

FAST HIERARCHICAL BACKPROJECTION FOR HELICAL CONE-BEAM TOMOGRAPHY

Yoram Bresler and Jeffrey Brokish

Coordinated Science Laboratory
and Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
ybresler@uiuc.edu, brokish@uiuc.edu

ABSTRACT

Existing algorithms for exact helical cone beam (HCB) tomographic reconstruction are computationally infeasible for clinical applications. Their computational cost is dominated by 3-D backprojection, which is generally an $O(N^4)$ operation. We present a fast hierarchical 3-D backprojection algorithm, generalizing fast 2-D parallel beam and fan beam algorithms, which reduces the overall complexity of this step to $O(N^3 \log N)$, greatly accelerating the reconstruction.

1. INTRODUCTION

Helical cone-beam tomography has several advantages over traditional two dimensional tomographic imaging, including decreased scanning times and increased x-ray source utilization. However, image reconstruction from cone beam projections relies on inversion formulas [3], [2] of higher complexity than those found in two dimensional tomography. These algorithms consist of two phases. First, the cone beam projections are individually “filtered” by a spatially varying filter implemented by a combination of weighting, 2-D parallel-beam radon transform, and differentiation. Second, the filtered data is backprojected over the image volume. This second phase has complexity of $O(N^3 P)$ for reconstruction of an $N \times N \times N$ voxel image from P projections. Generally $P = O(N)$, which results in an $O(N^4)$ operation and accounts for a large amount of the computation in the reconstruction process.

Several fast algorithms for backprojection in two dimensional tomography exist. Algorithms based on hierarchical decomposition reduce the complexity of the backprojection operation by successively subdividing the reconstruction area into smaller nonoverlapping regions. As the region size decreases, the number of projections necessary for

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship and by NSF grants Nos. CCR 99-72980 and CCR 02-09203.

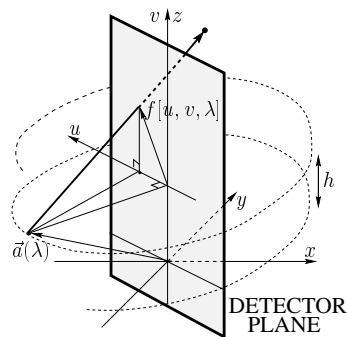


Fig. 1. Helical Cone-Beam Acquisition

accurate reconstruction also decreases. The number of projections can then be reduced, which reduces the computational complexity. This hierarchical decomposition of the backprojection operation initially developed for 2-D parallel beam [1], was extended to fan beam [5] and 3-D cone beam with a circular trajectory (CCB) [4], [6]. Here we extend the method to the more general divergent-beam geometry of a helical trajectory. HCB does not suffer from the inherent artifacts present in CCB but carries a higher computational cost. The fast CCB method decomposes the volume only in the x and y dimensions. Here we fully decompose the volume along all three dimensions. This is important for accurate reconstruction with a helical trajectory, and for extensions to an arbitrary trajectory.

2. ALGORITHM DESCRIPTION

2.1. 3-D Cone Beam Backprojection

Figure 1 shows the setup for helical cone beam projection data acquisition. An x-ray source is placed at equally spaced intervals along a helical trajectory parameterized by $\vec{a}(\lambda) = (R \cos \lambda, R \sin \lambda, \frac{h\lambda}{2\pi})$ where R is the distance between the source and the z-axis and h is the pitch of the helix. The de-

