HAL
open science

Shear based Bijective Digital Rotation in Triangular Grids

Éric Andrès, Gaëlle Largeteau-Skapin, Rita Zrour

To cite this version:

Éric Andrès, Gaëlle Largeteau-Skapin, Rita Zrour. Shear based Bijective Digital Rotation in Triangular Grids. 2018. hal-01900149

HAL Id: hal-01900149

https://hal.science/hal-01900149v1

Preprint submitted on 21 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DRAFT

Shear based Bijective Digital Rotation in Triangular Grids

Eric Andres, Gaëlle Largeteau-skapin, Rita Zrour[a]

[a] *University of Poitiers, Laboratory XLIM, ASALI, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France*

Abstract

In this paper we are proposing a way to perform bijective digital rotations on a triangular cell grid. The method is based on a decomposition of a rotation into shear transforms and on a way to transform the original triangle centroids in order to have a regular grid. The rotation method works for any angle and achieves an average distance between the digital rotated point and the continuous rotated point of about 0.5 (with 1.0 the side of a triangle and a distance between two neighboring triangle centroids of 0.57).

Keywords: Triangular grid, Rotation, Bijective Rotation, Shear Transforms

1. Introduction

The aim of this paper is to propose an algorithm for performing a bijective digital rotation on triangular grids. On one hand, rotation transforms are one of the most basic and fundamental operations in fields like computer graphics, computer vision, pattern recognition, etc. On the other hand, triangular grids (Freeman (1979), Nagy (2003a,b, 2004)) have enjoyed a continuous interests over the years in various fields like geometric modelling Botsch et al. (2006), computer vision Cooper et al. (2005), Ogawa (1986), image processing Sun et al. (2002), digital tomography Nagy and Moisi (2017), networks Koster and Muoz (2009), etc. To the best of the author's knowledge, there are very few papers yet on triangular cell grid transforms in general and none on digital rotations for such grids.

Let us first note that there are two different types of triangular grids in the literature, based on a triangle plane tessellation: the *triangular vertex grid* defined by the triangle vertices and the *triangular cell grid* formed by the centroids of the triangles. It is the latter, the triangular cell grid, that we are considering since we are interested in triangular grid image transforms. In the classical triangular cell grid, each triangle is identified by its centroid. The problem with this is that there are two types of cells (*odd* and *even* triangles) and, more importantly, the centroids do not form a *regular* 2*D grid* (defined by a point and two vectors). One of the first steps of our method is therefore to propose a small modification of the triangular cell grid in order to create a new grid that is regular. Each triangle is now identified by a point that is slightly different from the classical centroid.

Different methods have been used to rotate an image. There exists angles for which digitized rotations (a continuous rotation followed by a rounding operator) are bijective in the square grid Jacob and Andres (1995), Nouvel and Rmila (2005), Roussillon and Coeurjolly (2016). Recently, Pluta et al. (2018) showed that there also exists angles for which such a digitized rotation is bijective in the hexagonal grid. The problem is that the digitized rotations are not bijective for all angles. Paeth-Tanaka proposed an approach Paeth (1986), Tanaka et al. (1986), that consists in decomposing a rotation matrix into three shear matrices. They used this decomposition to perform antialiased image rotations (bijectivity was not considered). In this paper we are following the idea of Paeth-Tanaka

*Corresponding author

Email address: `{eric.andres ; gaelle.largeteau.skapin ; rita.zrour}@univ-poitiers.fr` (Eric Andres, Gaëlle Largeteau-skapin, Rita Zrour)

by considering a rotation decomposition into shear matrices that are each combined with a rounding function. The idea is to push grid cells (triangles) rows (aligned cells) by an integer number of *cell spots* in a given direction. Pushing a cell row by an integer number of cells means that the whole process is reversible and that the resulting transform will be bijective. The idea of pushing rows of pixels has been originally introduced by J-P Reveillès (1991), improved by Andres (1996) for square grids and recently adapted to the hexagonal grid by Andres (2018).

The rotation method works for any angle and achieves an average distance between the digital rotated point and the continuous rotated point of about 0.5 (with 1.0 the side of a triangle and a distance between two neighboring triangle centroids of 0.57).

The organization of the paper is as follows, Section 2 presents the basic notions and notations. Section 3 describes the rotation in the triangular grid. Section 4 provides examples as results and qualitative comparison with continuous euclidean rotation. The last section gives some conclusions and perspectives.

2. Triangular grid and Shear transforms

2.1. Description of the Triangular grid

We are considering a triangular cell grid with equilateral triangles of side one. One side of the cells is parallel to the Euclidean abscissa axis of the regular coordinate system. As shown in Figure 1, we consider a two coordinates system, where the tx-axis is the horizontal axis defined by the vector $(1, 0)$ (in the triangular grid coordinate system) and the ty-axis the vertical axis defined by the vector $(0, 1)$ (in the triangular grid coordinate system). The origin cell has the coordinates $(0, 0)$ (in the triangular grid coordinate system). B. Nagy (2012, 2015) proposed a coordinate system with three coordinates instead of two. It is also the coordinate system we used in a previous paper on triangular grids Dutt et al. (2018). Let us note that our two coordinate system and Nagys' three coordinate system are linked by simple formulas ($a\%b$ is the Modulo of a by b):

- $(x, y, z) \mapsto (x - z, -y)$;
- $(x, y) \mapsto \left(\frac{x+y+(x+y)\%2}{2}, -y, \frac{-x+y+(x+y)\%2}{2}\right)$.

