

© [2006] IEEE. Reprinted, with permission, from [Xiangjian He, Wenjing Jia, Namho Hur, Qiang Wu and Jinwoong Kim, Image Translation and Rotation on Hexagonal Structure, Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on Sept. 2006]. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it

Image Translation and Rotation on Hexagonal Structure

Xiangjian He^{1,2}, Wenjing Jia¹, Namho Hur², Qiang Wu¹, and Jinwoong Kim²

¹Computer Vision Research Group

University of Technology, Sydney

{sean, wejia, wuq}@it.uts.edu.au

²Digital Broadcasting Research Division

Electronics Telecommunications Research Institute

161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea

{namho, jwkim}@etri.re.kr

Abstract

Image translation and rotation are becoming essential operations in many application areas such as image processing, computer graphics and pattern recognition. Conventional translation moves image from pixels to pixels and conventional rotation usually comprises of computation-intensive CORDIC operations. Traditionally, images are represented on a square pixel structure. In this paper, we perform reversible and fast image translation and rotation based on a hexagonal structure. An image represented on the hexagonal structure is a collection of hexagonal pixels of equal size. The hexagonal structure provides a more flexible and efficient way to perform image translation and rotation without losing image information. As there is not yet any available hardware for capturing image and for displaying image on a hexagonal structure, we apply a newly developed virtual hexagonal structure. The virtual hexagonal structure retains image resolution during the process of image transformations, and almost does not introduce distortion. Furthermore, images can be smoothly and easily transferred between the traditional square structure and the hexagonal structure.

1. Introduction

Fast (or real-time) image transformation including image translation and rotation forms core operations in many applications such as medical image processing, computer vision, computer graphics and pattern recognition [1]. Conventional image transformation involves the mapping of pixels in the domain which is the original image coordinate system to pixels in the range which is the transformed image coordinate

system [2] based on the traditional square image structure. Rotation of an image usually performs trigonometric operations on each pixel location and is not reversible. CORDIC, developed by Volder [3] is a popular hardware efficient algorithm that can be used to compute the trigonometric operations. Although a CORDIC engine requires low hardware resources, it is iterative in nature, which can lead to intensive computations. In this paper, we perform fast, more flexible and reversible image translation and rotation based on a hexagonal image structure.

The advantages of using a hexagonal structure to represent digital images have been investigated for more than thirty years. The arrangement of hexagonal pixels on a hexagonal structure is similar to the distribution of cones on human retina [4-5]. The hexagonal image structure has features of higher degree of circular symmetry, uniform connectivity, greater angular resolution, and a reduced need of storage and computation in image processing operations [6].

In spite of its numerous advantages, hexagonal grid has so far not yet been widely applied in computer vision and graphics fields. The main problem that limits the use of hexagonal image structure is due to the lack of hardware for capturing and displaying hexagonal-based images.

In order to perform image processing based on hexagonal structures, He et al. [7] have proposed a new virtual hexagonal structure. In this paper, we will present image translation and rotation on this virtual structure.

The rest of this paper is organized as follows. In Section 2, the algorithms to convert images between the traditional square structure and the virtual hexagonal structure are introduced. Image translation is performed in Section 3 and followed by image rotation

in Section 4. Experimental results are demonstrated in Section 5. We conclude in Section 6.

2. Virtual Hexagonal Structure

In order to properly address and store hexagonal images data, Sheridan [8] proposed a one-dimensional addressing scheme for a hexagonal structure, called Spiral Architecture, as shown in Figure 1.

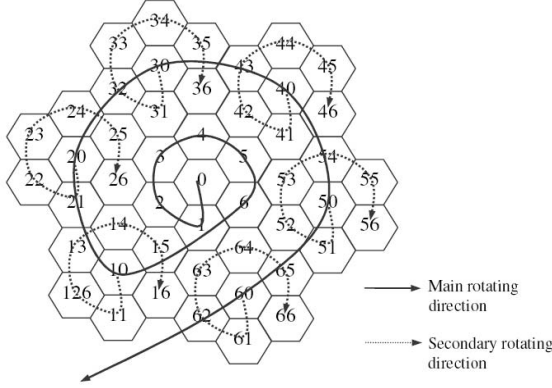


Figure 1. Spiral Architecture with spiral addressing

For the whole image, following the spiral-like curve, as shown in Figure 1, one can find out the location of any hexagonal pixel with a given spiral address starting from the central pixel of address 0. From Figure 1, it is easy to see that the location of the pixel with a given spiral address

$$a_n a_{n-1} \cdots a_1, (a_i = 0, 1, 2, \dots, 6 \text{ for } i = 1, 2, \dots, n.)$$

can be found from the locations of

$$a_i \times 10^{i-1} \text{ for } i = 1, 2, \dots, n.$$

Because there has been no hardware available for image display and capture on hexagonal structure, in this section, we introduce a software approach to the construction of virtual hexagonal structure [7]. To construct hexagonal pixels, each square pixel is first separated into 7×7 small pixels, called *sub-pixels*. To be simple, the light intensity for each of these sub-pixels is set to be the same as that of the pixel from which the sub-pixels are separated. Each virtual hexagonal pixel is formed by 56 sub-pixels as shown in Figure 2. Figure 2 shows a collection of seven hexagonal pixels constructed with spiral addresses from 0 to 6.

