



HAL
open science

Shear based Bijective Digital Rotation in Hexagonal Grids

Éric Andrès

► **To cite this version:**

| Éric Andrès. Shear based Bijective Digital Rotation in Hexagonal Grids. 2018. hal-01900148

HAL Id: hal-01900148

<https://hal.science/hal-01900148>

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 Hexagonal Grids

Eric Andres^a

^aUniversity of Poitiers, Laboratory XLIM, ASALI, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France

Abstract

In this paper, a new bijective digital rotation algorithm for the hexagonal grid is proposed. The method is based on an original decomposition of rotations into shear transforms. It works for any angle with an hexagonal centroid as rotation center and is easily invertible. The algorithm achieves an average distance between the digital rotated point and the continuous rotated point of about 0.42 (for 1.0 the distance between two neighboring hexagons).

Keywords: Hexagonal grid, Digital Rotation, Bijective Digital Rotation, Shear Transform

1. Introduction

Applying a continuous geometric transform $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ to a 2D regular grid (defined by two vectors and an origin) is not trivial, although it is a fundamental problem in imaging. A transformed grid point through a continuous transform is usually not a grid point. One of the ways to deal with this, is to add a digitization transform D , classically the nearest grid point. The *digitized transform* $D \circ T$, has however no reasons, in general, to be injective nor surjective, even when the original continuous transform T is bijective. This leads to information loss. A *digitized rotation* may cause up to 17% information loss. For this reason, in imaging softwares, interpolation and sometimes oversampling are used to perform digital rotations (Danielsson and Hammerin (1992), Sterling and Sterling (1998)). This is however not always possible when the image has low color resolution or when grid points carry information that are not meant to be interpolated. That is where bijective digital rotations are useful. In our case, we are interested in *bijective digital rotations on an hexagonal grid*.

Hexagonal grids have enjoyed renewed interest these last couple of years (Middleton and Sivaswamy (2005))

with, in particular applications in 3D printing filling with honeycomb structures (Compton and Lewis (2014), Gibson et al. (2014)), networks (Kayaturan and Vernitski (2016), Stojmenovic (1995)), image processing (Grigoryan and Agaian (2016), Karkishchenko and Mnukhin (2017), Mostafa and Her (2016)), computer vision (He and Jia (2005), Veni and Narayanankutty (2014)), etc. Our motivation for lossless rotations in hexagonal grids came from recent work by K. Pluta and al. (Pluta et al. (2017, 2018)) and the perspectives it opened for 3D bijective rotations of honeycomb structures for 3D printing.

Transformations in hexagonal grids are still a largely open problem with few references (see Her (1995) for a discussion on transforms on hexagonal grids). Recently K. Pluta et al. showed that, as for the square grid (Jacob and Andres (1995), Nouvel and Remila (2005)), there are angles for which the digitized rotation is bijective (Pluta et al. (2017, 2018)) in the hexagonal grid. The problem is that these angles represent only a subset of all angles (Pluta et al. (2017, 2018)). This is the limitation we propose to overcome in this paper.

In order to propose a bijective rotation in the hexagonal grid that works for all angles, we propose to revisit an idea the author of the present paper has already used for bijective rigid motions in the classical square grid: decomposing a rotation into a sequence of shear transforms (Andres (1996) developed out of an original idea by Reveillès

^{*}Corresponding author

Email address: eric.andres@univ-poitiers.fr (Eric Andres)

(1991)). A.W. Paeth used such a decomposition in order to propose a fast anti-aliasing method for image rotations (Paeth (1986)). The idea proposed by J-P. Reveilles is to use the same decomposition to push rows of pixels by an integer number of pixels, ensuring bijectivity and trivial reversibility. In this paper we propose to consider three directions (the three symmetry axes of an hexagon) for the shear transforms. With three such shear transforms in the hexagonal coordinate system, one for each hexagonal symmetry direction, a novel rotation decomposition is proposed. The idea behind the shear transforms is to push whole rows of hexagons successively in those three directions by integer numbers of hexagons, ensuring bijectivity. As for the shear based rotation proposed by A.W. Paeth, we have angles where the formula diverges. There is however a very simple way to overcome this problem. This leads to a digital rotation algorithm for all angles in the hexagonal grid, with the limitation that the center is a grid point (the centroid of an hexagon). A future work could consist in lifting this limitation and proposing a rigid motion transform (rotation with an arbitrary center).