There are two types of triangles. The up-pointing triangles $\triangle$ (yellow triangles in our figures), are called *Even*, the down-pointing triangles $\triangledown$ (blue ones), are called *Odd*. Using our triangular coordinate system, we can see that an *Even* (resp. *Odd*) triangle of coordinates (i, j) has an even (resp. odd) sum $i + j$.

Traditionally, for cell transforms, the centroid is considered as *representative* of the cell. If we consider the triangle centroids, we can see that *Even* and *Odd* triangles centroids do not form a regular grid defined by two vectors and a center (see Figure 1). The centroids of the triangles form two intertwined grids. One solution could be to rotate each grid separately but this poses the problem of the rotation center which cannot be a grid point for both grids. To avoid this problem, we propose to change the point that will represent the cell in the transform. The new triangular grid points are defined as follows:

- For even triangles, gridpoint = centroid $+\left(\frac{\sqrt{3}}{12}, 0\right)$;
- for odd triangles, gridpoint = centroid $-\left(\frac{\sqrt{3}}{12}, 0\right)$.

In Figure 1, the red circles show the triangle centroids while the red disks show the grid points we are considering. As one can see, the centroids do not form a regular grid while the new grid points form a (rectangular) regular grid defined by the center $(0, 0)$ and the Euclidean vectors $\left(\frac{1}{2}, 0\right)$ and $\left(0, \frac{\sqrt{3}}{2}\right)$.

The coordinate transform from our triangular grid points to cartesian grid and vice versa are given by the following transforms :

$$Tri2Cart: \quad (x_t, y_t) \mapsto (x, y) = \left(\frac{x_t}{2}, \frac{\sqrt{3}y_t}{2}\right) \tag{1}$$

$$Cart2Tri: \quad (x, y) \mapsto (x_t, y_t) = \left(2x, \frac{2y\sqrt{3}}{3}\right) \tag{2}$$

2.2. Shear Transforms for the Triangular Grid

The idea is to propose a rotation decomposition into a sequence of shear transforms in the (new) triangular coordinate system. A *shear transform* is a linear mapping that translates a point in a given direction by a vector proportional (by a shear factor) to the signed distance to a line parallel to that direction. Shear transforms preserve

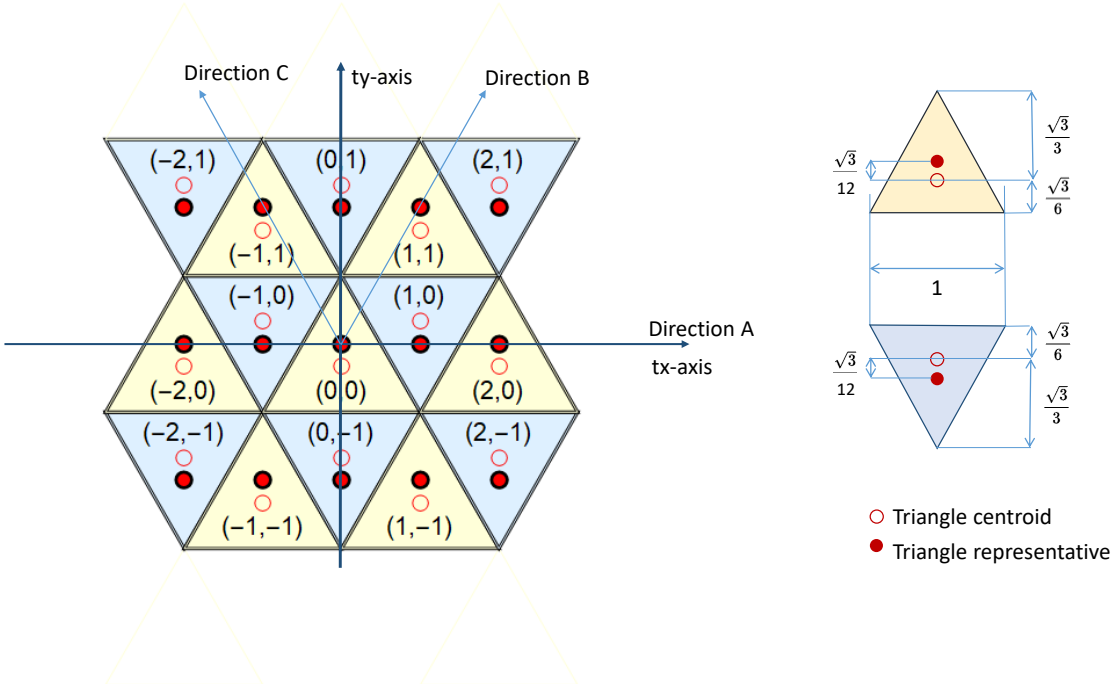

Figure 1: a) Coordinate system on the triangular grid. The yellow triangles are called *Even*, the blue ones are called *Odd*. The centroid of each triangle (empty circle) on the grid doesn't form a regular grid which brought us to translate the centroids by $+\sqrt{3}/12$ and $-\sqrt{3}/12$ for respectively *Even* and *Odd* triangles on the Cartesian grid to obtain a regular grid (full red circles). b) Triangle size.

