We already know the direction  $(p'', q'')$  given the value  $(p', q')$ . And

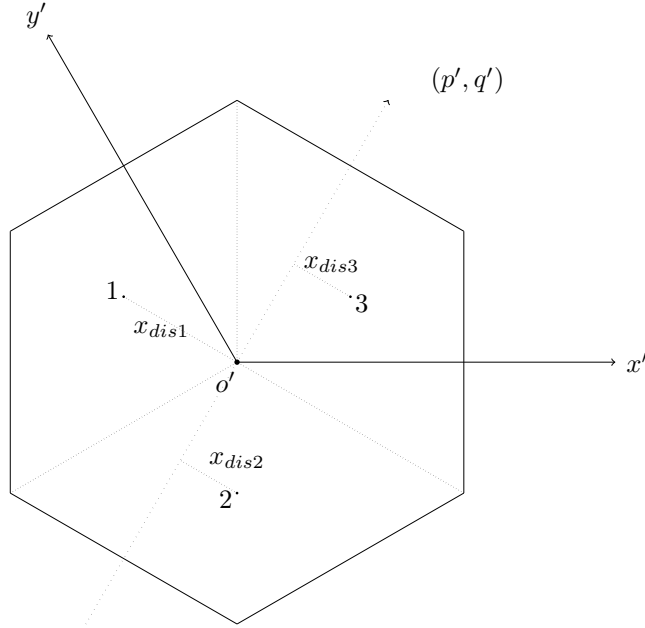


Figure 29 – The decomposition of an hexagon

the value of  $n$  is also given<sup>6</sup>. The job now is to give different values of  $x$  according to different parallelograms. From Fig 24 we can see that the value of  $x$  for one parallelogram is the distance to the center point  $H$  in the 1-Dimension projections, which is equivalent to the distance between to projection angle  $(p', q')$  and the centers of the parallelograms. It is more clearly in Fig 29.

Let us suggest that the function (37) as  $P(p'', q'', n, x)$ . So the final function for getting the length in an hexagon is (53).

$$L(x, y, p', q', k, l, n) = P(T(p', q'), n, x_{dis1}) + P(T(p', q'), n, x_{dis2}) + P(T(p', q'), n, x_{dis3}) \quad (53)$$

where

$$T(p', q') = Tr \times \begin{pmatrix} p' \\ q' \end{pmatrix}, \text{ where } Tr = T_1 \text{ if it is first parallelogram, otherwise } Tr = T_2 \quad (54)$$

and the values of  $x_{dis1}$ ,  $x_{dis2}$  and  $x_{dis3}$  can be got from equation (55).

$$D(x, y, p'', q'', k, l) = \frac{|-p \times center_x + q \times center_y + b|}{\sqrt{p^2 + q^2}}, \text{ where } b = py - qx \quad (55)$$

<sup>6</sup>We will give the value of  $n$  in Section 3.4.3

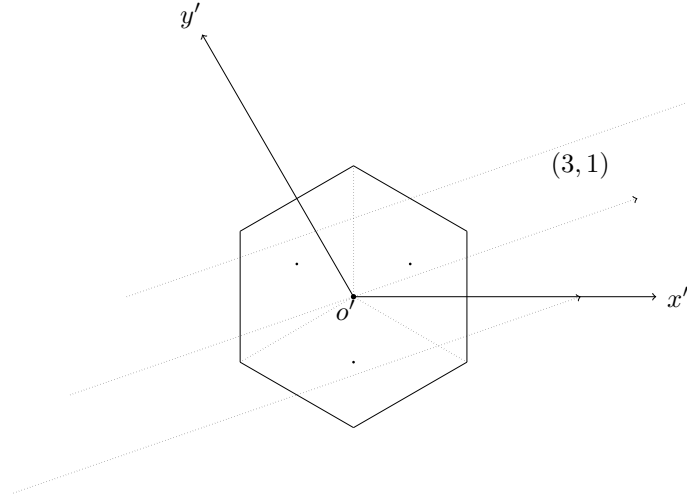


Figure 30 – The lines with direction  $(3, 1)$ (hexagonal direction) in the hexagon at origin points

where the value of  $(x, y)$  means the point that the projection line  $qX - pY + b = 0$  go through. And for the value of  $(p, q)$ , we use equation (32) to transfer it from hexagonal lattice to square lattice.

And the value of  $(k, l)$  is the center coordinate of one hexagon that the projection line go through. With the value of  $(k, l)$ , we can get the center coordinate  $(center_x, center_y)$  of different 3 parallelogram. For parallelogram 1, the center coordinate is  $(k' - 1/6, l' + 1/6)$ . And for parallelogram 2 and 3, they are  $(k' - 1/6, l' - 1/3)$  and  $(k' + 1/3, l' + 1/6)$ . Here  $(k', l')$  is the coordinate that transferred from square lattice coordinate  $(k, l)$  using equation (34).

**Kernel Generating** We have the function (53) to get the length in an hexagon now. Our next step is to get the kernel. The generating of the kernel can be considered as following: with a given direction  $(p', q')$ . we can get how many lines go through the original hexagon. It is clear to see Fig 30.