The organization of the paper is as follows: in section two, we present the preliminaries. In particular, we present shear transforms and how shear transforms have been used in the classical square grid to define digital rotations. An error criteria based on the distance between the continuous and the digital rotated points are presented. In section three, we introduce our method of digital bijective rotation for the hexagonal grids. We conclude and present perspectives in the last section.

2. Preliminaries

2.1. Hexagonal grid

We are considering an hexagonal cell (centered on the grid point) with "pointy top" hexagons (two sides of the hexagons are parallel to the ordinate axis of the classical Euclidean coordinate system). It should be easy to transpose this work for other hexagon orientations or hexagon sizes. The hexagons are regular with a side length of $\frac{1}{\sqrt{3}}$. This means that between the centroids of two neighboring hexagons, there is a distance of 1. There are various ways of creating a coordinate system for hexagonal grids. We chose a simple 2D coordinate system where the first coordinate, the hexagonal hx -axis, is given by the Cartesian

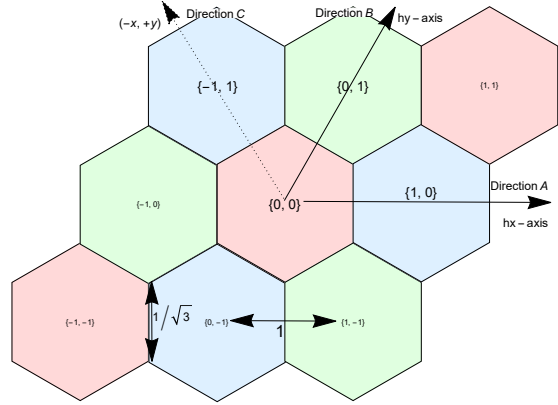


Figure 1: coordinate system, hexagon size and A, B and C directions

vector $(1, 0)$ and the second coordinate, the hexagonal hy -axis, is defined by the Cartesian vector $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$ (Snyder et al. (1999)). The coordinate transforms from hexagonal grid to the Cartesian grid and vice versa are given by the following transforms (Snyder et al. (1999)):

$$Cart2Hex : (x, y) \mapsto (x_h, y_h) = \left(x - \frac{y}{\sqrt{3}}, \frac{2y}{\sqrt{3}}\right)$$

$$Hex2Cart : (x_h, y_h) \mapsto (x, y) = \left(x_h + \frac{y_h}{2}, \frac{y_h \sqrt{3}}{2}\right)$$

We define three directions: direction A defined by vector $(1, 0)$, direction B defined by vector $(0, 1)$ and direction C defined by vector $(-1, 1)$. These vectors are given in the hexagonal coordinate system (see Figure 1). A A-row is a set of hexagonal grid points with hexagonal coordinates $(x, y) \in \mathbb{Z}^2$ such that they have all the same y value. All the hexagons of an A-row are aligned in the A-direction and are generated by hexagonal coordinate vector $(1, 0)$ and one of its points. A B-row is a set of hexagonal grid points with hexagonal coordinates $(x, y) \in \mathbb{Z}^2$ such that they have all the same x value. All the hexagons of a B-row are aligned in the B-direction and are generated by hexagonal coordinates vector $(0, 1)$ and one of its points. A C-row is a set of hexagonal grid points with hexagonal coordinates $(x, y) \in \mathbb{Z}^2$ such that they have all the same $x + y$ value. All the hexagons of a C-row are aligned in the

C-direction and are generated by hexagonal coordinates vector $(-1, 1)$ and one of its points.

2.2. Shear Transforms

The idea behind the method proposed in this paper is to decompose a rotation transform into a sequence of shear transforms in the hexagonal coordinate system. A *shear transform* is a linear mapping that translates a point in a given direction by an vector proportional (by a shear factor) to the signed distance to a line parallel to that direction. A typical shear transform uses the axis-lines as shear lines. Shear transforms preserve areas. This is why it is quite natural to decompose isometries into sequences of shear transforms (as atomic 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 A.W. Paeth (Paeth (1986)):

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

This rotation decomposition leads to a fast and simple antialiased rotation that is still in use in some image libraries such as ImageMagick.

2.3. Bijective Digital Rotations in the square grid

In this paper we are interested in bijective digital rotations on hexagonal grids. Let us first see how this problem has been tackled in the classical square grid. Let us consider a digitized rotation $DR(\alpha)$:

$$DR(\theta) : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \lfloor x \cos \theta + y \sin \theta + 0.5 \rfloor \\ \lfloor -x \sin \theta + y \cos \theta + 0.5 \rfloor \end{pmatrix}$$

with $\lfloor u \rfloor$ the biggest integer smaller or equal to u (i.e. floor function).