areas and because of this, it is quite natural to decompose isometries into sequences of shear transforms. As an example, the decomposition of a rotation into three shear transforms with the Cartesian axes as shear lines (an 'ULU' decomposition with '1's on the diagonals) used by Paeth (1986) and Tanaka et al. (1986):

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \sin\theta & 1 \end{pmatrix}\begin{pmatrix} 1 & -\tan\frac{\theta}{2} \\ 0 & 1 \end{pmatrix}$$

Our goal is to propose a bijective rotation and for that, following the original idea of Reveillès (1991), we are going to *push rows* of triangular cells. The choice we made in this paper is to consider shear transforms corresponding to the directions of angle 0, $\pi/3$ and $2\pi/3$. We have called those directions, for the sake of simplicity, directions A, B and C (See Figure 1). Direction A corresponds to a translation along the tx-axis: vector $(1,0)$. Direction B corresponds to a translation of vector $(1,1)$ in the triangular coordinate system. Direction C corresponds to a translation of vector $(-1,1)$ in the triangular coordinate system. The triangles can be pushed in those three directions without tearing the grid apart. Those transforms are linked to the following matrices in the triangular coordinate system:

$$matA_{tri} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \quad matB_{tri} = \begin{pmatrix} 1+b & -b \\ b & 1-b \end{pmatrix}$$

$$matC_{tri} = \begin{pmatrix} 1-c & -c \\ c & 1+c \end{pmatrix}$$

Figure 2 shows the effect of the shear transforms for coefficients $a = b = c = 1$. In order to explain what those matrices mean, let us define three type of cell rows: let us call an A-row a set of cells whose triangular ty-coordinate is constant: $\mathcal{A}_s = \{(x,y) \in \mathbb{R} : y = s\}$ is an A-row. For a point (x,y), $matA_{tri} \cdot (x,y) = (x + ay, y)$, so $matA_{tri} \cdot \mathcal{A}_y =$

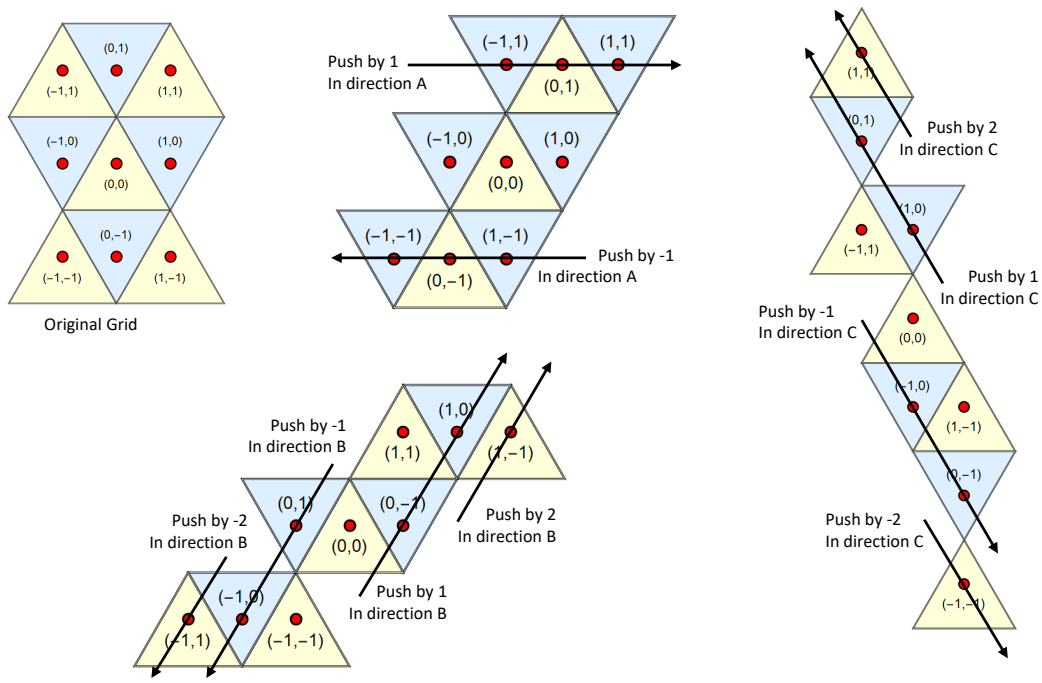

Figure 2: The three Shear transforms in directions A, B and C for shear coefficients $a = b = c = 1$.