In Fig 30, there are 3 lines that go through the hexagon. With the function we previously mentioned, we can easily get their lengths in the hexagon. And these lengths consist of our kernel, that means  $kernel = (length_1, length_2, length_3)$ . But the question is how to determine the numbers of lines that go through the hexagon? We can see that the kernel is self-symmetric. So we only need to get half of the part of the kernel.

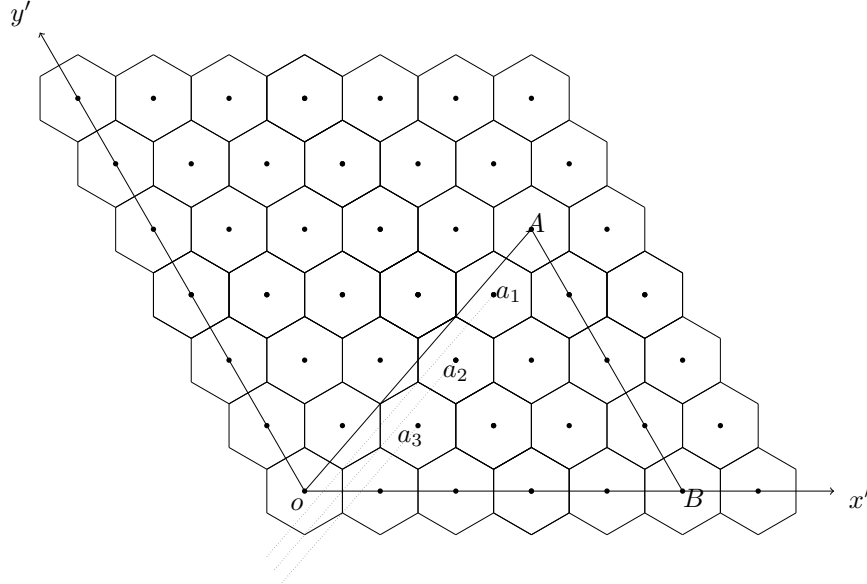


Figure 31 – The area of needed hexagons for generating kernel

In Fig 31, the direction is  $(5, 4)$ . We can see that only the triangle area  $OAB$  might give an valid length (the length is greater than 0) in the hexagon at origin point (the zero point  $o$ ). That means we only need to check the direction  $(p', q')$  which go through the point coordinate  $(x', y')$  in the range of  $p'y' - q'x' < 0$  and  $0 \leq x' < p$ . So only the points  $a_1(4, 3)$ ,  $a_2(3, 2)$  and  $a_3(2, 1)$  satisfy the restrictions.

After we found these points, we can use the function (53) to get the lengths and use these lengths to generate the kernel. Before we generate the kernel, the lengths we have got should be sorted twice. The first time is by descend order  $d\_order$  and the second time is by ascend order  $a\_order$ . So the kernel is  $[a\_order, center\_line, d\_order]$ . The  $center\_line$  is the length from the line in the same direction but go through the origin point. And in order to get the exactly result, we need to normalize the kernel by the equation (56), which makes  $\sum(kernel[1 : n]) = 1$ .

$$kernel[1 : n] = \frac{kernel[i]}{\sum_{i=1}^n kernel[i]} \quad (56)$$

### 3.3.3 Convolution with the Kernel

Since we have got the kernel, we can use the same idea as we used in the Spline 0 Mojette transform in square lattice. We convolve the bins from the Dirac



Figure 32 – Shepp-Logan phantom example. The size is  $256 \times 256$ .

Mojette transform in hexagonal lattice with the kernel. And then we get the bins for Spline 0 Mojette transform in hexagonal lattice.

## 3.4 Interpolation in Voronoi Cells

### 3.4.1 Shepp-Logan Phantom

The Shepp-Logan phantom[9] is consisted by 10 different ellipses. The gray level in each ellipse is different. And all the information of each ellipse is given such as the major axis, center and rotation angle  $\theta$ .

Because every parameter is given, we can easily get the ground truth for comparison. It is quite useful when we want to compare the different transformation methods either in 2-Dimension or in 3-Dimension. But here we only consider the 2-D condition. Fig 32 shows the Shepp-Logan phantom of size  $256 \times 256$ .

### 3.4.2 Shepp-Logan in Square Lattice

Since we have all the parameters of the 10 ellipses, we can easily draw the ellipses in the plane. But how can we use the square lattice to represent those

ellipses? Here we defined that the area of each Voronoi Cell is 1. And let us suggest the density of ellipse (in this thesis, the density is the gray level of the image) is  $v$ . If the whole cell is inside the ellipse, we can say that the area of the Voronoi Cell, we can also say the Square, represents the exactly the same value as the region in ellipse. So the value for the square is  $1 \times v = v$ .

But in fact things do not go such easily. Not all of the squares are inside the ellipse fully. As shown in Fig 33, the square labelled 1, 2, 3 are fully outside of the ellipse border and 7, 8, 9 are fully inside the ellipse. So the value for them is quite easy to get. They are 0 and  $v$  respectively. But for 4, 5 and 6 squares, only a part of them inside the ellipse. What is the proper value for them?