A digitized rotation is not, in general, surjective or injective, except for some angles where the transform is actually bijective. For the interested reader, please refer to Jacob and Andres (1995), Nouvel and Remila (2005), Roussillon and Coeurjolly (2016). The problem with the angles for which the digitized rotation is bijective is that they do not cover all angles. J.P. Reveilles took the same decomposition than A.W. Paeth to propose a bijective digital rotation that works for all angles (Reveillès (1991)).

E. Andres improved on this idea and gave formulas for an improved bijective digital rotation and a bijective rigid motion in the classical square grid (Andres (1996)). The idea that leads to a bijective digital rotation is the following: the shear coefficients multiplied by x and y respectively, $-x \tan \frac{\theta}{2}$ and $y \sin \theta$, are approximated by their closest integer $\lfloor -x \tan \frac{\theta}{2} + 0.5 \rfloor$ and $\lfloor y \sin \theta + 0.5 \rfloor$. Moving a grid point in the x -axis direction and y -axis direction by an integer displacement is obviously a reversible operation. One could think that this leads to a very coarse approximation of the continuous rotation, but this is not the case. The way to measure this "approximation" has lead E. Andres to propose an error measure for digital bijective rotations (Andres (1992, 1996)) which we'll present in the next subsection.

2.4. Error Measure

Each grid point has one and only one image through a bijective digital rotation but that does not mean that the digital rotation is a good approximation of the continuous one. To measure how "wrong" we are by choosing the digital rotation over the continuous one, we are considering two distance criterii (Andres (1992, 1996)). Let us first denote $\mathcal{G} = \{ax_g + by_g | (a, b) \in \mathbb{Z}^2\} \subset \mathbb{R}^2$ a grid defined by the point $(0, 0)$ and two vectors x_g and y_g . Let us denote $R_\theta(p)$ the continuous rotation of center $(0, 0)$ and angle θ of a grid point $p \in \mathcal{G}$ and $\mathcal{R}_\theta(p)$ its digital rotation of center $(0, 0)$ ($\mathcal{R}_\theta(p)$ is a grid point in this case).

The *Maximum Distance error criteria (MD)* consists in computing $\max_{p \in \mathcal{G}} (d(R_\theta(p), \mathcal{R}_\theta(p)))$. The *Average Distance error criteria (AD)* consists in computing $avg_{p \in \mathcal{G}} (d(R_\theta(p), \mathcal{R}_\theta(p)))$, where $avg_{p \in \mathcal{G}}$ is the average distance over the grid.

3. Bijective Digital Rotation in the Hexagonal Grid

3.1. Rotation decomposition into shears in the hexagonal grid

K. Pluta showed that there exists a subset of angles for which the digitized rotation of an hexagonal grid is bijective (Pluta et al. (2017, 2018)). Our idea is to use shear transforms to define a digital rotation in the hexagonal grid that works for all angles by *pushing* grid points into specific directions. Since we are working in an hexagonal grid, we have three privileged directions to push grid

points: direction A defined by vector $(1, 0)$, direction B defined by vector $(0, 1)$ and direction C defined by vector $(-1, 1)$ (vectors expressed in the hexagonal coordinate system. See Figure 1). The shear transforms for the three directions, expressed in the hexagonal grid coordinate system, correspond to the following matrices:

$$\begin{aligned} \text{mat}A_{\text{hex}} &= \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}; \text{mat}B_{\text{hex}} = \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \\ \text{mat}C_{\text{hex}} &= \begin{pmatrix} 1-c & -c \\ c & 1+c \end{pmatrix} \end{aligned}$$

$\text{mat}A_{\text{hex}}$ and $\text{mat}B_{\text{hex}}$ are classical shear transforms with the axis as shear lines. For the third matrix, it is the same idea: We have a direction C defined by the vector $(-1, 1)$ with an orthogonal direction $(1, 1)$ (in the hexagonal coordinate system). This leads to matrix $\text{mat}C_{\text{hex}}$.