$\mathcal{A}_y$. Let us call B-row a set of cells such that the difference of their abscissa and ordinates are constant (in the triangular coordinate system): $\mathcal{B}_s = \{(x, y) \in \mathbb{R} : x - y = s\}$ is a B-row. For a point (x, y), $matB_{tri} \cdot (x, y) = (x + bx - by, y + bx - by)$, so $matB_{tri} \cdot \mathcal{B}_s = \mathcal{B}_s$. Let us call C-row a set of cells such that the sum of their abscissa and ordinate are constant (in the triangular coordinate system): $C_s = \{(x, y) \in \mathbb{R} : x + y = s\}$ is a C-row. It is easy to see that in the same way, $matC_{tri} \cdot C_s = C_s$.

Note that a row corresponds to a set of triangles with aligned grid points. As such, a given B-row or a given C-row is composed of the same type of triangles (odd or even). Only an A-row is composed of odd and even triangles.

3. Bijective Digital Rotation

Let us now find values for a, b and c such that the composition of three such shear transforms corresponds to a rotation of center $(0, 0)$ and angle θ. For the sake of simplicity, we will write simply a, b, c instead of $a(\theta), b(\theta)$ and $c(\theta)$. Since it is simpler to express the rotation matrix in the classical Cartesian coordinate system, we solved the following equation:

$$Tri2Cart \circ matA \circ matB \circ matC \circ Cart2Tri = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

which has a unique solution for a, b and c (function of θ):

$$a = -2 + \frac{-6 + 4\sqrt{3}\sin\theta}{-3\cos\theta + \sqrt{3}\sin\theta} \quad (3)$$

$$b = \frac{1}{6}\left(3 - 3\cos\theta + \sqrt{3}\sin\theta\right) \quad (4)$$

$$c = -\frac{a}{4} \quad (5)$$

3.1. Dealing with divergence

Let us first note that we consider here, that all angles are between $-\pi$ and π. The solutions (a, b, c) do not represent a complete solution since the denominator of a (and equivalently c) has a value zero for angles θ equal to $-2\pi/3$ and $\pi/3$. The zeros for both angles are not of the same nature.

As can be seen in Figure 3, the zero in $\pi/3$ is a 0/0 singularity where $lim_{\theta\to\frac{\pi}{3}}(a) = -1$ and $lim_{\theta\to\frac{\pi}{3}}(c) = 0.25$. To avoid this singularity problem, one can simply set the values of coefficients a and c by their limit value for angle $\theta = \pi/3$ (see line 1 in Algorithm 1 and Algorithm 2).

The problem for angle $-2\pi/3$ has to be dealt in an entirely different way. As one can see in Figure 4, the error measures (see section 4) start to increase rapidly when approaching the angle $-2\pi/3$. The idea here is to avoid altogether the problem of the interval from $-\pi$ to 0 and simply consider that a digital rotation of such an angle is the inverse of the digital rotation of angle $-\theta$ (that falls then in the interval 0 to π). That way we avoid the problems around $-2\pi/3$ and ensure that the digital rotation of angle θ is the inverse of the digital rotation of angle $-\theta$ (note that it is an important side effect as a bijective transform is not always easily invertible). Note that the decomposition proposed by Paeth (1986) and Tanaka et al. (1986) has a similar divergence problem around angle π.

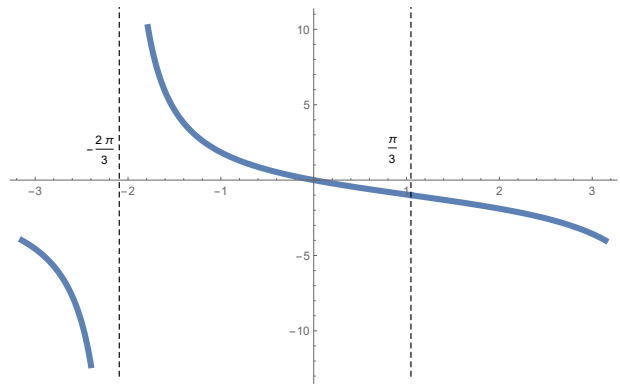

Figure 3: The graph of coefficient $a = -2 + \frac{-6+4\sqrt{3}\sin\theta}{-3\cos\theta+\sqrt{3}\sin\theta}$.

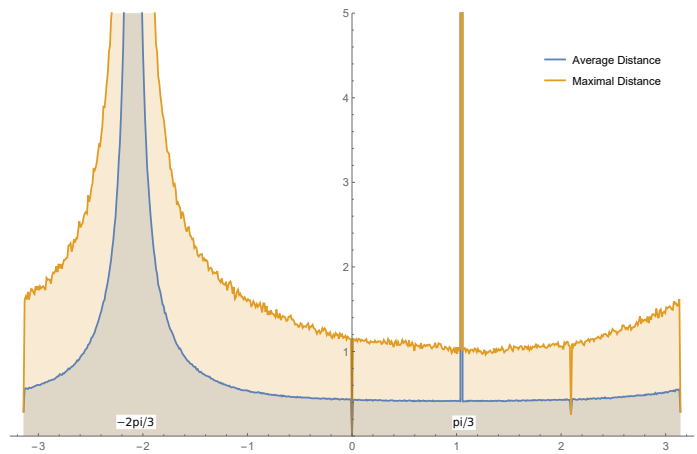

Figure 4: Average (AD) and Maximal (MD) Distance error criteria for Algorithm 1, RotCBA, for angles $-\pi$ to π