One solution for them is to compute the area of region inside the ellipse. It is quite accurate but hard to operate. And the computation is time-consuming. Obviously, this is not a good idea for implementation.

Our other option is to split the square into small squares to check whether the centers of the small squares are inside the ellipse. For any of them is inside the ellipse, we increase the counter by 1. After the iteration of all small squares, we check the value of counter which means the number of small squares that are inside the ellipse. Let us suppose the value of counter is  $n$ . And the total number of small squares are  $m$ . Then we get the suitable value for this cell:  $value = \frac{nv}{m}$ . Notice that the value of  $m$  should be 1, 4, 16, 25, ... which are the power of Natural Number. If we get a greater  $m$ , which means we divided the square into more small squares, we will get more accurate result, but it will spend more time to calculate the result.

**Example** It is easier for us to understand by an example. We still use Fig 33 as illustration. Here we split the square into 4 parts. So the value of  $m$  is 4. There is no doubt for the small Squares 1, 2, 3 and 7, 8, 9. The values are either 0 or  $v$ . For the Square 4, we find 4 points are all inside the ellipse, including the one on the border. So the value of Square 4 is  $4/4 \times v = v$ . And it is the same for Square 5. But for Square 6, only one point are inside the ellipse. So its value is  $1/4 \times v = \frac{v}{4}$ .

### 3.4.3 Shepp-Logan in Hexagonal Lattice

We face the same question when we change the lattice form square to hexagon. In order to make a valid comparison with square lattice, we also assume the area of an hexagon is 1. So here we can get the length of the edge in the parallelogram which have not given in Section 3.3.2 and Equation 53. From the Fig 22, we know the area of each parallelogram is  $1/3$  cause the area of the whole hexagon is 1. We still suppose that the edge of the parallelogram is  $n$ . And we can have the equation  $\frac{\sqrt{3}}{2}n^2 = \frac{1}{3}$ . Then we get the value of  $n$  which equals to  $\sqrt{\frac{2}{3\sqrt{3}}} \approx 0.6204$ .