Since the rotation matrix is simplest expressed in the classical Cartesian coordinate system, we switch back to the Cartesian coordinate system, which leads to:

$$\begin{aligned} \text{mat}A_{\text{cart}} &= \begin{pmatrix} 1 & \frac{2a}{\sqrt{3}} \\ 0 & 1 \end{pmatrix} \\ \text{mat}B_{\text{cart}} &= \begin{pmatrix} 1 + \frac{b}{2} & -\frac{b}{2\sqrt{3}} \\ \frac{b\sqrt{3}}{2} & 1 - \frac{b}{2} \end{pmatrix} \\ \text{mat}C_{\text{cart}} &= \begin{pmatrix} 1 - \frac{c}{2} & -\frac{c}{2\sqrt{3}} \\ \frac{c\sqrt{3}}{2} & 1 + \frac{c}{2} \end{pmatrix} \end{aligned}$$

Each of the matrices $\text{mat}X_{\text{cart}}$ corresponds to $\text{mat}X_{\text{hex}}$ expressed in the classical Cartesian coordinate system. The idea of our method is to push grid points in those three directions in order to approximate a rotation. Therefore, we solved the following equation:

$$\text{mat}A_{\text{cart}} \cdot \text{mat}B_{\text{cart}} \cdot \text{mat}C_{\text{cart}} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

which has a unique solution for a, b, c :

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -1 + \frac{\sqrt{3}-2\sin\theta}{\sqrt{3}\cos\theta-\sin\theta} \\ 1 - \cos\theta + \frac{\sin\theta}{\sqrt{3}} \\ 1 - \frac{\sqrt{3}-2\sin\theta}{\sqrt{3}\cos\theta-\sin\theta} \end{pmatrix}$$

It is easy to see that $a = -c$. Let us note that from here on, we are only going to work in the hexagonal coordinate system with matrices $\text{mat}A_{\text{hex}}$, $\text{mat}B_{\text{hex}}$ and $\text{mat}C_{\text{hex}}$.

Algorithm 1: $\text{RotCBA}(x, y, \theta)$: POINT ROT. CBA OF CENTER $(0, 0)$ AND ANGLE θ

Input : $(x, y) \in \mathbb{Z}^2, 0 \leq \theta < 2\pi/3$
Output: $(x_3, y_3) \in \mathbb{Z}^2$

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

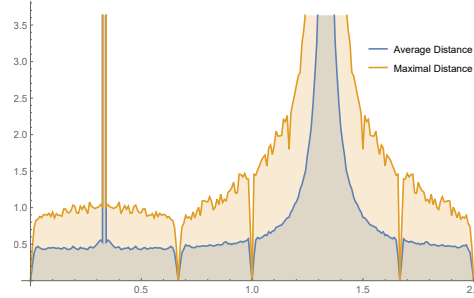


Figure 2: Average (AD) and Maximal (MD) Distance error criteria for Rotation CBA for angles 0 to 2π

3.2. Dealing with the divergence and the inverse

The proposed solution for (a, b, c) is not universal: the denominator for a (and c), $\sqrt{3}\cos\theta - \sin\theta$, is equal to 0 for θ equal to $\pi/3$ or $4\pi/3$ (all angles are considered to be between 0 and 2π). As one can see in Figure 2, the zeros for $\pi/3$ and $4\pi/3$ are not of the same nature: $\lim_{\theta \rightarrow \pi/3} a = 0.5$ and $\lim_{\theta \rightarrow 4\pi/3} a = \pm\infty$. Angle $\pi/3$ is a point singularity that can be easily dealt with since it is trivial to rotate a hexagonal image by such an angle bijectively: $R_{\pi/3}(x, y) = (-y, x + y)$ is still a hexagonal grid point if (x, y) is one. This leads to the digital Rotation RotCBA (Algorithm 1) that rotates an hexagonal grid point for angles from 0 to $2\pi/3$.

Algorithm 2: $RotCBANeg(x, y, \theta)$: INVERSE ROT. CBA OF CENTER $(0, 0)$ AND ANGLE θ

Input : $(x, y) \in \mathbb{Z}^2, 0 \leq \theta < 2\pi/3$
Output: $(x_3, y_3) \in \mathbb{Z}^2$