3.2. Algorithms

Let us now focus on the algorithm. We have found a shear based decomposition for a rotation that preserves A, B and C-rows respectively. Of course, a triangular cell is only transformed into another triangular cell if the coefficients a, b and c are integers. For this, we simply considered the $Floor(x + 0.5)$ function ($Floor(x)$ is the biggest integer smaller or equal to x). Careful here because the classical rounding function might lead to errors since it is not translation invariant. Some attention has also to be taken for values b and c for angles very close to $\pm\pi/3$. Computation errors may lead to slightly different values for b and c depending if you compute $b(\theta)$, $b(-\theta)$ and $b(2\pi - \theta)$ (the problem is similar for c). Since $b(\pi/3) = 0.5$, the rounding of $b(x-y)$ for angles very close to $\pm\pi/3$ may lead to incorrect values for an odd $x - y$. A simple way around this problem is to consider that for all angles close to $\pi/3$, we consider a b value equal to 0.5 and a c value equal to 0.25. The term *close* here can be defined based on the size of the image that will be rotated. That is also the reason why this algorithm is set up for angles

Algorithm 1: $RotCBA(x, y, \theta)$: Point Rotation of center $(0, 0)$ and angle θ

Input : $(x, y) \in \mathbb{Z}^2, 0 \leq \theta \leq \pi$
Output: $(x_3, y_3) \in \mathbb{Z}^2$

1 If $\theta = \frac{\pi}{3}$ Then $\begin{pmatrix} a \\ c \end{pmatrix} \leftarrow \begin{pmatrix} -1 \\ 0.25 \end{pmatrix}$
2 Else $a \leftarrow -2 + \frac{-6+4\sqrt{3}\sin\theta}{-3\cos\theta + \sqrt{3}\sin\theta}$
3 $\quad c \leftarrow -a/4$
4 $b = \frac{1}{6}(3 - 3\cos\theta + \sqrt{3}\sin\theta)$.
5 $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \leftarrow \begin{pmatrix} x - Floor(0.5 + c * (x + y)) \\ y + Floor(0.5 + c * (x + y)) \end{pmatrix}$
6 $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \leftarrow \begin{pmatrix} x_1 + Floor(0.5 + b * (x_1 - y_1)) \\ y_1 + Floor(0.5 + b * (x_1 - y_1)) \end{pmatrix}$
7 $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \leftarrow \begin{pmatrix} x_2 + Floor(0.5 + a * y_2) \\ y_2 \end{pmatrix}$
8 **return** (x_3, y_3)

Algorithm 2: $RotCBANeg(x, y, \theta)$: Inverse Point Rotation of center $(0, 0)$ and angle θ

Input : $(x, y) \in \mathbb{Z}^2, 0 \leq \theta \leq \pi$
Output: $(x_3, y_3) \in \mathbb{Z}^2$

1 If $\theta = \frac{\pi}{3}$ Then $\begin{pmatrix} a \\ c \end{pmatrix} \leftarrow \begin{pmatrix} -1 \\ 0.25 \end{pmatrix}$
2 Else $a \leftarrow -2 + \frac{-6+4\sqrt{3}\sin\theta}{-3\cos\theta + \sqrt{3}\sin\theta}$
3 $\quad c \leftarrow -a/4$
4 $b = \frac{1}{6}(3 - 3\cos\theta + \sqrt{3}\sin\theta)$.
5 $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \leftarrow \begin{pmatrix} x - Floor(0.5 + a * y) \\ y \end{pmatrix}$
6 $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \leftarrow \begin{pmatrix} x_1 - Floor(0.5 + b * (x_1 - y_1)) \\ y_1 - Floor(0.5 + b * (x_1 - y_1)) \end{pmatrix}$
7 $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \leftarrow \begin{pmatrix} x_2 + Floor(0.5 + c * (x_2 + y_2)) \\ y_2 - Floor(0.5 + c * (x_2 + y_2)) \end{pmatrix}$
8 **return** (x_3, y_3)

between $-\pi$ and π rather than for angles between 0 and 2π.

This leads to the first two algorithms that correspond to digital cell rotations: Algorithm 1 for rotations of angles between 0 and π. The second algorithm, Algorithm 2, is exactly the same algorithm but all the operations of Algorithm 1 are performed in reverse order with negative push coefficients: $x + Floor(0.5 + a * y)$ becomes $x - Floor(0.5 + a * y)$. This ensures that the digital rotation is invertible and that the rotation of angle θ is the inverse of rotation of angle $-\theta$. Algorithm 3 combines both Algorithm 1 and Algorithm 2 to perform a digital rotation of a triangular cell grid. Finally Algorithm 4 shows the final digital algorithm for a triangular grid image.

Algorithm 3: $RotPoint(x, y, \theta)$: Rotation of center $(0, 0)$ and angle θ of point (x,y) in the triangular Grid.