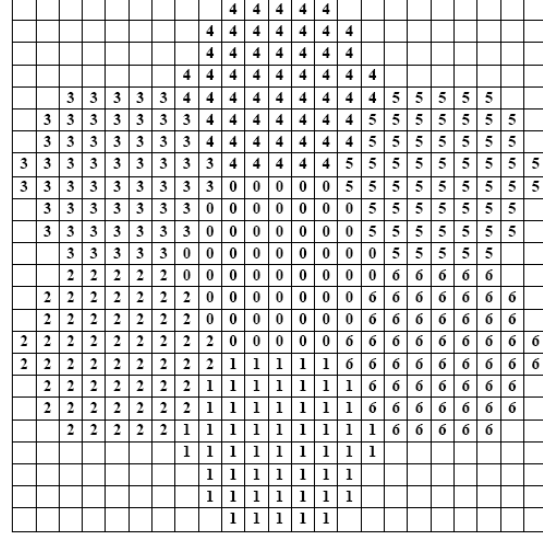


Figure 2. A cluster of seven hexagonal pixels

If we use $L(a)$ to denote the relative location of the hexagonal pixel with spiral address a to the central pixel, then the relative location of the pixel with a given spiral address

$$a_n a_{n-1} \cdots a_1, (a_i = 0, 1, 2, \dots, 6 \text{ for } i = 1, 2, \dots, n.) \quad (1)$$

can be computed by

$$L(a_n a_{n-1} \cdots a_1) = \sum_{i=1}^n L(a_i \times 10^{i-1}). \quad (2)$$

Let us assume that original images are represented on a square structure arranged as $2M$ rows and $2N$ columns, where M and N are two positive integers. $M=0$ corresponds to the first row and $N=0$ corresponds to the first column. Let the centre of the virtual hexagonal structure be located at the middle of rows M and $M+1$, and at column N . Note that there are $14M$ rows and $14N$ columns in the (virtual square) structure consisting of virtual sub-pixels obtained from the original square pixels. Let us construct the first hexagonal pixel using the 56 sub-pixels with centre located in the middle of rows $7M$ and $7M+1$ and the column $7N$ of the virtual square structure. After the 56 sub-pixels for the first hexagonal pixel are allocated, all sub-pixels for all hexagonal pixels can be assigned from (2). For our translation and rotation algorithms in this paper, the assignment of sub-pixels to corresponding hexagonal pixels is not required. We do not need to compute the intensities for the virtual hexagonal pixels either.

3. Image Translations

In [9] we have performed image translation in any direction based on movement of hexagonal pixels. Let us assume that we wish to translate an image such that the central hexagonal pixel O is moved to the hexagonal pixel with spiral address A represented in the form of (1). Then, from Equation (2), we can compute the numbers of rows and columns in the virtual square structure consisting sub-pixels to move, and the direction of the movement in order to move the central pixel from O to A . If, for example, the computed numbers of rows and columns for the movement are R and C , the translation transformation can be performed by moving all sub-pixels $|R|$ rows downwards (upwards if R is negative) and $|C|$ columns rightwards (leftwards if C is negative).



Figure 3. A 512×384 car image



Figure 4. Lossy image translation

From the statement above for a translation, we can see that if a sub-pixel is moved out from the original image area, its intensity information is not retained, and if a sub-pixel cannot be obtained by moving any sub-pixel in the given direction and for the given

distance, its intensity cannot be assigned. Figure 4 shows an image translated from Figure 3 that moves the central pixel O to a hexagonal pixel A with address 334036. This is equivalent to moving all sub-pixels downwards by 1000 sub-pixel distance and rightwards by 1120 sub-pixel distance. In Figure 4, the intensity for the pixels that cannot be obtained by this simple translation has been replaced by 255 (here we are working on a 256 grey level image) while the intensity information for the pixels that have been moved out from the original image area has gone. Therefore, the translation becomes irreversible, and the image processing requesting all original image information cannot be completed if performed based on the translated image. In order to overcome these disadvantages, we present an intensity-lossless translation in the following by replacing the sub-pixels in the white area as shown in Figure 4 with the sub-pixels that were moved out from the original image area.