- 1 If $\theta = \frac{\pi}{3}$ Then $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \leftarrow \begin{pmatrix} x+y \\ -x \end{pmatrix}$ – Rotation $-\pi/3$
- 2 Else $\begin{pmatrix} a \\ b \end{pmatrix} \leftarrow \begin{pmatrix} 1 - \frac{\sqrt{3}-2\sin\theta}{\sqrt{3}\cos\theta - \frac{\sin\theta}{\sqrt{3}}} \\ -1 + \cos\theta - \frac{\sin\theta}{\sqrt{3}} \end{pmatrix}$
- 3 $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \leftarrow \begin{pmatrix} x + \lfloor y * a + 0.5 \rfloor \\ y \end{pmatrix}$
- 4 $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \leftarrow \begin{pmatrix} x_1 \\ y_1 + \lfloor x_1 * b + 0.5 \rfloor \end{pmatrix}$
- 5 $\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} \leftarrow \begin{pmatrix} \lfloor x_2 * (1+a) + y_2 * a + 0.5 \rfloor \\ \lfloor -x_2 * a + y_2 * (1-a) + 0.5 \rfloor \end{pmatrix}$
- 6 **return** (x_3, y_3)

Let us see how we can handle the problems around $4\pi/3$. In Figure 2, we can see that all the values around angle $4\pi/3$ are affected by the divergence. Actually, starting at $2\pi/3$ the average (AD) and maximal (MD) distance error values start to increase. The idea here is to consider only rotations for angles between 0 and $2\pi/3$ and map all the other angles to this interval. This is not that difficult since, in the hexagonal grid, rotations by angles $k\pi/3$ (with a grid point center) are naturally bijective so that for an angle θ , we can always decompose it as $\theta = k\pi/3 + \theta'$ where $0 \leq \theta' \leq 2\pi/3$.

There is however another problem that needs to be dealt with if we want an easy to use digital rotation algorithm: *invertibility*. Indeed, with Algorithm 1, $RotCBA(x, y, \theta)^{-1}$ is not equal to $RotCBA(x, y, -\theta)$. Having a bijective transform does not automatically mean that we have an easy way of performing the inverse transform. The approach we adopted to be sure that our rotation for angle θ is easily invertible is to perform all hexagon push operations in the reverse order for an angle $-\theta$ (or more precisely $2\pi - \theta$). For this, we introduce Algorithm 2 which is literally the inverse mapping of the mapping of algorithm 1, with shear factors $-(a, b)$ and reversed shear transforms application order. With all this, we can now propose a rotation for an hexagonal point. The rotation of a point (x, y) by an angle $0 \leq \theta < 2\pi$ is performed in the following way (see Algorithm 3):

Algorithm 3: $RotPoint(x, y, \theta)$: ROTATION OF CENTER $(0, 0)$ AND ANGLE θ OF POINT (x, y) IN THE HEXAGONAL GRID.

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

- 1 $\theta = \theta \text{ Mod } 2\pi$
- 2 If $0 \leq \theta \leq \frac{2\pi}{3}$ Then $(x'', y'') \leftarrow RotCBA(x, y, \theta)$
- 3 Elself $\frac{2\pi}{3} < \theta \leq \pi$ Then
 $(x'', y'') \leftarrow RotCBA(-x - y, x, \theta - \frac{2\pi}{3})$
- 4 Elself $\pi < \theta < \frac{4\pi}{3}$ Then
 $(x', y') \leftarrow RotCBANeg(x, y, \frac{4\pi}{3} - \theta)$;
 $(x'', y'') \leftarrow (y', -x' - y')$
- 5 Else $(x'', y'') \leftarrow RotCBANeg(x, y, 2\pi - \theta)$
- 6 **Return** (x'', y'')

1. For an angle $0 \leq \theta \leq 2\pi/3$: apply algorithm 1 to the point (x, y) with angle θ ;
2. For an angle $2\pi/3 < \theta \leq \pi$: apply algorithm 1 to the point $(-x - y, x)$ with angle $\theta - 2\pi/3$ (the point $(-x - y, x)$ is the rotation of (x, y) by $2\pi/3$);
3. For an angle $\pi < \theta < 4\pi/3$: apply algorithm 2 to the point (x, y) with angle $4\pi/3 - \theta$ and then perform a rotation of angle $-2\pi/3$ on the resulting point (x', y') which maps the point (x', y') to $(y', -x' - y')$ (This ensures that operations are exactly performed in the reverse order of point 2 (angle $2\pi/3 < \theta \leq \pi$)).
4. For an angle $4\pi/3 \leq \theta < 2\pi$: apply algorithm 2 to the point (x, y) with an angle $2\pi - \theta$. This ensures that operations are exactly performed in the reverse order of point 1 (angle $0 \leq \theta \leq 2\pi/3$).

This leads to point rotation $RotPoint$ (Algorithm 3) and to the final bijective rotation $RotHexa$ (Algorithm 4). We'll explain why Rotation Algorithm 4 is bijective in the next subsection.