Input : $(x, y) \in \mathbb{Z}^2, \theta \in \mathbb{R}^2$
Output: $(x', y') \in \mathbb{Z}^2$

1 $\theta = \theta$ Mod 2π
2 If $0 \leq \theta \leq \pi$ Then $(x', y') \leftarrow RotCBA(x, y, \theta)$
3 $\quad$ Else $(x', y') \leftarrow RotCBANeg(x, y, 2\pi - \theta)$
4 Return(x', y')

Algorithm 4: $RotTri(ImageIn, \theta)$: Bijective Rotation for the triangular Grid of center $(0, 0)$ and angle θ

Input : ImageIn, θ
Output: ImageOut

1 For all (x, y) in ImageIn
2 $\quad (x', y') \leftarrow$ RotPoint(x, y, θ)
3 $\quad$ ImageOut $(x', y') \leftarrow$ ImageIn(x, y)
4 **return** *ImageOut*

3.3. Bijectivity

All has been done in order to have a bijective transform $RotTri$. Let us show that formally by examining the mappings of Algorithm 1. Let us define the mapping $\mathcal{A}(a) : (x, y) \mapsto (x + Floor(0.5 + ay), y)$, $a \in \mathbb{R}^2$ (line 7 in Algorithm 1). For an A-row, we have $\mathcal{A}(a)(S) = S$ (y does not change). All the triangular cells on a given A-row are *pushed* in direction A by the same integer number of grid points. Since A-rows partition the grid points, it proves that mapping $\mathcal{A}(a)$ is bijective over the grid.

Let us now consider the mapping $\mathcal{B}(b) : (x, y) \mapsto (x + Floor(0.5 + b(x - y)), y + Floor(0.5 + b(x - y)))$, $b \in \mathbb{R}^2$ (line 6 in Algorithm 1). Let us show that for a B-row S, we have $\mathcal{B}(b)(S) = S$ (i.e. that $x - y$ does not change). Let us consider a point

$(x, y) \in S$ and $(x', y') = \mathcal{B}(b)(x, y)$. We have $x' - y' = x - y + Floor(0.5 + b(x - y)) - Floor(0.5 + b(x - y)) = x - y$. All the triangular cells on a given B-row are *pushed* in direction B by the same integer number of grid points. Since B-rows partition the grid points, it proves that mapping $\mathcal{B}(b)$ is bijective over the grid.

The proof for C-rows is similar. All this is, of course, verified as well for Algorithm 2.

This proves that the transform presented in Algorithm 4 is bijective (as sequence of bijective mappings). By design, we have made sure that the transform is also easily invertible such that, for an image *ImageIn* in the triangular grid, we have $RotTri(ImageIn, \theta)^{-1} = RotTri(ImageIn, -\theta)$. Figure 2 shows how each matrix acts on the triangular grid. In order to illustrate the action of each shear transform, we set $a = b = c = 1$. The rows containing the rotation center $(0, 0)$ do not move. Note that mapping $\mathcal{B}$ and mapping C conserve the type (even or odd) of the triangles which is not the case for mapping $\mathcal{A}$ that changes the types of the triangles depending if $Floor(0.5 + ay)$ is odd or even.

4. Results

In this section, we present some results and error measurements. The algorithms have all been implemented in Mathematica.

4.1. Error Measure

In order to evaluate the quality of the rotation, two distance criterii Andres (1996, 2018) were considered: the Maximum Distance criteria (MD) and the Average Distance criteria (AD). We measure the average and the maximal distance between the digital points obtained after the digital rotation and the Euclidean points obtained after the corresponding continuous rotation. The goal is to measure the error committed by considering the digital point instead of the continuous point. Let us recall here that the triangular cells have sides of size 1 and two neighboring triangles centroids are at a distance of 0.57.

For the first error measure, we simply considered the average and maximal distance criterii for our triangular grid. We obtained an average distance bounded by 0.52 (mean average distance over all the angles is 0.40). The maximal distance is bounded by 1.47 (mean maximal distance over all the angles of about 0.93). In order to be consistent with a more classical way of representing a triangular grid, we recomputed our error measures with the centroids rather than our corrected grid points. The measure becomes therefore: at what average and maximal distance is the continuously rotated point from the centroid of the digitally rotated corresponding triangle (see Figure 5). Let us call $Rot_\theta(x, y)$ the continuous rotation transform of center $(0, 0)$ and angle θ for point (x, y).

For a point (x, y) of the triangular grid:

- Continuous rotation: $(x, y) \mapsto (x', y') = Tri2Cart(x, y) \pm \left(0, \frac{\sqrt{3}}{12}\right) \mapsto (x'', y'') = Rot_\theta(x', y')$.
- Digital rotation: $(x, y) \mapsto (x_1, y_1) = RotPoint(x, y, \theta) \mapsto (x_2, y_2) = Tri2Cart(x_1, y_1) \pm \left(0, \frac{\sqrt{3}}{12}\right)$

The error measure is therefore defined as : for all $(x, y) \in \mathbb{Z}^2$: average and maximal distance value between the points (x'', y'') and (x_2, y_2). In order to compute the final error (Figure 5), we considered a triangular grid of size 1000^2 for angles from $-\pi$ to π with steps of $k\pi/600$. Figure 5 shows an average error of about 0.5 (maximum average error is 0.56 near. Mean average error over all angles is around 0.42). The maximal distance is bounded by 1.6.

