Published in final edited form as:

IEEE Nucl Sci Symp Conf Rec (1997). 2012 ; 2012: 3295–3297. doi:10.1109/NSSMIC.2012.6551751.

Fast and accurate rotator for rotation-based iterative reconstruction algorithms

Krzysztof Kacperski [Member, IEEE] and Jakub Pietrzak

Maria Skłodowska - Curie Memorial Cancer Centre and Institute of Oncology, Warsaw, Poland

Abstract

We present an image rotation algorithm in which the values of pixels of the rotated image are computed as weighted averages of the original grid pixels, with the weights proportional to the overlapping areas of each corresponding pixel pair. The method preserves virtually perfectly the image uniformity and single pixel pulse value in the rotated image. These properties are particularly important to accurately model projections in iterative reconstruction algorithms. The matrix of pixel redistribution coefficients can be precomputed analytically and the algorithm is much faster than e.g. the widely used Gaussian rotator.

I Introduction

Statistical iterative image reconstruction algorithms have been shown to be superior to the previously used analytical methods, like the filtered backprojection, and are now a standard in nuclear medicine. There is also a growing interest in the use of this kind of reconstruction in computed tomography (CT). One of the efficient ways of modeling the projection of image estimate at a variable angle is to rotate the image and sum it along the pixel rows or columns, after applying the appropriate blurring to model the point response function of the imaging system [1],[2]. To assure high quality reconstruction it is important that the rotator does not introduce artefacts which would yield the resulting projections inconsistent. In particular, in the absence of other effects like photon attenuation, a single point value (activity) should give the same total projection value at each angle. Similarly, a uniform area should remain uniform after rotation. Although pixel blur is inevitable (single pixel is mapped into multiple pixels), it should be as low as possible.

In this paper we present an image rotation algorithm based on the assumption that the pixel value represents the intensity which is uniformly distributed within the pixel area. After rotation the intensity is redistributed over the new pixel grid proportionally to the overlap areas between the pixels of the original and rotated grids. This assures preservation of local intensity values as well as image uniformity. The redistribution coefficients can be precomputed for each required angle, allowing for a fast and efficient implementation.

II Materials and Methods

A Overlap of Pixel Areas Rotator

We assume that the pixel value represents the intensity distributed uniformly within the pixel area, and consequently after rotation the value of the pixel is mapped to the new grid proportionally to the fractions of area of the original pixel overlapping the new pixels (Fig. 1). We will refer to the algorithm as the Overlap of Pixel Areas (OPA) rotator.

The fractions of overlapping pixel areas can be computed for each pixel at each required rotation angle either analytically, or numerically e.g. by dividing the pixels into a fine array of subpixels and using simple box-counting. However, the numerical computation requires very long computing time (a few hours for a 128^3 image) and the accuracy of the resulting matrix depends on the fine sampling of pixel grid. We therefore implemented an analytical algorithm which computes the overlapping areas using geometric formulas. Although the algorithm is a bit complicated to code, it yields accurate results and is very fast (a few sec. for a 128^3 image).

As the result, for a $N_{pix} \times N_{pix}$ image and N_{proj} projection angles we get a rotation matrix $R[N_{pix} \times N_{pix} \times N_{proj} \times N_{OL} \times 2]$ containing the addresses of N_{OL} ($=1, 4, 5$ or 6) overlapping target pixels along with the corresponding overlapping fraction coefficients. For a 3D image the same matrix is applied for each slice separately. The rotation is done by multiplying the original image pixels by the corresponding rotation matrix elements and adding the obtained values to the appropriate target pixels:

$$v_{x,y,\varphi_k}^{(new)} = \sum_{i=1}^{N_{OL}} R_{x,y,\varphi_k,i,2} v_{x,y,\varphi_k,i,1}^{(old)}$$

The algorithm can be applied as forward or inverse mapping, yielding the same final result.

B Implementation

We implemented the OPA rotator algorithm using CUDA technology and tested it on Nvidia GeForce GTX 480 and GT 420 graphics cards.