Let us quickly comment Figure 3 which illustrates the distance error criterii for the bijective digital rotation algorithm. In order to have a point of comparison, let us note that the best MD and AD error values are obtained when considering digitized rotations in the hexagonal grid (that are not necessarily bijective). Except for the angles $k\pi/3$, we obtain mostly MD error values of $\frac{\sqrt{3}}{3} \approx 0.57$ and AD error values of ≈ 0.35 (which can be analytically calculated as the distance from a randomly point inside a

Algorithm 4: $RotHexa(ImageIn, \theta)$: BIJECTIVE ROTATION FOR THE HEXAGONAL GRID OF CENTER $(0, 0)$ AND ANGLE θ

Input : ImageIn, θ
Output: ImageOut

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

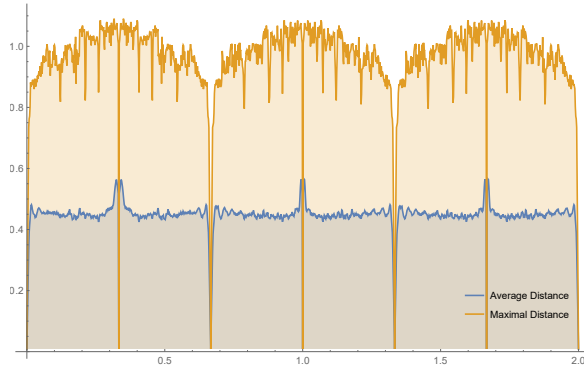


Figure 3: Average (AD) and Maximal (MD) Distance error criteria for the hexagonal grid Rotation Algorithm for angles 0 to 2π

unit hexagon to its center). As a reminder, the distance between two neighboring grid points is 1. In figure 3, we see that for our algorithm, the average distance (AD criteria) between a point rotated continuously and a point rotated discretely is very stable and close to 0.42 (compared to 0.35). The maximal distance (MD criteria) does not exceed 1.1 (compared to 0.57). This means basically that an hexagon *lands* usually in the immediate grid neighborhood of the spot where he should be. Let us note that we obtain slightly better error values than for the similar type of algorithm in the square grid (Andres (1996)).

3.3. Bijectivity

All has been done in order to have a bijective transform $RotHexa$. Let us show that formally by examining the mappings of Algorithm 1. Let us define the mapping $\mathcal{A}(a) : (x, y) \mapsto (x + \lfloor ay + 0.5 \rfloor, y)$, $a \in \mathbb{R}^2$ (line 5 in Algorithm 1). For a A -row, we have $\mathcal{A}(a)(S) = S$ (y does

not change). All hexagons on a given A -row are *pushed* in direction A by the same integer number of grid points.

In the same way, for the mapping $\mathcal{B}(b) : (x, y) \mapsto (x, y + \lfloor bx + 0.5 \rfloor)$, $b \in \mathbb{R}^2$ (line 4 in Algorithm 1), if S is a B -row, we have $\mathcal{B}(b)(S) = S$ (x does not change). All hexagons on a given B -row are pushed in direction B by the same integer number of grid points.

Let us now consider the mapping $\mathcal{C}(a) : (x, y) \mapsto (\lfloor x(1+a) + ay + 0.5 \rfloor, \lfloor -ax + y(1-a) + 0.5 \rfloor)$, $a \in \mathbb{R}^2$ (line 3 in Algorithm 1). Let us show that for a C -row S , we have $\mathcal{C}(a)(S) = S$ (i.e. that $x + y$ does not change). Let us consider a point $(x, y) \in S$ and $(x', y') = \mathcal{C}(a)(x, y)$. We have $x' + y' = \lfloor x(1+a) + ay + 0.5 \rfloor + \lfloor -ax + y(1-a) + 0.5 \rfloor = x + y + \lfloor a(x+y) + 0.5 \rfloor + \lfloor -a(x+y) + 0.5 \rfloor$. Now $\lfloor x + 0.5 \rfloor$ is the rounding function and it is easy to see that $\lfloor -u + 0.5 \rfloor = -\lfloor u + 0.5 \rfloor$ which proves that $x' + y' = x + y$ and thus that $\mathcal{C}(a)(x, y) \in S$. All hexagons on a given C -row are pushed in direction C by the same integer number of grid points. All this is of course verified as well for Algorithm 2. All the other mappings (rotations by $k\pi/3$ angles) are trivially bijective.