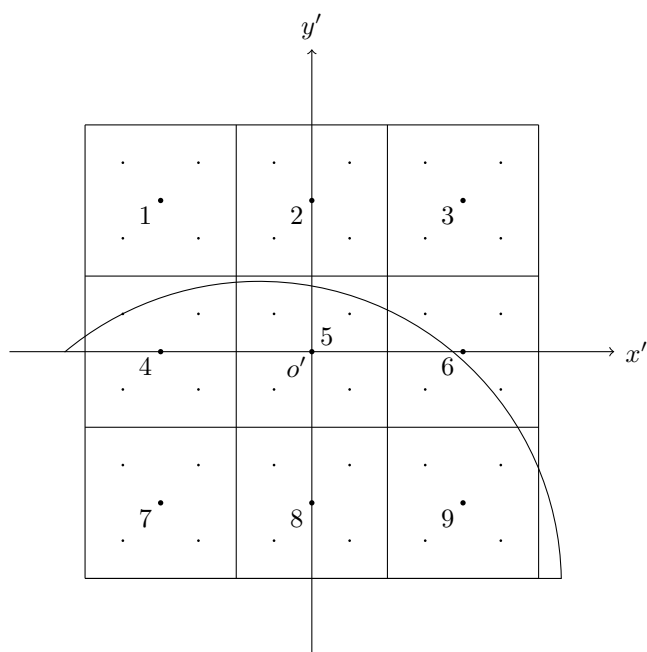


Figure 33 – The method of fill the Shepp-Logan phantom with square lattice

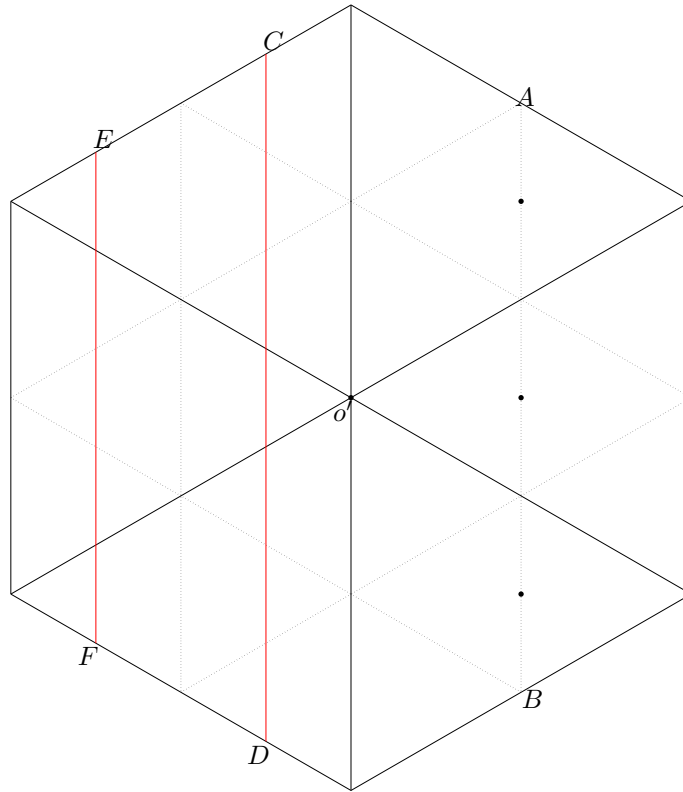


Figure 34 – The method of generating valid points in an hexagon

Since we get the value of  $n$ , we can know the center coordinate of each hexagon. The idea is the same. We need to divide one hexagon into small pieces. But we can not divide the hexagon into smaller hexagons(those smaller hexagons should in the same size) like that we did in squares(one square into smaller squares in same size). But we can split the hexagon into triangles in same size.

As shown in Fig 34, the first level is 6 triangles which is shown in solid lines. And the next level is 24 triangles with dotted lines. Notice that when we go to the next level, one triangle can split to the same 4 triangles. So the number of points should be  $6 \times 4^{k-1}$ ,  $k > 1$ , where  $k$  is the level we want.

Since we have got the way of splitting the hexagon, now our goal is to find the points for check if they are inside the ellipse. At first we think it is better to choose the gravity centers of those triangles as we did in square lattice. But it is hard to operate and we choose another simple but still effective way: find



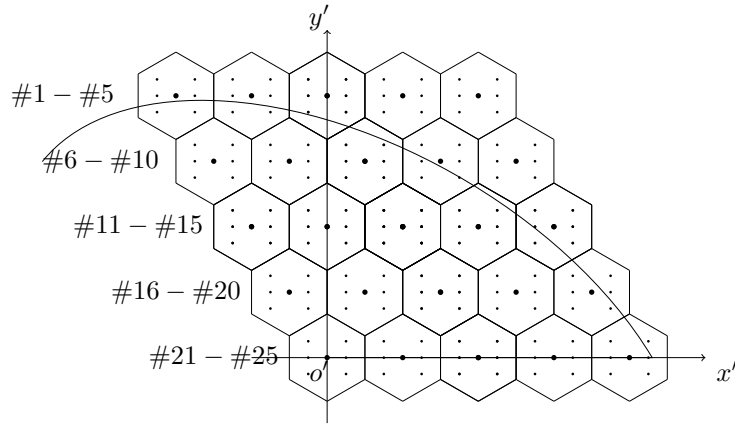


Figure 35 – The method of fill the Shepp-Logan phantom with hexagonal lattice

the points on the *Median Line* of those triangles. And the points locate exactly on the middle of the those *Median Line*. Seen from Fig 34, for the first level 6 triangles, those points locate on Line *AB* and its horizontal symmetry. And for the second level 24 triangles in dotted line, the points are on line *CD* and *EF* and their horizontal symmetries.

After we get the coordinate of the points, we need to check if they are inside the ellipse. The trick is the same as we did in square lattice. But the number of points are not the same. Here the possible point numbers are 1, 6, 24, 96, .... Then we can get the number of points which are inside the ellipse  $l$ , and its total number of points  $m$ . we now get the suitable value for the hexagon cell:  $value = \frac{lv}{m}$ , where  $v$  is the density or gray level as we mentioned in previous section.

**Example** In order to get a better understanding, we will get an example. The ellipse border and the hexagons are shown in Fig 35, We have labelled the number of each hexagon on the left. For the values of those hexagons which are completely outside or inside the ellipse are certain—0 for the former and  $v$  for the latter. But for the Hexagon 1, 2, 3, ... and so on, a part of the hexagon region is inside the ellipse. In this example, we only divide one hexagon into 6 triangles, which means there are only 6 points need to be checked. Here we only take the first 3 hexagons to calculate. For the first hexagon, there are 2 points inside the ellipse, so the value is  $2/6 \times v = \frac{v}{3}$ . And for the second hexagon, there are still 2 point inside the ellipse. So the value is the same as the first one— $\frac{v}{3}$ . But for the third one, there is only one point inside. Its value is  $\frac{v}{6}$ . And for saving the time, we do not continue to other points. They are based on the same idea.

### 3.5 Conclusion of the Jobs

In this section, we have addressed the works that we have done. We changed the Farey Series from square lattice to hexagonal lattice and give the transformation matrix to achieve the goal. Then we found that the Dirac Mojette transform in hexagonal lattice is the same as the one in square lattice. So our job focused on the Spline 0 Mojette transform. And then we gave the way of computing the kernel for the convolution in Spline 0 Mojette transform. At last, we gave the way that filling the values of the cells in both square lattice and hexagonal lattice.

## 4 Experiments

### 4.1 Getting Ground Truth

At the beginning of the experiments, we have to get the Ground Truth. Since we have used the Shepp-Logan phantom to generate the testing data, all parameter about the ellipses are known. Our job now is only to get the length inside each ellipse and multiply the length with its density or gray level (In order to get a better understand, we will use gray level to instead of density in the following part).