1. Let us look at the remaining image block. For the image shown in Figure 4, this image block is the one sitting at right bottom corner.
2. Along the horizontal direction of the remaining block, find the missing image block. If the image has been moved rightwards (leftwards), the missing block must be on the right-hand (left-hand) side of remaining block. This missing block is then assigned to the empty area at the same level of the remaining block but on the opposite side of it. For our example shown in Figure 4, the right-hand missing block is filled into the area on the left-hand side of the remaining block.
3. Along the vertical direction, find the remaining missing blocks, which are the blocks either above or below the blocks found in Steps 1 and 2. These missing blocks are then assigned to the remaining empty area on the opposite side of the found blocks. For our example shown in Figure 4, the remaining two missing blocks (that are below the blocks found in Steps 1 and 2) are filled into the area above them.

The experimental results for the lossless translation that improve the results shown in Figure 4 will be demonstrated in Section 5.

4. Image Rotations

In [9] we have performed image rotation with an angle of multiple 60° . Figure 6 shows the rotated image of Figure 5. Similar to the work shown in [9] for image translation, the rotation operation has lost some image (intensity) information (see the white areas in Figure 6) and hence is not reversible. In this section, we will present lossless image rotation of multiple 30° .



Figure 5. A sample 384x384 image, “building”



Figure 6. The “building” in Figure 5 is rotated by 60 degrees in clockwise direction

In this paper, we show only the lossless rotation of a $2M \times 2M$ image by 30° in counterclockwise direction

and 60° in clockwise direction. The lossless rotation in other multiple 30 degrees can be accomplished in exactly the same way.

As shown in Figure 7, let us draw 6 lines from the central hexagonal pixel along the directions to hexagonal pixels with spiral addresses 1, 2, 3, 4, 5 and 6 respectively. Hexagonal pixels in any region, formed by two adjacent lines of the 6 lines obtained above, are arranged as multiple layers from the central pixel onwards. The first layer of each region is formed by a single pixel that is closest to the central pixel (but not the central pixel). The second layer consists of two pixels next to the first layer. Every upper layer has one more pixel included. All pixels at the same layer are on the same line.

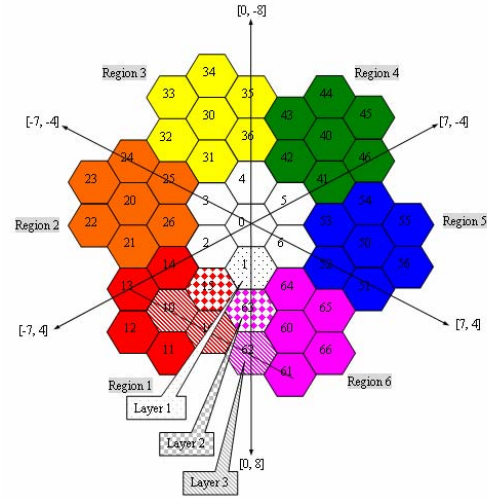


Figure 7. Regions, layers and positions of hexagonal pixels

Now, in order to perform image rotation, for any sub-pixel location, we need to know the region it belongs to. We need also know which layer it is on, its position at the layer, and its relative location to the center of the hexagonal pixel that it lies on. After the region, layer, position and relative location of any given sub-pixel are computed, the algorithm for rotation by 60° in clockwise direction can be followed. The idea is to replace any sub-pixel in a region r at layer s located at position t with relative location l by the sub-pixel in the previous region $r-1$ at the same layer s located at the same position t with the same relative location l . The lossy rotation is as shown in Figure 6. Similar to image translation, the lossless rotation by 60° is accomplished by replacing the sub-pixels in the white areas as shown in Figure 6 with the sub-pixels that were rotated out from the original image area. The detail is described as follows.

For every sub-pixel (reference sub-pixel) in the white areas compute its values of r , s , t and l . If the sub-pixel with the values of $r+l$, s , t and l (that is the sub-pixel in the next region at the same layer located at the same position with the same relative location) does not fall onto the original image area ($14M \times 14M$ area of virtual image consisting of sub-pixels), then intensity value of this reference sub-pixel is assigned to be the intensity of the original image at this sub-pixel. Otherwise, the intensity value of the reference sub-pixel is set to be the intensity of the sub-pixel in region $r+1$ (the next region) at layer s located at position t with the relative location l . The experimental results of lossless rotation of 60° clockwise will be displayed in Section 5.

The algorithm for lossless rotation of 30° counterclockwise is detailed as follows.

1. As the original image is represented on a square structure, we can easily perform a rotation of 90° counterclockwise through quickly swapping the sub-pixels in rows with those in columns without losing any image information and resolution.
2. Rotate the rotated image obtained in the previous step by 60° clockwise as described above. The original image is thus rotated by 30° counterclockwise. The image information is not lost during the rotation process.

