Some efficient algorithms for morphological operations on hexagonal lattices and regular hexagonal domains

Zheng, Xiqiang

Xiqiang Zheng, "Some efficient algorithms for morphological operations on hexagonal lattices and regular hexagonal domains," Proc. SPIE 11584, 2020 International Conference on Image, Video Processing and Artificial Intelligence, 115840B (10 November 2020); doi: 10.1117/12.2579531

SPIE.

Event: Third International Conference on Image, Video Processing and Artificial Intelligence, 2020, Shanghai, China

Some efficient algorithms for morphological operations on hexagonal lattices and regular hexagonal domains

Xiqiang Zheng*
Dept. of Science, Technology, Health, and Human Services,
Voorhees College, Denmark, SC 29042, USA

ABSTRACT

In this paper, morphological operations on regular hexagonal structures are considered, where a regular hexagonal structure is a subset of a hexagonal lattice and consists of the sampled points for the discretization of a regular hexagonal region. First, a sequence of reasonable structure elements (SE) for hexagonal lattices are provided, and the decompositions of the SEs are shown. Second, based on the decompositions of the SEs, some efficient algorithms for morphological operations on the regular hexagonal structures are developed. Finally, the algorithms are tested using a computerized tomography (CT) image, and promising applications of such algorithms on CT image reconstruction and segmentation are pointed out.

Keywords: Image sampling, hexagonal lattices, morphological operations, image reconstruction, image segmentation

1. INTRODUCTION

Let R and Z denote the set of real numbers and integers, respectively. As in [1] by Zheng, if U and V are two vectors (in R^2) that have the same length and the angle between them is 60° or 120°, then the set $\{k_1 \bullet U + k_2 \bullet V \mid k_1, k_2 \in Z\}$ is called a *hexagonal lattice* generated by U and V. Let H be a hexagonal lattice and $p \in H$, the *Voronoi cell* associated with the lattice point p is the set

$$\{x \in R^2 \mid d(x,p) \le d(x,q) \text{ for any } q \in R^2\}.$$

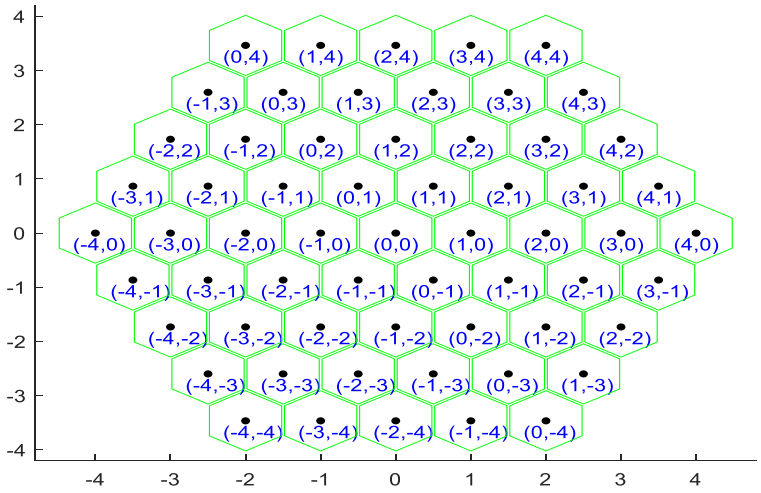

Figure 1: A regular hexagonal structure consisting of the sampled points of a regular hexagonal region.