But before we get the length in the ellipse, we need to know the expression of the ellipse. In the paper of Shepp-Logan phantom, the author list 5 parameters for each ellipse. They are *center*  $(c_x, c_y)$ —the coordinate of gravity center, *Major Axis*  $a$ —the length of the major axis, *Minor Axis*  $b$ —the length of the minor axis,  $\theta$ —the rotation of the ellipse, and *gray level*—the density that we have mentioned above.

With those parameters, we can address the expression of the ellipse like equation (57). And we also can get the function for a certain line. Because the direction is the projection direction, and we know the certain point that the line go through. So the function of the line is shown in equation (58).

$$\frac{[\cos \theta(x - c_x) + \sin \theta(y - c_y)]^2}{a^2} + \frac{[\cos \theta(y - c_y) - \sin \theta(x - c_x)]^2}{b^2} = 1 \quad (57)$$

$$qx - py + b = 0, \quad (58)$$

where  $b = p \times oy - q \times ox$ ,  $(ox, oy)$  is the coordinate of a given point.

With Function (57) and (58), we can the the intersection point(s). There are 3 cases. The first one is that there is no solution for the 2 functions, which means that the intersection point of the ellipse and line does not exist. The second one is that only one point exists. So the length in ellipse is 0. The last one is that there are 2 intersection points, and the length in ellipse is the distance between 2 points. As shown in Fig 36, the length in ellipse is  $AB$  and also the distance between point  $A$  and  $B$ .

In Function 58, we mentioned that the value of  $(ox, oy)$  are given. We extract those coordinates from the edge of the matrix. For example, a  $256 \times 256$  image, we start from coordinate  $(-127, 128)$  and go straightly down to coordinate  $(-127, -127)$  and then go straightly right to  $(128, -127)$ . The step of movement is a parameter. If it is one, we can get totally  $128 + 128 - 1 = 511$  coordinates. So we can test 511 lines to get the ground truth distribution, which will be used for comparison. And if the step is 0.5, the number of lines can increase to  $(x + y - 1) * (1/step_{length}) - (1/step_{length} - 1) * 2 = 1020$ , where  $x = y = 128$ .

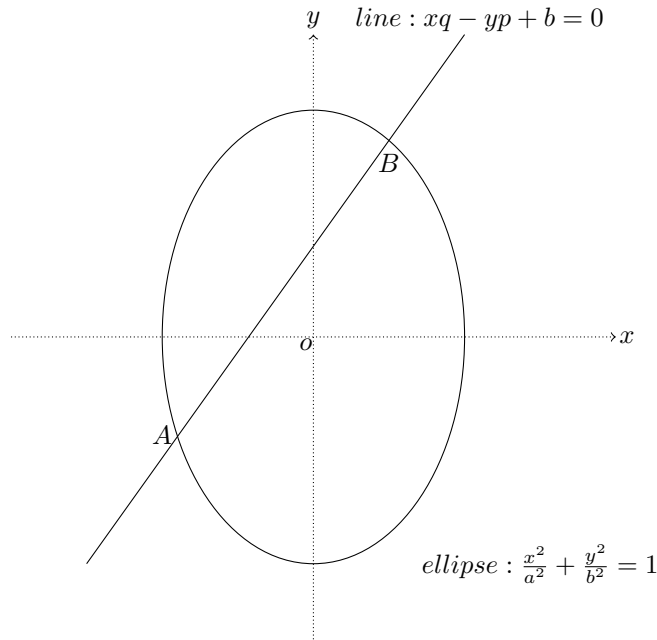


Figure 36 – The illustration of computing ground truth

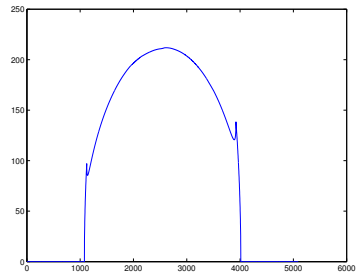
Since we get the ground truth and the bins including bins of Dirac and Spline 0 in both square lattice and hexagonal lattice, we can compare them by Correlation Coefficient(CC)<sup>7</sup>, which will check the linear relationship between the data and the ground truth. The example of bins and ground truth are shown as Fig 37.

## 4.2 The Experiments in Circles

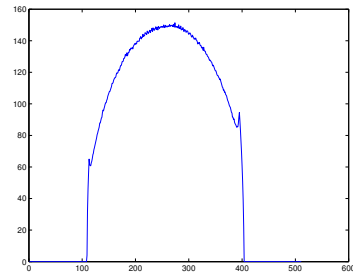
### 4.2.1 The Effect of Valid Points

Let us start from an easy way. We do not test with the ellipses at first, but we start with a circle. At the beginning, we will test the effect of the number of valid points. As we mentioned at the last part of previous section, in order to fill those cells we divide the hexagons or square into small pieces. Here for square lattice we have tested the following numbers of valid points: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. And for hexagonal lattice we have tested the numbers like [1, 6, 24, 96, 384,  $96 * 4^2$ ,  $96 * 4^3$ ,  $96 * 4^4$ ]. And the result are shown in Fig 38. The  $y$  axis stands for the CC value of certain valid point number. The  $x$  axis means the location in testing ranges(e.g. 1 mean the first number of testing range [1, 4, 9, 16, 25...] or [1, 6, 24, 96, 384...] according to different lattices).