One last question was to verify that there is no rotation center problem since the rotation center for the Euclidean rotation is Euclidean point $(0, 0)$ while the rotation center used for our triangular grid rotation is the one with triangular grid coordinates $(0, 0)$ and therefore with Euclidean coordinates $\left(0, \frac{\sqrt{3}}{12}\right)$. There is however no reason for rotation center error since the error between both ways of considering the point representing a cell creates an additional small error localized around a cell. To validate these values and ensure that there is no drift when considering triangles at a long distance, we took 10^6 randomly chosen points at a distance of 10^9 from the center and obtained similar results than the one presented in Figure 5.

4.2. Examples

Lastly, Figure 6 shows the result of the bijective rotation on a 'Lena' image of size 64x64 for different angles. The holes that appear on the boundary of the rotated image is not a problem of bijectvity but simply because the

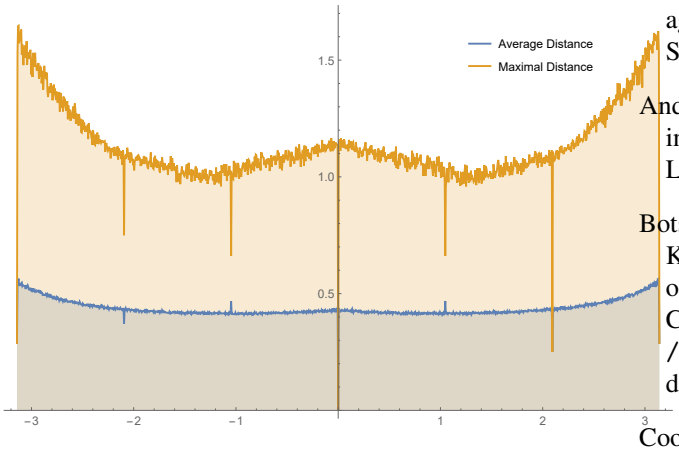

Figure 5: Average (AD) and Maximal (MD) Distance error criteria for the triangular grid Rotation Algorithm for angles 0 to 2π

rotation does not guarantee that a straight line is transformed into a straight line.

5. Conclusions

In this paper, we have presented a method to build bijective digital rotations on a triangular grid using three shear transforms. The main advantages of the method is that it works for all angles, is one to one, easy to implement and provides a good approximation of the continuous rotation. In order to evaluate the error committed using such a digital rotation when compared to a continuous rotation, the average distance, on an image, between the point rotated by continuous rotation and the one rotated by digital one has been computed. Experiments showed that this average distance is not too high and oscillates around 0.5.

Acknowledgments

The authors gratefully acknowledge the help of Ranita Biswas, Assistant-Professor from IIT Roorkee, India.

References

Andres, E., 1996. The quasi-shear rotation, in: Int. Workshop on Discrete Geometry for Computer Imagery 1996, Lyon (France), Lecture Notes in Computer Science (Springer), vol. 1176, pp. 307–314.

Andres, E., 2018. Shear based bijective digital rotation in hexagonal grids. Submitted to Pattern Recognition Letters .

Botsch, M., Pauly, M., Rossl, C., Bischoff, S., Kobbelt, L., 2006. Geometric modeling based on triangle meshes, in: ACM SIGGRAPH 2006 Courses, ACM, New York, NY, USA. URL: `http://doi.acm.org/10.1145/1185657.1185839`, doi:`10.1145/1185657.1185839`.

Cooper, O., Campbell, N.W., Gibson, D.P., 2005. Automatic augmentation and meshing of sparse 3d scene structure. 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1 1, 287–293.

Dutt, M., Andres, E., Largeteau-Skapin, G., 2018. Characterization and generation of straight line segments on triangular cell grid. Pattern Recognition Letters 103, 68–74.

Freeman, H., 1979. Algorithm for generating a digital straight line on a triangular grid. IEEE Transactions on Computers C-28, 150–152.

Jacob, M., Andres, E., 1995. On discrete rotations, in: Int. Workshop on Discrete Geometry for Computer Imagery 1995, Clermont-Ferrand (France), pp. 161–174.

Koster, A., Muoz, X., 2009. AGraphs and Algorithms in Communication Networks: Studies in Broadband, Optical, Wireless and Ad Hoc Networks. Springer.

Nagy, B., 2003a. A family of triangular grids in digital geometry, in: Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, ISPA 2003, Rome, Italy. pp. 101–106.

Nagy, B., 2003b. Shortest path in triangular grids with neighbourhood sequences. Journal of Computing and Information Technology 11, 111–122.

Nagy, B., 2004. Generalized triangular grids in digital geometry. Acta Mathematica Academiae Paedagogicae Nyíregyháziensis. 20, 63–78.