*xzheng@voorhees.edu

2020 International Conference on Image, Video Processing and Artificial Intelligence,
edited by Ruidan Su, Dora Zhang, Proceedings of SPIE Vol. 11584, 115840B
 · doi: 10.1117/12.2579531

Let $U=[1,0]$ and $V=\left[-\frac{1}{2},\frac{\sqrt{3}}{2}\right]$. If $L:=\{k_1\bullet U+k_2\bullet V \mid k_1,k_2\in Z\}$, then L is a hexagonal lattice generated by U and V. For each positive integer n, let

$$\Xi_n := \left\{k_1\bullet U+k_2\bullet V \;\middle|\; k_1,k_2\in Z, |k_1|\le n, |k_2|\le n, \textit{and } |k_1-k_2|\le n\right\}. \quad (1)$$

Ξ_n is called a *regular hexagonal structure* (RHS) with parameter n. Fig. 1 shows Ξ_4 along with the coefficient vector $[k_1,k_2]$ of the lattice points in Ξ_4 in terms of U and V. As displayed in Fig. 1, each lattice point in a hexagonal lattice has six equidistant neighbors; each Voronoi cell is well connected to six adjacent Voronoi cells; and the lattice points of Ξ_n are well indexed because the coefficient vectors of lattice points in Ξ_n constitute the set

$$\left\{[k_1,k_2]\in Z^2 \;\middle|\; -n\le k_2\le 0, -n\le k_1\le n+k_2\right\}\cup\left\{[k_1,k_2]\in Z^2 \;\middle|\; 1\le k_2\le n, -n+k_2\le k_1\le n\right\}. \quad (2)$$

As shown at Section 6 in [2] by Zheng and Gu, and in [3] by Zheng, because of rotations of computerized tomography (CT) machines, a 2D CT image may be assumed to be circular. The circular region may be embedded into a regular hexagonal region more compactly than the corresponding square region. Hence 2D CT image reconstruction on Ξ_n may not only provide better image quality, but may also produce a smaller set of data. Hence the computational time for some image processing tasks such as image segmentation using graph-cut methods may be reduced.

In [4] by Chackalackal and Basart, morphological techniques are used in areas such as medical imaging and cellular biology. In [5] and [6] by Mostafa et al., binary and grayscale morphological operators on hexagonal images were applied to edge-detection and noise-removal, and achieved improvement over the corresponding operators on the commonly used rectangular Cartesian images. In [7] by Mostafa et al., fuzzy morphological operations with different geometric shapes and directional fuzzy structuring elements on hexagonal lattices were developed and applied to noise removal and edge detection to overcome the deficiency of the corresponding methods for Cartesian images. As shown in Fig. 9 (a) of [6], the previous computer algorithms and codes are for morphological operations on rectangular regions. In this paper, we develop efficient algorithms for morphological operations on images defined on RHSs for regular hexagonal regions and hexagonal lattices. Those algorithms have promising applications on CT image processing.

2. BINARY MORPHOLOGY OPERATORS ON REGULAR HEXAGONAL STRUCTURES

The terminologies for morphological operations including dilation, erosion, opening, and closing can be found in references such as [8] by Youkana, [9] by Bartovsky et al., and [10] by Bartovsky. Let us first consider the morphological operations on binary images defined on RHSs. For each integer $m>0$, a binary image defined on the RHS Ξ_m is just an *indicator function* b on a subset S of Ξ_m, in other words, $b(p)=\begin{cases}1, & if\ p\in S\\ 0, & if\ p\notin S\end{cases}$ for each $p\in\Xi_m$. Hence the binary image can be just denoted as S. Let $m,n\in Z$ with $m\ge n>0$. By Equation 1 for the definitions of RHSs, we have $\Xi_m\supseteq\Xi_n$. The *dilation* of the binary image Ξ_m by the structure element Ξ_n is the binary image defined to be $\Xi_m\oplus\Xi_n:=\{p+q \mid p\in\Xi_m, q\in\Xi_n\}$. The *erosion* of Ξ_m by Ξ_n is defined to be $\Xi_m\Theta\Xi_n:=\{p\in\Xi_m \mid p+q\in\Xi_m \textit{ for each } q\in\Xi_n\}$. The *opening* of Ξ_m by Ξ_n is defined to be $\Xi_m\circ\Xi_n:=(\Xi_m\Theta\Xi_n)\oplus\Xi_n$, and the *closing* of Ξ_m by Ξ_n is defined to be $\Xi_m\bullet\Xi_n:=(\Xi_m\oplus\Xi_n)\Theta\Xi_n$.