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 $ImageIn$ an image in the hexagonal grid, we have $RotHexa(ImageIn, \theta)^{-1} = RotHexa(ImageIn, -\theta)$. Figure 4 shows how each matrix acts on the hexagons. In order to illustrate the action of each shear transform, we set $a = b = c = 1$. The rows containing the rotation center $(0, 0)$ (marked by an ellipse) do not move. On the right of Figure 4, we show an actual rotation of the proposed image by an angle of $\pi/8$. Lastly, Figure 5 shows the result of the bijective rotation on a 'Lena' image of size 32×32 and 128×128 for different angles.

4. Conclusions

In this paper, we have proposed a bijective digital rotation method for hexagonal grids that works for all angles. For this rotation, the average distance, on an image, between a point rotated by a continuous rotation and the point rotated by the digital rotation is about 0.42 (with a distance of 1.0 between two neighboring hexagons), while the maximal distance is bounded by 1.1. This algorithm extends the direct mapping proposed by K. Pluta in the

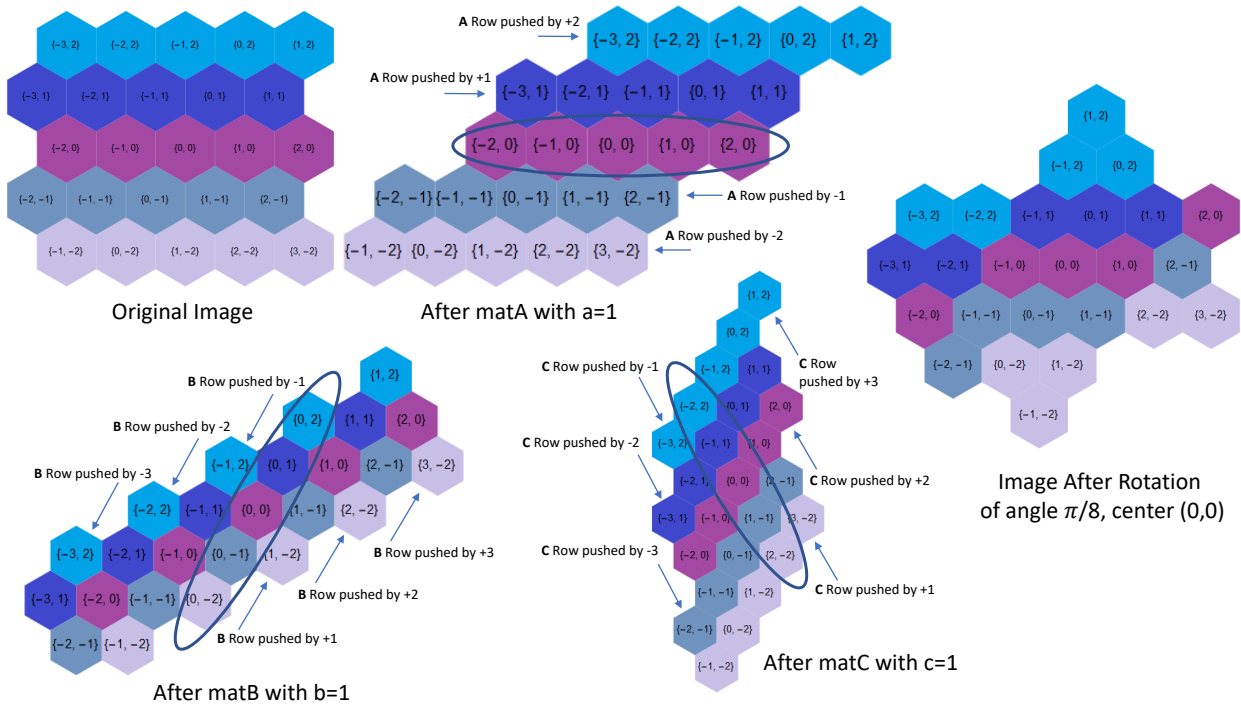


Figure 4: shears in directions A,B,C with $a=b=c=1$. The rows containing $(0,0)$ (marked by an ellipse) do not move. On the right, rotation by $\pi/8$ and center $(0,0)$.

sense that it works for all angles. However, contrary to his work, we have only an algorithm that works for hexagonal grid points as center. Extending our method to arbitrary centers would be a very interesting extension. There are many other questions that are raised by this work. Using shears for directions that are not the usual axis directions can of course be applied to the regular square grid. How does that type of rotation measure up to the previously published bijective rotations Andres (1996)? the triangular grid is a dual grid of the hexagonal grid however so may be there is something that can be done there. What about bijective rotations in arbitrary grids? One aspect that is of special for us is the extension to higher dimensions especially for honeycomb type grids with possible applications in 3D printing.