The experimental results of lossless rotation of 30° counterclockwise will also be demonstrated in Section 5.

5. Experimental Results

The lossless image translation and rotation on the newly designed virtual hexagonal structure are implemented using C++ programming language and tested on a desktop of Pentium IV, 2.8GHz CPU and 480MB memory. Experimental results of the image transformations on two grey-level images are presented here.

For any given destination spiral address, the image can be translated fast to the target address. A sample car image, with size of 512×384 is shown in Figure 3. An example of lossless image translation on the virtual hexagonal structure is shown in Figure 8. In Figure 8 and also in Figure 4, the car image is translated to a new location while the central hexagonal pixel is moved to the location at the spiral address of 334036. It is easily computed that the transformation shifts the image by 1000 sub-pixels downwards and 1120 sub-pixels rightwards. All original image information is

retained. Furthermore, the operation can be completed very fast. The total time to complete the translation of an image with size of 512×384 is close to 0 second.



Figure 8. A lossless translation of the car image



Figure 9. The “building” in Figure 5 is rotated by 30 degrees in counterclockwise direction

Figure 6 and Figure 9 shows the results of lossy rotations of the 384×384 image “building” by 60 degrees clockwise and 30 degrees counterclockwise respectively.

The corresponding results for lossless rotations are shown in Figure 10 and Figure 11. Again, there is almost no loss of image resolution and image information in the rotated images, and no distortion introduced. Also, the processing speed is very fast. The total time to complete the rotation computation for an image with size of 384×384 is within 1 second. When more powerful computers are used and the algorithms

are optimized, the computation speed can be further improved and real-time rotation can be achieved.



Figure 10. The lossless rotation of 60 degrees clockwise of the “Building”



Figure 11. The lossless rotation of 30 degrees counterclockwise of the “Building”

6. Conclusions

In this paper, we have introduced a newly developed virtual hexagonal structure. Based on the virtual hexagonal structure, algorithms for lossless image translation and rotation have been presented. As we do not compute the light intensities for virtual hexagons, image resolution is maintained during the transformations, and we save the processing time and memory storage.

The two transformations performed in this paper provide a more flexible, reversible and fast tool for image translation and rotation. Using our virtual structure, we can not only translate images from pixels to pixels as in square structure, but also translate images from sub-pixels to sub-pixels showing potential for more accurate image matching. Furthermore, in addition to the ability to perform image rotation with 90° as in square structure, we have shown efficient ways to rotate images by multiple 60° s and 30° s.

This virtual hexagonal structure almost does not change the image resolution and introduce image distortion. It retains the advantages of the real hexagonal system such as higher degree of symmetry, uniformly connected and closed-packed form. As there are simple non-overlapping mappings between the sub-pixels and the square pixels, and the mappings between the sub-pixels and the hexagonal pixels, the results of image processing on the hexagonal structure can be easily mapped back to the square structure for display.

References

- [1] S. Suchitra, S. K. Lam and T. Srikanthan, *High-throughput rotation using sign-prediction based redundant CORDIC algorithm*, Proceedings of the International Conference on Image Processing (ICIP04), 2004, pp.2833-2836.
- [2] S. Srinivasan, *Reversible image rotations with modulo transforms*, Proceedings of the International Conference on Image Processing (ICIP05), 2005, pp.II-137-140.
- [3] J. E. Volder, *The CORDIC trigonometric computing techniques*, JRE Trans. Electron. Computing, Vol.EC-8, pp.330-334, 1959.
- [4] R. Staunton, *The design of hexagonal sampling structures for image digitization and their use with local operators*. Image and Vision Computing, 1989. 7(3): p. 162-166.
- [5] X. He, *2-D Object Recognition with Spiral Architecture*. 1999, PhD Thesis, University of Technology, Sydney.
- [6] X. He and W. Jia, *Hexagonal structure for intelligent vision*, Proceedings of the International Conference on Information and Communication Technologies (IEEE), 2005, pp.52-64.
- [7] X. He, T. Hintz, Q. Wu, H. Wang and W. Jia, *A new simulation of Spiral Architecture*, Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, June 2006, to appear.
- [8] R.M. Mersereau, *The processing of Hexagonally Sampled Two-Dimensional Signals*. Proceedings of the IEEE, 1979. 67: p. 930-949.
- [9] X. He, W. Jia, Q. Wu, N. Hur, T. Hintz, H. Wang and J. Kim, *Basic transformations on virtual hexagonal structure*, Proceedings of the 3rd International Conference on Computer Graphics, Imaging and Visualization (CGIV06), 2006, to appear.