When m and n are big, if we compute $\Xi_m\Theta\Xi_n$ directly by the definition of $\Xi_m\Theta\Xi_n$, then the computational time may be huge because the computer needs to check whether $p+q$ is in Ξ_m or not for each $q\in\Xi_n$. However, when

$n=1$, the computational time for $\Xi_m \Theta \Xi_1$ is much less because, for each interior point p of Ξ_m, $p+q$ is always in Ξ_m for each $q \in \Xi_n$. Furthermore, if $p \in \Xi_m$ and p lies at the border of Ξ_m, then p is called a *boundary point* of Ξ_m as in [11] and [12] by Zheng. Because Ξ_m is a RHS, if p is a boundary point of Ξ_m and if $q \in \Xi_1$, then (based on the geometry of Ξ_m and q) it is easy to determine whether $p+q$ is in Ξ_m or not.

As in [13] by Ohn, it is easy to verify that $\Xi_{n+1} = \Xi_n \oplus \Xi_1$ for each integer $n > 0$; in other words, we have the decomposition formulas $\Xi_2 = \Xi_1 \oplus \Xi_1$, $\Xi_3 = (\Xi_1 \oplus \Xi_1) \oplus \Xi_1$, and so on. By Equation 5 in [13], $\Xi_m \Theta \Xi_n$ can be computed iteratively and efficiently. For example, when $n=3$, $\Xi_m \Theta \Xi_3 = ((\Xi_m \Theta \Xi_1) \Theta \Xi_1) \Theta \Xi_1$. This is one of the advantages of hexagonal lattices over *square lattices* (whose Voronoi cells are squares).

For square lattices, as shown in [14] by Solomon and Breckon, the Matlab command se3 = strel('disk',5) creates a structure element which corresponds to a disk of radius 5. The Matlab command decomp=getsequence(se3) decomposes se3 into six basic structure elements. But those six basic structure elements are not the same, and hence difference computer codes are needed for the efficient computation of the corresponding morphological operations on those six basic structure elements.

For hexagonal lattices, as displayed in [6] by Mostafa and Her, the structure element Ξ_m also has the advantage that Ξ_m represents the discretization of a circular region well. Usually we prefer the circular shapes for the structure elements because the neighborhoods for domains of continuous functions are often assumed to be circular. As shown in Fig. 9 (a) in [6], Mostafa and Her studied the morphological operations on square regions with RHSs as structure elements. In this paper, morphological operations on regular hexagonal regions with RHSs as structure elements are considered.

3. GREYSCALE MORPHOLOGY OPERATORS ON REGULAR HEXAGONAL STRUCTURES AND EFFICIENT COMPUTATIONAL ALGORITHMS

The results on the last section can be generalized to greyscale images defined on RHSs. Let $m, n \in Z$ with $m \geq n > 0$; and let f be a greyscale image defined on Ξ_m. As in [10] by Bartovsky, the *greyscale dilation* and *greyscale erosion* of f by the structure element Ξ_n is defined to be $\left[\delta_{\Xi_n}(f)\right](p) := \max\{f(p+q) \mid q \in \Xi_n\}$ and $\left[\varepsilon_{\Xi_n}(f)\right](p) := \min\{f(p-q) \mid q \in \Xi_n\}$ for each $p \in \Xi_m$, respectively. Because Ξ_n is symmetric about the origin, we have $\left[\varepsilon_{\Xi_n}(f)\right](p) = \min\{f(p+q) \mid q \in \Xi_n\}$.