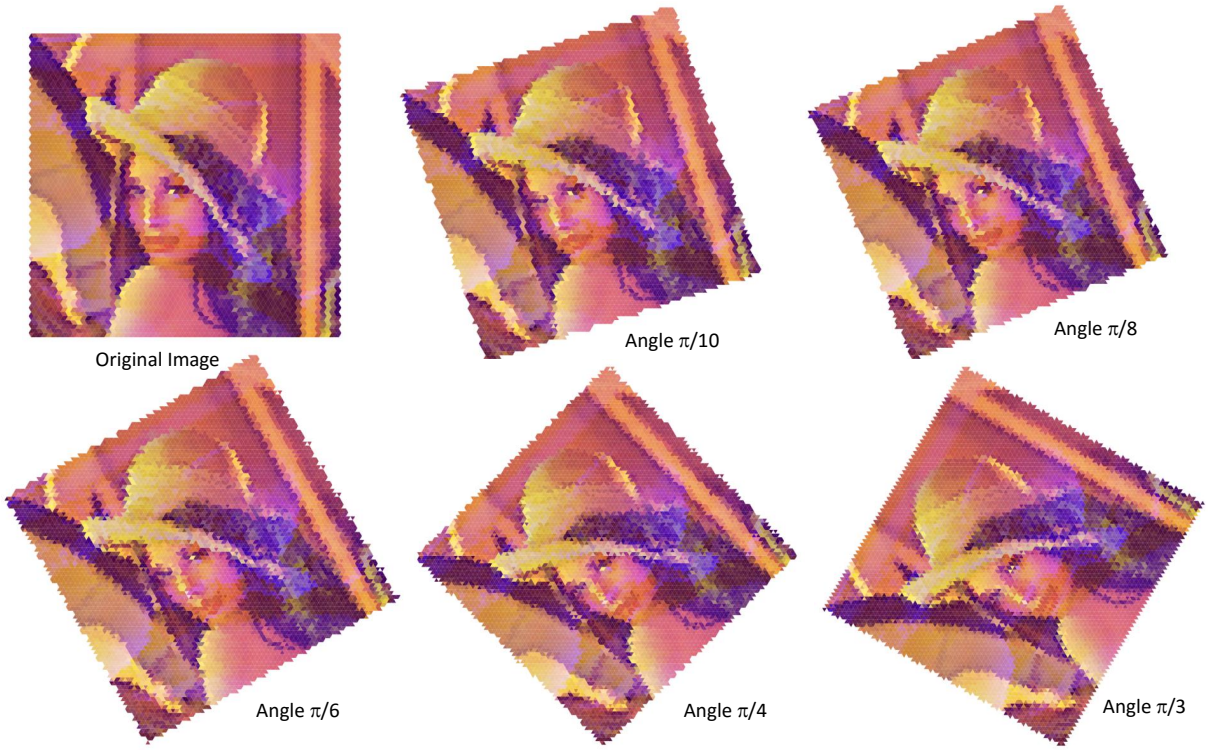

Figure 6: A lena 64x64 image on a triangular grid rotated at various angles.

Nagy, B., 2012. Cellular topology on the triangular grid, in: Combinatorial Image Analaysis. IWCIA 2012, Lecture Notes in Computer Science, vol. 7655 (Springer). pp. 143–153.

Nagy, B., 2015. Cellular topology and topological coordinate systems on the hexagonal and on the triangular grids 75, 117–134.

Nagy, B., Moisi, E.V., 2017. Memetic algorithms for reconstruction of binary images on triangular grids with 3 and 6 projections. Applied Soft Computing 52, 549–565.

Nouvel, B., Rmila, E., 2005. Characterization of bijective discretized rotations, in: Combinatorial Image Analysis, Lecture Notes in Computer Science, pp. 248–259.

Ogawa, H., 1986. Labeled point pattern matching by delaunay triangulation and maximal cliques. Pattern Recognition 19, 35–40.

Paeth, A., 1986. A fast algorithm for general raster rotation, in: Graphic Interface 86 (reprinted with corrections in Graphic Gems (Glassner Ed.) Academic 1990, pp.179-195), pp. 77–81.

Pluta, K., Roussillon, T., Cœurjolly, D., Romon, P., Kenmochi, Y., Ostromoukhov, V., 2018. Characterization of bijective digitized rotations on the hexagonal grid. Journal of Mathematical Imaging and Vision URL: https://doi.org/10.1007/s10851-018-0785-1, doi:10.1007/s10851-018-0785-1.

Reveillès, J.P., 1991. Calcul en Nombres Entiers et Algorithmique. Ph.D. thesis. Université Louis Pasteur, Strasbourg, France.

Roussillon, T., Coeurjolly, D., 2016. Characterization of bijective discretized rotations by Gaussian integers. Research Report. LIRIS UMR CNRS 5205. URL: https://hal.archives-ouvertes.fr/hal-01259826.

Sun, Y., Page, D.L., Paik, J.K., Koschan, A., Abidi, M.A., 2002. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing, in: IEEE International Conference on Image Processing, pp. 825–828.

Tanaka, A., Kameyama, M., Kazama, S., Watanabe, O., 1986. A rotation method for raster image using skew transformation, in: Proc. IEEE Conf,. Comput. Vision and Pattern Rec., pp. 272–277.