<sup>7</sup>In the following sections, we will use CC instead of Correlation Coefficient



(a) The plotting of the ground truth



(b) The plotting of the bins of spline 0

Figure 37 – The plotting of the bins

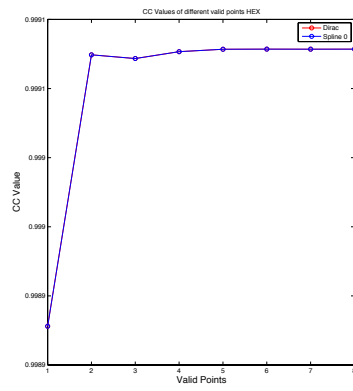
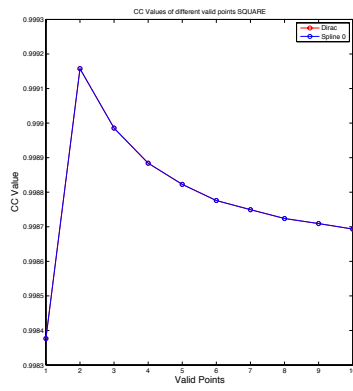


Figure 38 – The effect of valid points

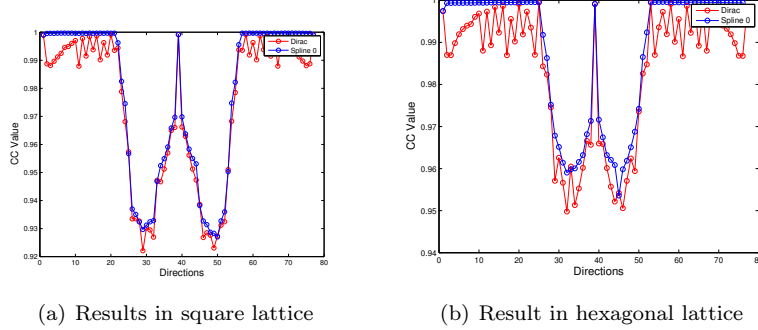


Figure 39 – The effect of different directions in both square and hexagonal lattice

Seen from the Fig 38, we can see that the CC values of hexagonal lattice are totally higher than those of square lattice. And it is interesting that in square lattice, the values drop with the increasing number of valid point. The reason why it drops and why the hexagonal lattice is better maybe due to the excellent perform of the hexagonal lattice on the previously mentioned *Covering* and *Packing* problems.

#### 4.2.2 The CC on Different Directions

In this experiment, we will test the CC by different directions which generate by Farey Series. The Fig 39 shows the results of the experiment. We have generate the directions from 0 to  $\frac{\pi}{2}$  (or 0 to  $\frac{2\pi}{3}$  in hexagonal lattice) by  $F(16)$ , which is the direction  $(p, q) \in [(1, 0), (16, 1), (15, 1), (14, 1) \dots (1, 1), (5, 8), (3, 5) \dots (0, 1)]$ . And of cause, those direction expressions are in their coordinate system. Before using the direction in hexagonal lattice, we need to transform the directions into ordinary square coordinate system by the transform matrix (32).

In Fig 39,  $y$  axis represents the CC value. And  $x$  axis means the location of the projection direction in  $F(16)$ . For example, the 1 means the first array in  $F(16)$  which is  $(1, 0)$ .

From the figure, we can see that the CC value of Spline 0 Mojette transform in both square and hexagonal lattice are greater than those of Dirac Mojette transform. It proves that the Spline 0 Mojette transform performs better than Dirac Mojette transform either in square lattice or in hexagonal lattice. And it is much more stable than those in Dirac.