The performance of the algorithm has been evaluated and compared with the commonly used rotation algorithm based on bilinear interpolation [3], as well as the Gaussian rotator [4] designed particularly for iterative reconstruction algorithms, with 3×3 pixels and $\sigma=1$ kernel. Images of a single off-centre non-zero pixel (point source), a uniform disc and a 1 pixel thick section (line source), defined on 128×128 pixel grid, were rotated by 100 angles uniformly distributed over 360° . Subsequently, parallel projections of the rotated images were produced. We computed the integrated amplitudes (total numbers of counts) and the spatial blur of the projected point source at each angle, as well as the uniformity of the rotated discs.

III Results

Table 1 presents the summary of the obtained numeric results.

As expected, the Gaussian rotator, which uses the forward mapping, preserves perfectly the pixel values, but characteristic patterns of the max. amplitude of 2.5 % can be observed in rotated uniform images (Fig. 2).

On the other hand, the inverse mapping bilinear interpolation algorithm preserves perfectly image uniformity, but significant variations in the single pixel values, up to 34%, can appear. The proposed OPA algorithm achieves perfect performance in both projection value consistency and image uniformity. The average pixel blur due to rotation is lower than for the Gaussian rotator. Superior performance can also be observed on the projections of line sources, Fig. 3.

IV Conclusions

The proposed image rotation algorithm based on pixel area redistribution shows the properties desired for iterative tomographic reconstruction algorithms: perfect preservation of single pixel value and image uniformity, and small image blurring. The execution of the algorithm is fast, requiring at most 6 multiplication and summation operations for each rotated pixel. The pixel redistribution matrix can be precomputed for each rotation angle using a fast analytical algorithm. The total execution time is a few times shorter than that of the Gaussian rotator algorithm. The size of the rotation matrix is about $10\times$ the size of the single slice sinogram, and can be easily stored in the computer's RAM.

Although small imperfections of the rotator in an iterative reconstruction algorithm, as those observed for the Gaussian rotator, may be insignificant in the case of noisy, blurred data, like in the typical SPECT scans, they may produce noticeable artefacts in the case of e.g. performance testing when using high count scans of point or line sources, or in the statistical reconstruction of the CT images.

Acknowledgments

This work was supported by the Wellcome Trust under Grant No 084288/Z/07/Z.

References

- [1]Di Bella EVR, et al. A comparison of rotation-based methods for iterative reconstruction algorithms. *IEEE Trans Nucl Sci.* 1996; 43:3370–3376.
- [2]Yendiki A, Fessler JA. A comparison of rotation-and blob-based system models for 3D SPECT with depth-dependent detector response. *Phys Med Biol.* 2004; 49:2157–2168. [PubMed: 15248570]
- [3]Wolberg G. *Digital Image Warping* Los Alamitos, CA: IEEE Computer Society Press; 1990
- [4]Wallis J, Miller T. An optimal rotator for iterative reconstruction. *IEEE Trans Med Imaging.* 1997; 16:118–123. [PubMed: 9050414]

Fig. 1.
Scheme of the Overlap of Pixel Area (OPA) rotation algorithm.

Fig. 2.

Images of the rotated uniform disc rotated using the Gaussian rotator (left) and the proposed Overlap of Pixel Area (OPA) rotator (right).

Fig. 3.

Projection profiles of line sources rotated at two selected angles using three rotation algorithms.

Table I

Performance of the tested rotation algorithms

Performance parameter		Rotator		
		Bilinear	Gaussian	OPA
Single pixel value error (centre/off centre)	min/max [%]	-0/+34.2 -13/+7.3	0/0	0/0
	std. dev. [%]	11.9/4.6	0	0
Uniformity error	min/max [%]	0/0	-2.43/+2.51	0/0
	std. dev. [%]	0	0.65	0
Blur	FWHM [pixel]	0.96	1.22	0.97
Computing time	100 angles, on GTX480 [ms]	48	6046	507
	OSEM, 100 iter. on GT420 [s]	92.8	812	270