References

- Andres, E., 1992. Cercles Discrets et Rotations Discrettes. Ph.D. thesis. Université Louis Pasteur, Strasbourg, France.
- Andres, E., 1996. The quasi-shear rotation, in: International Conference on Discrete Geometry for Computer Imagery, Springer. pp. 307–314.
- Compton, B.G., Lewis, J.A., 2014. 3d-printing of lightweight cellular composites. *Advanced materials* 26, 5930–5935.
- Danielsson, P.E., Hammerin, M., 1992. High-accuracy rotation of images. *CVGIP: Graphical Models and Image Processing* 54, 340–344.
- Gibson, I., Rosen, D., Stucker, B., 2014. Additive manu-

- facturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing. Springer.
- Grigoryan, A.M., Agaian, S.S., 2016. 2d hexagonal quaternion fourier transform in color image processing, in: *Mobile Multimedia/Image Processing, Security, and Applications 2016*, International Society for Optics and Photonics. p. 98690N.
- He, X., Jia, W., 2005. Hexagonal structure for intelligent vision, in: *Information and Communication Technologies, 2005. ICICT 2005. First International Conference on*, IEEE. pp. 52–64.
- Her, I., 1995. Geometric transformations on the hexagonal grid. *IEEE Transaction on Image Processing* 4, 1213–1221.
- Jacob, M., Andres, E., 1995. On discrete rotations, in: *Int. Workshop on Discrete Geometry for Computer Imagery 1995*, Clermont-Ferrand (France), pp. 161–174.
- Karkishchenko, A., Mnukhin, V., 2017. Hexagonal images processing over finite eisenstein fields. *Procedia Engineering* 201, 287–295.
- Kayaturan, G.Ç., Vernitski, A., 2016. Routing in hexagonal computer networks: How to present paths by bloom filters without false positives, in: *Computer Science and Electronic Engineering (CEECE), 2016 8th*, IEEE. pp. 95–100.
- Middleton, L., Sivaswamy, J., 2005. *Hexagonal Image Processing: A Practical Approach*. Springer Verlag.
- Mostafa, K., Her, I., 2016. An edge detection method for hexagonal images. *International Journal of Image Processing (IJIP)* 10, 161.
- Nouvel, B., Remila, E., 2005. Characterization of bijective discretized rotations, in: *Combinatorial Image Analysis, Lecture Notes in Computer Science*, pp. 248–259.
- 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., Romon, P., Kenmochi, Y., Passat, N., 2017. Honeycomb geometry: Rigid motions on the hexagonal grid, in: Kropatsch, W.G., Artner, N.M., Janusch, I. (Eds.), *Discrete Geometry for Computer Imagery - 20th IAPR International Conference, DGCI 2017*, Vienna, Austria, September 19-21, 2017, Proceedings, Springer. pp. 33–45. URL: https://doi.org/10.1007/978-3-319-66272-5_4, doi:10.1007/978-3-319-66272-5_4.
- Pluta, K., Roussillon, T., Coeurjolly, D., Romon, P., Kenmochi, Y., Ostromoukhov, V., 2018. Characterization of bijective digitized rotations on the hexagonal grid. Submitted to *Journal of Mathematical Imaging and Vision*.
- 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>.
- Snyder, W.E., Qi, H., Sander, W.A., 1999. Coordinate system for hexagonal pixels, in: *Medical Imaging 1999: Image Processing*, International Society for Optics and Photonics. pp. 716–728.
- Sterling, I., Sterling, T., 1998. Two-Dimensional Image Rotation in: *Mathematical Visualization: Algorithms, Applications and Numerics*. Springer Berlin Heidelberg. pp. 195–206.
- Stojmenovic, I., 1995. Honeycomb network, in: *Proc. Math. Foundations of Computer Science - MFCS 95*, Lecture Notes in Computer Science, pp. 267–276.
- Veni, S., Narayanankutty, K., 2014. Vision-based hexagonal image processing using hex-gabor. *Signal, Image and Video Processing* 8, 317–326.



Figure 5: Examples with a Lena image size 32x32 and 128x128 in a hexagonal grid.