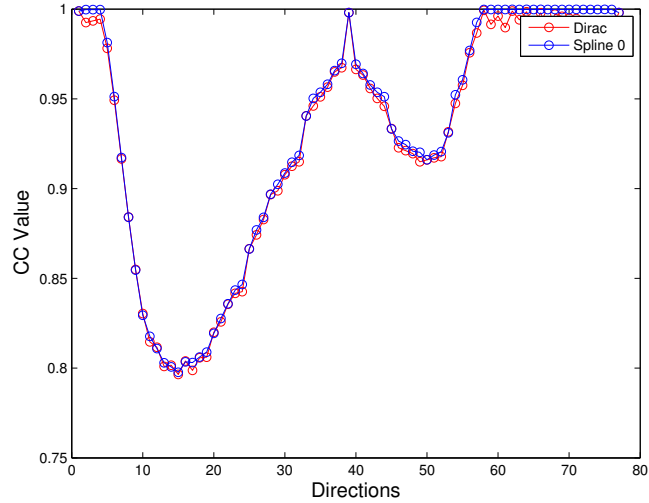


Figure 40 – The CC values of an ellipse in square lattice

Through we seen the values drop and increase sharply in both 2 figures, we notice that the real difference in digits are not so great. The highest one in both 2 figures are nearly 1 and the lowest ones are around 0.92 and 0.95 respectively. It is understandable that for a circle any direction is the same in getting the length in it. And we also can see that the values in hexagonal lattice are totally above those in square lattice, which also proves that hexagonal lattices performs *BETTER*. We also notice that the highest value for both are in the same direction –  $(1, 1)$  in their own coordinate system. It is interesting that the CC values near  $(1, 1)$  drop and those near  $(1, 0)$  and  $(0, 1)$  get high CC value. We have not got any exactly explanation for it yet. Maybe it is due to the missing directions in Farey Series (Because they are discrete directions) or the cell value in the border changes sharply.

### 4.3 The Experiments in One Ellipse

Since we have got the first results from the previous simple experiment. Now we can go to a more complex one. In this experiment, we change the circle to an ellipse which is the first ellipse in Shepp-Logan Phantom. We also need to test the CC values in different directions. Here the same as we did in the last experiment, we use  $F(16)$  as the directions from 0 to  $\frac{\pi}{2}$ . The results are shown in Fig 40 and 41.

In both Fig 40 and 41,  $y$  axis represents the CC value. And  $x$  axis means the location of the projection direction in  $F(16)$ . For example, the 1 means the first

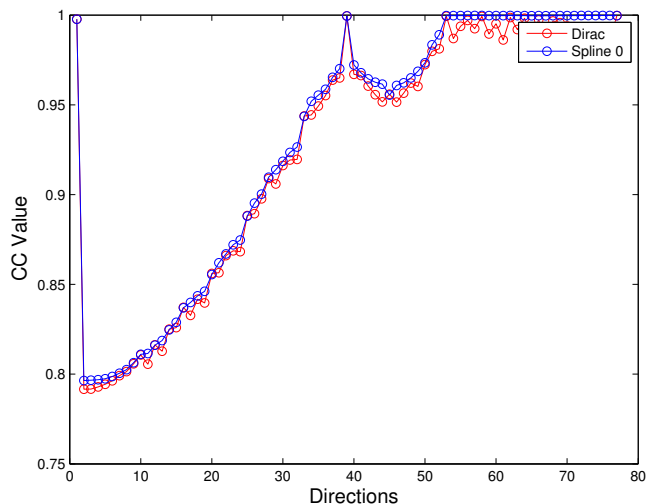


Figure 41 – The CC values of an ellipse in hexagonal lattice

array in  $F(16)$  which is  $(1, 0)$ .

From the figures we can see that the Spline 0 Mojette transform performs better than Dirac Mojette transform both in 2 lattices. And it is much more stable. And for the directions in the first ten position of the Farey Series, the CC values of square lattice are better than those of hexagonal lattices. But the rest are the opposite. Although for both hexagonal lattice and square lattice, they have their own strong aspect(directions), in general we can still say that the hexagonal lattice plays *BETTER* perform.

#### 4.4 The Experiments in Shepp-Logan Phantom

Now the warm-up is over. We need to go to the experiments on Shepp-Logan Phantom. In this experiment, we change the single ellipse to the whole 10 ellipses of Shepp-Logan Phantom. The same as we did before, we need to test the CC values in different directions. We still use  $F(16)$  to generate those directions from 0 to  $\frac{\pi}{2}$ . The results are shown in Fig 42 and 43.

In both Fig 42 and 43,  $y$  axis represents the CC value. And  $x$  axis means the location of the projection direction in  $F(16)$ . For example, the 75th direction means the first array in  $F(16)$  which is  $(0, 1)$ .

From the figures we can see that the Dirac Mojette transform really plays bad. Its CC values are lower than those of Spline 0 in both 2 lattices obviously