As discussed in the last section, $\delta_{\Xi_n}(f)$ and $\varepsilon_{\Xi_n}(f)$ can be computed iteratively and efficiently using the basic structure element Ξ_1 for n times. Because the difference between the computations of $\delta_{\Xi_n}(f)$ and $\varepsilon_{\Xi_n}(f)$ is just $\max$ versus $\min$ operations, in the following, we just show efficient computational algorithms for $\delta_{\Xi_n}(f)$. In the actual computation, the function f defined on Ξ_m is represented as a $2m+1$ by $2m+1$ matrix $M = (M_{i,j}) \in R^{(2m+1)\times(2m+1)}$ such that

$$f(k_1 \bullet U + k_2 \bullet V) = M_{k_1+m+1,\, k_2+m+1}, \tag{3}$$

for $k_1, k_2 \in Z$ satisfying $|k_1| \leq m$ and $|k_2| \leq m$.

ALGORITHM 1: Efficient algorithm for greyscale dilation of images defined on RHSs by the structure element Ξ_1

Input: A square matrix M with odd number of rows, where M represents a greyscale image defined on a RHS as in Equation 3.

Output: A matrix N of the same size as M, where N represents the dilated image by the structure element Ξ_1.

In the following, we show the details of Algorithm 1 using some Matlab and algorithmic notations. Let r and c denote the rows and columns of the output matrix N, respectively. In this algorithm, c proceeds from 1 to $2m+1$ corresponding to the lowest row to the highest row of Ξ_m as displayed in Fig. 1. For a given c, the boundary points of the corresponding row of Ξ_m are dealt first.

- $m \leftarrow \dfrac{size(M,1)-1}{2}$; $N \leftarrow zeros(2m+1, 2m+1)$;
- $N(1,1) \leftarrow \max\{M(1,1), M(2,1), M(1,2), M(2,2)\}$; $N(m+1,1) \leftarrow \max\{M(m,1), M(m+1,1), M(m+1,2), M(m+2,2)\}$;
- for $r = 2 : m$,

  $N(r,1) \leftarrow \max\{M(r-1,1), M(r,1), M(r+1,1), M(r,2), M(r+1,2)\}$;

  end
- for $c = 2 : m$,

  $N(1,c) \leftarrow \max\{M(1,c), M(2,c), M(1,c-1), M(1,c+1), M(2,c+1)\}$;

  $N(\mathrm{m+c},c) \leftarrow \max\{M(\mathrm{m+c-1},c), M(\mathrm{m+c},c), M(\mathrm{m+c},c+1), M(\mathrm{m+c+1},c+1), M(\mathrm{m+c-1},c-1)\}$;

  for $r = 2 : (m+c-1)$,

  $N(r,c) \leftarrow \max\{M(r-1,c), M(r,c), M(r+1,c), M(r-1,c-1), M(r,c-1), M(r,c+1), M(r+1,c+1)\}$;

  end

  end
- $N(1,m+1) \leftarrow \max\{M(1,m+1), M(2,m+1), M(1,m), M(2,m+2)\}$;
- $N(2m+1,m+1) \leftarrow \max\{M(2m,m+1), M(2m+1,m+1), M(2m+1,m+2), M(2m,m)\}$;
- for $r = 2 : (2m)$,

  $N(r,m+1) \leftarrow \max\{M(r-1,m+1), M(r,m+1), M(r+1,m+1), M(r-1,m), M(r,m), M(r,m+2), M(r+1,m+2)\}$;

  end
- for $c = (m+2) : (2m)$,

  $N(c-m,c) \leftarrow \max\{M(c-m,c), M(c-m+1,c), M(c-m,c-1), M(c-m-1,c-1), M(c-m+1,c+1)\}$;

  $N(\mathrm{2m+1},c) \leftarrow \max\{M(2m,c), M(\mathrm{2m+1},c), M(\mathrm{2m+1},c+1), M(\mathrm{2m+1},c-1), M(2m,c-1)\}$;

for $r=(c-m+1):(2m)$,

$N(r,c) \leftarrow \max\{M(r-1,c), M(r,c), M(r+1,c), M(r-1,c-1), M(r,c-1), M(r,c+1), M(r+1,c+1)\}$;

end

end

- $N(m+1,2m+1) \leftarrow \max\{M(m+1,2m+1), M(m+2,2m+1), M(m+1,2m), M(m,2m)\}$;
- $N(2m+1,2m+1) \leftarrow \max\{M(2m,2m+1), M(2m+1,2m+1), M(2m+1,2m), M(2m,2m)\}$;
- for $r=(m+2):(2m)$,