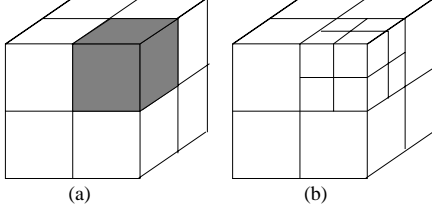


Fig. 2. (a)Octant Decomposition, (b)Recursive Decomposition

ector plane is assumed to contain the z-axis and be perpendicular to $\vec{a}(\lambda)$. After the filtering step is completed for each cone beam projection, the filtered projection data $\tilde{f}[u, v, \lambda]$ is backprojected onto the image volume according to

$$I(\vec{x}) = \sum_{\lambda} \frac{R\sqrt{R^2 + h^2}}{|\langle \vec{x} - \vec{a}(\lambda), (R \cos \lambda, R \sin \lambda, 0) \rangle|^2} \times \tilde{f}(u(\vec{x}, \vec{a}(\lambda)), v(\vec{x}, \vec{a}(\lambda))) \quad (1)$$

where $\langle \cdot, \cdot \rangle$ represents an inner product and the detector plane coordinate $[u(\vec{x}, \vec{a}), v(\vec{x}, \vec{a})]$ is the projection of the point \vec{x} when the source is located at $\vec{a}(\lambda)$. Each object point requires a sum over all the projections, leading to an $O(N^3 P)$ complexity for the entire image. This reconstruction of the object directly from the filtered projection set using (1) will be referred to as the conventional backprojection algorithm for the remainder of this paper.

2.2. Object Decomposition

Consider reconstructing regions of the final object separately as depicted in figure 2a. Each octant can be reconstructed in turn, and after being placed in the proper location will yield the final reconstruction. Although each octant requires only 1/8 the amount of work of entire reconstruction, the total amount of processing is unchanged.

Hierarchical algorithms are based on the principle that the number of projections needed to accurately reconstruct an image is proportional to the size of the image. A single octant in the decomposition has reduced each dimension by half relative to the original problem size. The number of projections required to accurately reconstruct each octant should therefore also be reduced. The method for achieving this reduction in projections is motivated by the procedure developed for fast 2-D fan beam [5] and 3-D circular trajectory cone beam backprojection [6], [4].

The filtered projection data is denoted as $\tilde{f}(\vec{t}, \theta)$, where \vec{t} is the (possibly two dimensional) detector position and θ indexes the source position. Studies on the spectral support of \tilde{f} state that the number of projections required for an object with support restricted to a radius N from the origin is proportional to N . If the number of projections exceeds

that value, the projections are oversampled with regards to θ and can be decimated to reduce the number of projections while still maintaining accurate reconstruction. This is implemented via an angular filter followed by decimation

$$\tilde{f}_D[\vec{t}, \theta] = (\tilde{f} * G)[\vec{t}, 2\theta] \quad (2)$$

where G is a 1-D low-pass filter and the convolution occurs only along the θ dimension. The set \tilde{f}_D now contains half as many projections as \tilde{f} , but can still accurately reconstruct the desired region. The backprojection itself now only requires half as many operations.

The above analysis holds for objects centered at the origin. To adapt this to non-origin centered objects, the projection data is first shifted relative to the projection of the center of the region. The data then mimics the case that the object center was at the origin. Then the projection data can be decimated and the resulting projections shifted back to their original positions for backprojection. In the parallel beam case, this shift exactly represents moving the object to be centered at the origin. In the case of divergent beam tomography, this shift is approximate but successfully reduces the required number of projections for reconstruction.

For helical cone beam data, this process is described as

$$\begin{aligned} \tilde{f}_s[u, v, \lambda] &= \tilde{f}[u - P_\lambda^U(\vec{c}), v - P_\lambda^V(\vec{c}), \lambda] \\ \tilde{f}_d[u, v, \lambda] &= (\tilde{f}_s * G)[u, v, 2\lambda] \\ \tilde{f}_{ds}[u, v, \lambda] &= \tilde{f}_d[u + P_\lambda^U(\vec{c}), v + P_\lambda^V(\vec{c}), \lambda] \end{aligned} \quad (3)$$

$P_\lambda^U(\vec{c})$ and $P_\lambda^V(\vec{c})$ are the u and v coordinates in the detector plane of the projection of the region center \vec{c} when the source is at position λ . The second step involves convolution of \tilde{f}_s with the 1-D low-pass filter G along the λ dimension. The projection data \tilde{f}_{ds} is then backprojected over the volume of interest using (1).