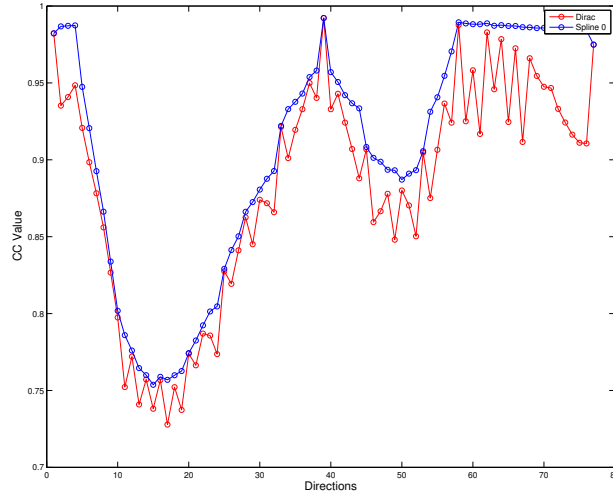


Figure 42 – The CC values of Shepp-Logan Phantom in square lattice

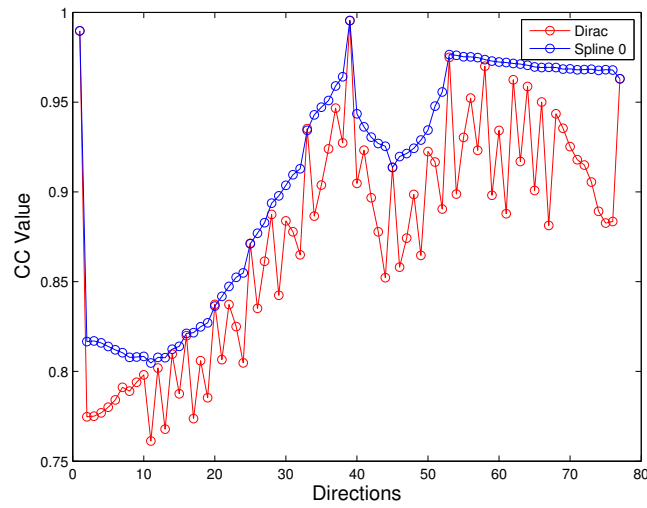


Figure 43 – The CC values of Shepp-Logan Phantom in hexagonal lattice

and more unstable. Compared with the result of the last experiment, we can see that the trend of CC are totally the same. But we can also notice that the CC value are lower than the last one because of the effect of multiple ellipses. Although the average of the CC value are both reduced, but the CC in the best projection direction  $(1, 1)$  in different respective coordinate system still keep high value, which approximately equals to 1.

Comparing the 2 different data, we find that the CC values in hexagonal lattices are all above 0.75, especially for Spline 0, the values are all above 0.8. They are obviously higher than those in square lattice. Although it is the same as we found in last experiment—the directions in first ten position of Farey Series in square lattice got higher CC than those in hexagonal one, it can not change the fact that hexagonal lattice really plays *BETTER* than square lattice. It is even better in Shepp-Logan Phantom than in one single ellipse.

#### 4.5 Conclusion of Experiments

In this section, we have addressed the way of getting ground truth. And we also proposed the method of comparing the bins and ground truth using Correlation Coefficient(CC). We have tested the result on three objects—one circle, one ellipse(the first ellipse in Shepp-Logan Phantom) and the Shepp-Logan Phantom. In these experiments, we found that hexagonal lattice really play better in both filling cells and the linear relationship with ground truth than square lattice.

## 5 Conclusion

In this thesis we have proposed a new way of Mojette transform using not square but hexagonal lattice. In order to get a better understanding, we discussed 2 main knowledges that are strongly related to our topic at first. Then we addressed the improved algorithm of generating Farey Series in hexagonal lattice and its transformation matrix. And we represented the method of Dirac and Spline 0 Mojette transform in hexagonal lattice. Especially in the Spline 0 Mojette transform, we have changed the algorithm of generating the kernel. We also gave a way of computing the ground truth which is used in experiments and how we compare the ground truth with our data. Finally during experiments on 3 different object, we found that hexagonal lattice really play better in both filling cells and the linear relationship with ground truth than square lattice.

Because of insufficient time, we did not go to back projection part of Mojette transform. Our next step can focus on this part. Besides, we noticed that the algorithm for generating kernel is quite complex, we need to find a way to simplify it. And as we mentioned in experiments part, we have found the unusual decreasing area of Correlation Coefficient value near the respective  $(1, 1)$  direction in both square and hexagonal lattice. So this can be our another goal. And finally, from the results of experiments we can see that both 2 lattices have their own strong directions, which means in those directions the CC value is better than another one. It is also interesting if we can combine 2 lattices together to get a better performance.

## References

- [1] *Hadamard matrices of order 16*. Jet Propulsion Lab. Research Summary 36–10, Sept. 1 1961. pp.21–26.
- [2] *Maximal finite groups of integral  $(n \times n)$ -matrices and full groups of integral automorphism of positive quadratic forms (Bravais lattices)*. TMIS 128, p.183–211, 261=PSIM128, p. 217-250, Sept. 1 1972.
- [3] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, chapter Coverings, Lattices and Quantizers, page P.33. Springer-Verlag, 2nd edition edition, 1992.
- [4] Jeanpierre Guédon and Nicolas Normand. Spline Mojette transform. applications in tomography and communications. In *EUSIPCO 2002*, volume II, pages 407–410, Toulouse, September 2002.
- [5] C. Kittel. *Introduction to Solid State Physics*. Wiley, NY, 5th ed., 1976.
- [6] Servieres M. *Reconstruction tomographique Mojette*. PhD thesis, University of Nantes, 2005.
- [7] Servieres M., Normand N., Guedon J., and Bizais Y. "the mojette transform: Discrete angles for tomography". *Discrete Tomography and its applications*, vol. 20 of *Electronic Notes in Discrete Mathematics*(p. 587–606), 2005.
- [8] W. Plesken. *Bravais groups in low dimensions*. In *Comm. Match. in Math. Chem.*, pages 97–119, Oct. 1981.
- [9] Lawrence A. Shepp and Benjamin F. Logan. The Fourier reconstruction of a head section. 21(3):21–43, June 1974.