  $N(r,2m+1) \leftarrow \max\{M(r-1,2m+1), M(r,2m+1), M(r+1,2m+1), M(r,2m), M(r-1,2m)\}$;

end

ALGORITHM 2: Efficient algorithm for greyscale dilation of images defined on RHSs by the structure element Ξ_n

---

Input: An integer $n>0$ and a square matrix M with odd number of rows, where M represents a greyscale image defined on a RHS.

Output: A matrix N of the same size as M, where N represents the dilated image by the structure element Ξ_n.

- $N \leftarrow M$;
- for $k=1:n$

  $N \leftarrow$ the output of Algorithm 1 with input N ;

end

ALGORITHM 3: Naive algorithm for greyscale dilation of images defined on RHSs by the structure element Ξ_n

---

Input: An integer $n>0$ and a square matrix M with odd number of rows, where M represents a greyscale image defined on a RHS.

Output: A matrix N of the same size as M, where N represents the dilated image by the structure element Ξ_n.

- Let H_{coef} be a matrix whose rows are exactly the 2-dimensional vectors in the set $\left\{[k_1,k_2] \in Z^2 \;\middle|\; |k_1| \le n, |k_2| \le n, \text{ and } |k_1-k_2| \le n\right\}$;
- $L_r \leftarrow size(H_{coef},1)$; in other word, L_r denotes the number of rows of the matrix H_{coef} ;
- $m \leftarrow \frac{size(M,1)-1}{2}$; $N \leftarrow zeros(2m+1,2m+1)$; $b \leftarrow \min(\min(M))$;

- for $c = 1 : (m+1)$,

  for $r = 1 : (m+c)$,

  add the vector $[r, c]$ to each row of H_{coef}, and denote the resulting matrix as S;

  initialize T to be the row vector of length L_r such that each entry of T is b;

  for $g = 1 : L_r$,

  $g_1 \leftarrow S(g,1)$; $g_2 \leftarrow S(g,2)$;

  if $g_1 > 0$, $g_2 > 0$, $g_1 < 2m+2$, and $|g_1 - g_2| < m+1$,

  $T(1,g) \leftarrow M(g_1, g_2)$;

  end

  end

  $N(r,c) \leftarrow \max(T)$;

  end

  end

- for $c = (m+2) : (2m+1)$,

  for $r = (c-m) : (2m+1)$,

  add the vector $[r, c]$ to each row of H_{coef}, and denote the resulting matrix as S;

  initialize T to be the row vector of length L_r such that each entry of T is b;

  for $g = 1 : L_r$,

  $g_1 \leftarrow S(g,1)$; $g_2 \leftarrow S(g,2)$;

  if $g_1 > 0$, $g_2 > 0$, $g_1 < 2m+2$, and $|g_1 - g_2| < m+1$,

  $T(1,g) \leftarrow M(g_1, g_2)$;

  end

  end

  $N(r,c) \leftarrow \max(T)$;

  end

  end

4. TEST AND COMPUTATIONAL TIME COMPARISON OF THE ALGORITHMS

Because the inputs of those algorithms are images defined on RHSs, to test those algorithms, we use a greyscale CT image of size 496 by 496 as displayed at the top left of Table 1. Then resample this image into the image defined on the corresponding RHS (using Matlab command *griddata*) as displayed at the top right of Table 1. The resampled image is applied to Algorithms 2 and 3 with Ξ_1, Ξ_3 and Ξ_6 as structure elements, respectively. With Ξ_3 as a structure element, the dilated image and the internal gradient image are displayed at the bottom of Table 1, where the *internal gradient* of an image f with a structure element E is defined to be $f-(f\Theta E)$. For the dilation or erosion, Algorithms 2 and 3 have the same output image, but the computational time of Algorithm 2 is much less than Algorithm 3. Table 2 shows the amounts of computational time for the corresponding Matlab programs on a 2.7 GHz PC for the dilation of Algorithms 2 and 3. Thus, in terms of computational time, Algorithm 2 is much more efficient than Algorithm 3. The reason is that Algorithm 3 needs to check whether a lattice point in a neighborhood (of a lattice point in Ξ_m) is still in Ξ_m or not based on the constraint conditions for the definition of Ξ_m. But Algorithm 2 is based on Algorithm 1which can determine the neighborhood of a boundary lattice point of Ξ_m easily from the index of the boundary lattice point based on the geometry of Ξ_m.

