Equivalence of Pixel-Driven and Rotation-Based Backprojectors for Tomographic Image Reconstruction

Jeffrey A. Fessler
4240 EECS Bldg., University of Michigan, Ann Arbor, MI 48109-2122
Email: fessler@umich.edu, Voice: 313-763-1434, FAX: 313-764-8041

Abstract

All methods for tomographic image reconstruction require a backprojection step. There have recently been several papers reporting *empirical* comparisons of various backprojection methods. This paper derives an *analytical* expression for "rotation-based" backprojectors, and shows analytically that pixel-driven backprojection is exactly equivalent to rotation-based backprojection when using the same interpolator. This equivalence holds for a very broad class of interpolation methods.

I. BACKPROJECTION

Methods for the backprojection step in tomographic image reconstruction differ in their accuracy and computational complexity, and there have recently been several papers reporting empirical comparisons of various approaches [1–5]. This paper shows analytically that two-dimensional pixel-driven backprojection is exactly equivalent to "rotation-based" ray-driven backprojection under remarkably general conditions.

If complete 2D parallel projections $g_\theta(r)$ of a 2D object $f(x, y)$ are available:

$$g_\theta(r) = \int_{-\infty}^{\infty} f(r\cos\theta - l\sin\theta, r\sin\theta + l\cos\theta)\, dl,$$

for $\theta \in [0, \pi]$ and $r \in \mathbb{R}$, then the classical analytical filtered-backprojection method for tomographic reconstruction is given by

$$f(x, y) = \int_0^\pi q_\theta(x\cos\theta + y\sin\theta)\, d\theta, \quad x, y \in \mathbb{R}, \tag{1}$$

where $q_\theta(\cdot)$ is a ramp-filtered version of $g_\theta(\cdot)$ [6]. In practice only a finite collection of sampled projections $q_{kl} = q_{\theta_k}(r_l)$ is available, where $\theta_k = \pi(k-1)/n_\theta$ for $k = 1, \ldots, n_\theta$, and $r_l = \Delta_r(l - \tau_r)$ for $l \in \mathbb{Z}$, where Δ_r is the radial sample spacing and τ_r is a sample offset (typically 0 or $\frac{1}{2}$). We must reconstruct an estimate of $f(x, y)$ from $\{q_{kl}\}$ using one of several possible methods for discretizing (1).

A. Pixel-driven backprojection

The *pixel-driven* backprojection method can be expressed as follows:

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \hat{q}_{\theta_k}(x\cos\theta_k + y\sin\theta_k),$$

where $\hat{q}_{\theta_k}(r)$ is an interpolated projection:

$$\hat{q}_{\theta_k}(r) = \sum_l q_{kl} \Lambda\left(\frac{r - r_l}{\Delta_r}\right),$$

and $\Lambda(\cdot)$ is an *interpolating function*. Combining the above expressions gives the following explicit analytical form for pixel-driven backprojection:

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_l q_{kl} \Lambda\left(\frac{x\cos\theta_k + y\sin\theta_k - r_l}{\Delta_r}\right). \tag{2}$$

In practice, one evaluates (2) over some finite grid of x and y samples for image display.

This work was supported in part by NIH grants CA-60711 and CA-54362 and the Whitaker Foundation.

In what follows, we allow for any interpolating function that satisfies the following two conditions for any integer i:

$$\Lambda(i) = \begin{cases} 1, & i = 0 \\ 0, & i \neq 0, \end{cases} \tag{3}$$

$$\sum_i \Lambda(\tau - i) = 1, \;\; \forall \tau \in \mathbb{R}. \tag{4}$$

The first condition simply ensures that Λ truly interpolates the sample points. The second condition guarantees that interpolating constant samples gives a constant function, which is a very reasonable restriction. The triangular function (linear interpolation) is particularly popular in tomographic reconstruction, but our result holds in general under the above two conditions.

B. Rotator-based backprojection

An alternative method for backprojection is the ray-driven method based on image rotation. This approach is convenient for parallelization and for implementation on video chips with built-in fast image rotators. In the rotation-based method, one backprojects each projection as if it were at $\theta = 0$, rotates the resulting image by $-\theta$, and then computes the sum over all angles θ_k.

The first step is to backproject the kth projection as if it were at $\theta = 0$. This backprojection must be implemented onto a discrete grid; let x_i and y_j denote the equally-spaced (center) coordinates of pixels on that grid. In what follows it will turn out that a particular choice for x_i is desirable, but for now we allow for a general choice. Let $b_{ij,k}$ denote the grid (over i and j) of backprojected values from the kth projection angle. This step may require interpolation in general, and can be expressed as follows:

$$b_{ij,k} = \hat{q}_{\theta_k}(x_i) = \sum_l q_{kl} \Lambda\left(\frac{x_i - r_l}{\Delta_r}\right), \tag{5}$$

where we assume $\Lambda(\cdot)$ is the same interpolation function used for pixel-driven backprojection.