Table 1. Test of algorithms using a CT image. The top left is the original greyscale CT image; the top right is the resampled image using the associated hexagonal lattice; the bottom shows the dilated image (using Algorithm 2) and the internal gradient image, resp.

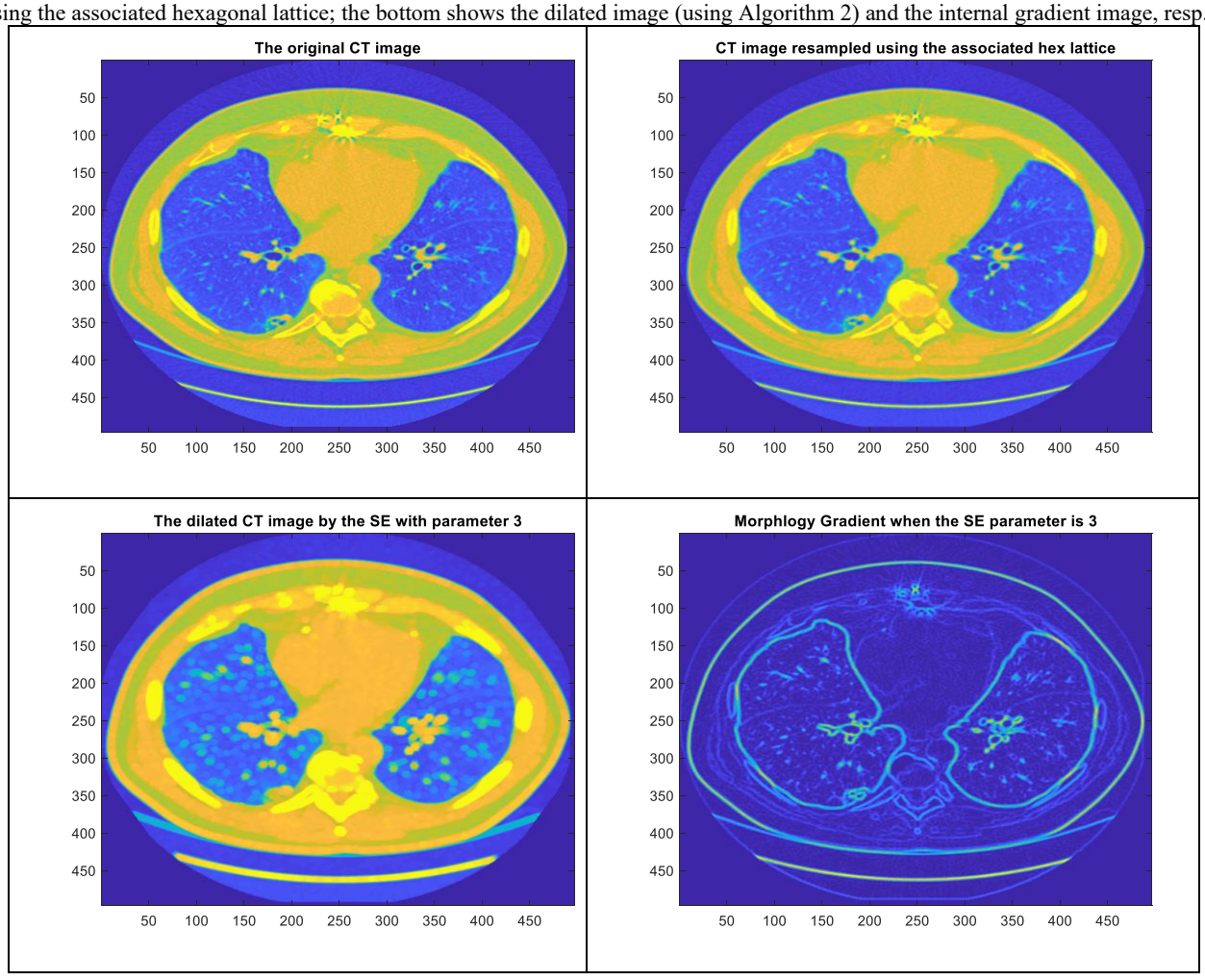

Table 2. The computational time of Algorithms 2 and 3 for the dilation of the image at the top right of Table 1.

Structure Element	Comput. time using Algorithm 2 (Efficient)	Comput. time using Algorithm 3 (Naïve)
Ξ_1	0.0383	0.2671
Ξ_3	0.0792	0.2834
Ξ_6	0.1747	0.6388

5. SUMMARY

We have developed algorithms for the computation of some usual morphological operations on images defined on RHSs, and have shown that the novel Algorithm 2 is very computationally efficient and hence has promising applications. Because Algorithm 2 uses iterative morphological operations with the basic structure element Ξ_1, Algorithm 2 is much more computationally efficient than the naïve Algorithm 3 which needs to check whether a point in a neighborhood is still inside the original image domain or not. We have successfully tested the algorithms using a greyscale CT image and displayed the results. As mentioned in the introduction, RHSs have advantages over the usual rectangular structures for 2D CT image reconstruction and segmentation because they approximate circular regions well and allow efficient data indexing as shown in Equation 2.

For 2D CT image reconstruction, hexagonal lattices have advantages over the usual Cartesian lattices as shown in [15] by Knaup et al., [16] by Mueller and Xu, [17] by Sjölin and Persson, and [3] by Zheng. In this paper, we have just used a usual CT image defined on a square lattice to test the algorithms. The square image is resampled to an image defined on the corresponding hexagonal lattice as the inputs of the algorithms. During the actual CT image reconstruction, a CT image may be reconstructed directly on a hexagonal lattice from a sinogram. Because hexagonal lattices have advantages over square lattices, the CT image reconstruction effect on a hexagonal lattice is better. Furthermore, as shown in the introduction and in Fig. 1, a circular region may be embedded into a regular hexagonal region more compactly the corresponding square region. RHSs can save the number of samples for 2D image reconstruction and processing, and hence the computational time for some image processing tasks such as image segmentation using graph-cut methods may be reduced.

Because each lattice point in a hexagonal lattice has six equidistant neighbors, hexagonal lattices may provide better effect of image segmentation or morphological operations than the usual square lattices as shown in [18] by Vartak and Mankar. In the future, we may do reasonable simulations as in [3] by Zheng to compare those two different kinds of lattices fairly for the effectiveness of those morphological operations on regular hexagonal regions. We may also generalize the research from 2D to 3D as in [1] by Zheng and in [2] by Zheng and Gu. The success of the image processing programs for 2D CT paves the way for the future 3D work. For the 3D CT, no matter whether images are reconstructed on the usual cubic lattices or 3D optimal sampling lattices, to display the reconstructed 3D images on a 2D monitor, the 3D images need to be resampled and the optimal sampling lattices may also provide better resampling effect than the 3D cubic lattices. As in [5, 6, 7] by Mostafa et al., [19] by Steppa and Holch, and [20] by Schlosser et al., another import future research topic is to explore the applications of those algorithms on CT image processing.

ACKNOWLEDGMENTS

This research is partially supported by a Henry C. McBay faculty research fellowship from the United Negro College Fund of the USA, and a National Science Foundation grant of the USA with the Federal Award ID Number 2000158.