Rotating the images requires interpolation. Separable interpolation is the natural choice, for which the interpolated image $b_k(\tilde{x}, \tilde{y})$ can be expressed:

$$b_k(\tilde{x}, \tilde{y}) = \sum_i \sum_j b_{ij,k} \Lambda\left(\frac{\tilde{x} - x_i}{\Delta_x}\right) \Lambda\left(\frac{\tilde{y} - y_j}{\Delta_y}\right),$$

where $\tilde{x}, \tilde{y}$ denote coordinates in the rotated coordinate system, and Δ_x and Δ_y are the sample spacings of the grid. Rotating the kth interpolated image by $-\theta_k$ and summing up the rotated images yields the final image estimate:

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} b_k(x \cos\theta_k + y \sin\theta_k, -x \sin\theta_k + y \cos\theta_k).$$

Combining the above expressions and rearranging gives the following explicit analytical forms for rotation-based backprojection:

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_i \sum_j b_{ij,k} \Lambda\left(\frac{x \cos\theta_k + y \sin\theta_k - x_i}{\Delta_x}\right) \Lambda\left(\frac{-x \sin\theta_k + y \cos\theta_k - y_j}{\Delta_y}\right)$$

$$= \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_i \sum_j \sum_l q_{kl} \Lambda\left(\frac{x_i - r_l}{\Delta_r}\right) \Lambda\left(\frac{x \cos\theta_k + y \sin\theta_k - x_i}{\Delta_x}\right) \Lambda\left(\frac{-x \sin\theta_k + y \cos\theta_k - y_j}{\Delta_y}\right)$$

$$= \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_l q_{kl} \left[\sum_i \Lambda\left(\frac{x_i - r_l}{\Delta_r}\right) \Lambda\left(\frac{x \cos\theta_k + y \sin\theta_k - x_i}{\Delta_x}\right)\right] \left[\sum_j \Lambda\left(\frac{-x \sin\theta_k + y \cos\theta_k - y_j}{\Delta_y}\right)\right].$$

The summation in the right-most brackets is simply unity by condition (4). Thus:

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_l q_{kl} \left[\sum_i \Lambda\left(\frac{x_i - r_l}{\Delta_r}\right) \Lambda\left(\frac{x \cos\theta_k + y \sin\theta_k - x_i}{\Delta_x}\right)\right].$$

If we choose $x_i = \Delta_r(i - \tau_r)$, then $\Delta_x = \Delta_r$ and $\left(\frac{x_i - r_l}{\Delta_r}\right) = i - l$, so by (3) the summation within the bracket above simplifies to

$$\hat{f}(x, y) = \frac{\pi}{n_\theta} \sum_{k=1}^{n_\theta} \sum_l q_{kl} \Lambda\left(\frac{x \cos\theta_k + y \sin\theta_k - r_l}{\Delta_r}\right),$$

which is identical to (2).

The choice $x_i = \Delta_r(i - \tau_r)$ is certainly the most natural since in this case the backprojection step (5) reduces to simply taking the vector of samples corresponding to the kth projection and replicating it to form a matrix.

II. Discussion

We have shown analytically that rotation-based ray-driven backprojection is equivalent to pixel-driven backprojection. Thus, *for filtered backprojection* one may implement whichever approach is more suitable for the computing architecture. In principle, for iterative image reconstruction with a system matrix $\boldsymbol{A}$, the backprojection operator should be the exact adjoint $\boldsymbol{A}^T$ of the system matrix. Therefore, the analytical results in this paper should not be taken as advocating cavalier mixing and matching of projection and backprojection methods for iterative image reconstruction, because the effects of mismatch between the two (on convergence, bias, accuracy, etc.) is poorly understood.

References

[1] S C B Lo. Strip and line path integrals with a square pixel matrix: A unified theory for computational CT projections. *IEEE Tr. Med. Im.*, 7(4):355–363, December 1988.

[2] M Tabei and M Ueda. Backprojection by upsampled Fourier series expansion and interpolated FFT. *IEEE Tr. Im. Proc.*, 1(1):77–87, January 1992.

[3] D C Yu and S C Huang. Study of reprojection methods in terms of their resolution loss and sampling errors. *IEEE Tr. Nuc. Sci.*, 40(4):1174–1178, August 1993.

[4] W Zhuang, S S Gopal, and T J Hebert. Numerical evaluation of methods for computing tomographic projections. *IEEE Tr. Nuc. Sci.*, 41(4):1660–1665, August 1994.

[5] E V R Di Bella, A B Barclay, R L Eisner, and R W Schafer. Comparison of rotation-based methods for iterative reconstruction algorithms. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 2, pages 1146–50, 1995.

[6] A C Kak and M Slaney. *Principles of computerized tomographic imaging*. IEEE Press, New York, 1988.

III. Notes

This paper was rejected from T-MI for being “too obvious.” In the words of one reviewer: “Most people can intuitively understand that pixel-driven backprojectors (PDB) and rotation-based backprojectors (RBB) are equivalent, or approximately equivalent, if not exactly so. The content of the paper is simply the proof of the intuition.”

Prior to working out the details, it was not obvious to me that the two would ever be equivalent, since PDB uses 1-D interpolation but RBB uses 2-D interpolation. In fact the only reason I investigated this was that I implemented both methods for the purpose of comparing them as part of a project I was preparing for my medical imaging class, and found them to give identical results, to my surprise. Maybe my intuition is not as refined as the reviewer’s...