REFERENCES

[1] Zheng, X., "Rectangular body-centered cuboid packing lattices and their possible applications," Journal of Computer Science Research 1(2), 1-7 (2019).

[2] Zheng, X. and Gu, F., "Fast Fourier transform on FCC and BCC lattices with outputs on FCC and BCC lattices respectively," J. Math. Imaging Vis. 49(3), 530-550 (2014).

[3] Zheng, X., "Computer simulations for the effectiveness of CT image reconstruction on hexagonal grids and efficient domains," To be published.

[4] Chackalackal, M. S. and Basart, J. P., [NDE X-Ray Image Analysis Using Mathematical Morphology], In: Thompson, D. O. and Chimenti, D. E. (eds) Review of Progress in Quantitative Nondestructive Evaluation, Springer, Boston, MA, 721-728 (1990).

[5] Mostafa, K., Chiang, J. Y. and Her, I., "Edge-detection method using binary morphology on hexagonal images," The Imaging Science Journal 63(3), 168-173 (2015).

[6] Mostafa, K. and Her, I., "An Edge Detection Method for Hexagonal Images," International J. of Image Processing 10(4), 161-173 (2016).

[7] Mostafa, K., Chiang, J. Y., Tsai, W. C. and Her, I., "Fuzzy Noise Removal and Edge Detection on Hexagonal Image," J. of the Chinese Society of Mechanical Engineers, 38(6), 659-667 (2017).

[8] Youkana, I., [Parallelization of Morphological Operators Based on Graphs], (Doctoral Thesis) University of Biskra, 1-94 (1990).

[9] Bartovsky, J., Dokladalova, E., Dokladal, P. and Georgiev, V., "Pipeline architecture for compound morphological operators," IEEE International Conference on Image Processing, 3765-3768 (2010).

[10] Bartovsky, J., [Hardware architectures for morphological filters with large structuring elements], (Ph D. Thesis) University of Paris-Est, 1-94 (2012).

[11] Zheng, X., "Box-counting dimension related to the boundary of hexagonal arrays for an efficient digital Earth model," Fractals 18(2), 1-11 (2010).

[12] Zheng, X., [Efficient Fourier transforms on hexagonal arrays], (Ph D. Thesis) University of Florida, 1-140 (2007).

[13] Ohn, S. Y., [Neighborhood Decomposition of Convex Structuring Elements for Mathematical Morphology on Hexagonal Grid], In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S. and Saygın, Y. (eds) Computer and Information Sciences – ISCIS, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Vol. 4263, 511-521 (2006).

[14] Solomon, C. and Breckon, T., [Fundamentals of Digital Image Processing: a Practical Approach with Examples in MATLAB], John Wiley & Sons, 197-234 (2011).

[15] Knaup, M., Steckmann, S., Bockenbach, O. and Kachelriess, M., "Image reconstruction using hexagonal grids," In: IEEE Nuclear Science Symposium Conf. Record, 3074–3076 (2007).

[16] Mueller, K. and Xu, F., "Optimal Sampling Lattices for High-Fidelity CT Reconstruction," In: IEEE Medical Imaging Conference Record, Orlando, FL, 1-5 (2009).

[17] Sjölin, M. and Persson, M., "Optimal sinogram sampling with temporally offset pixels in continuous rotation CT," Proceedings of the SPIE, Vol. 10132, 13-21 (2017).

[18] Vartak, A. P. and Mankar, V., "Morphological Image Segmentation Analysis," International Journal of Computer Science and Applications 6(2), 0974-1011 (2013).

[19] Steppa, C. and Holch, T. L., "HexagDLy-Processing hexagonally sampled data with CNNs in PyTorch," SoftwareX, 9, 193-198 (2019).

[20] Schlosser, T., Friedrich, M. and Kowerko, D., "Hexagonal Image Processing in the Context of Machine Learning: Conception of a Biologically Inspired Hexagonal Deep Learning Framework," Proceedings of the 18th IEEE International Conference on Machine Learning and Applications (ICMLA), 1866-1873 (2019).