To demonstrate the applicability of this approach for HCB, consider an object consisting of a small sphere. The hierarchical algorithm, described in the next section, was used to generate a decimated (by a factor of 8) set of projections that include a region of interest in the neighborhood of the sphere. This projection set is used to reconstruct the entire object space. Figure 3 shows a plane of the image that intersects the sphere. The colormap has been inverted to show detail. Spoke-like artifacts associated with an inadequate number of projections are visible in the reconstruction, but those artifacts lie outside the region of interest, denoted by the rectangle. Therefore, those artifacts will be truncated and do not affect the final image reconstruction.

2.3. Fast Hierarchical Algorithm

Decomposing the object into 8 octants, decimating the projections using the above procedure and backprojecting the the resulting projections reduces the overall computational

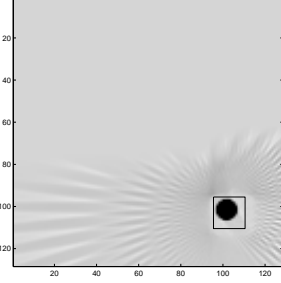


Fig. 3. Reconstruction from Decimated Projections

requirements of the algorithm. In order to reduce the complexity of the backprojection operation, this decomposition must be applied recursively. Each octant is successively decomposed into smaller suboctants as shown in figure 2b. After every subdivision, the projections are shifted and decimated using (3) and the center of the suboctant. This division continues until the octant has reached a minimum size, such as $N = 1$, which limits this decomposition to being performed at most $\log_2 N$ times. The number of projections is halved at each decomposition, so after $\log_2 N$ stages there are $O(\frac{P}{2^{\log_2 N}}) = O(\frac{P}{N}) = O(1)$ projections. Since each point is reconstructed from $O(1)$ projections, the entire object can now be reconstructed in $O(N^3)$. The algorithm does have additional work in the shifting and decimation of projections, and that portion will be what leads to the $O(N^3 \log N)$ complexity.

Under the approximation that an $N \times N \times N$ object generates a set of projections that are each $O(N^2)$ in size, shifting and decimating an object's projection set requires work proportional to $\frac{P N^2}{4}$. At a particular stage k in the decomposition, there are 8^k subregions of size $\frac{N}{2^k}$, each having $\frac{P}{2^k}$ projections. Shifting and decimating these projections for each subregion will require approximately $\frac{P}{2^k} \frac{N^2}{4^k}$, resulting in $P N^2$ work to shift and decimate all the subregions. This shows that the amount of work at each stage is roughly constant and $O(P N^2)$. The number of stages is $\log_2 N$, so the overall complexity becomes $O(P N^2 \log N)$. Replacing $P = O(N)$ yields the final complexity of the algorithm to be $O(N^3 \log N)$, which is a reduction in order from the $O(N^4)$ complexity of (1).

2.4. Tunable Parameters

The above algorithm involves shifting and angular filtering, which must be done accurately for good object reconstruction. Longer filters give better reconstructions but also require more computation, resulting in a trade off between speed and accuracy. An additional consideration is how well the angular oversampling assumption holds. The original projection data may have a source position sampling that

is very close to that of the rate required to reconstruct the object, and therefore the resulting decomposition is close to the limit of the necessary number of projections for decimation. A simple way to address this issue without significantly increasing the length of the angular filter is to perform a single level decomposition without performing the decimation step, or equivalently decomposing the object into 64 subregions for the first stage. This means that each octant now has twice as many projections, making it easier to satisfy the angular bandwidth requirement. The number of times a non-decimated decomposition is performed is referred to as the holdoff factor. The drawback is that since each octant now has twice as many projections, the algorithm must perform twice as much work than if no holdoff were used.

3. RESULTS

Simulated projection data for the standard 3-D Shepp-Logan phantom was generated using analytical expressions for the cone beam projections. The data was then filtered [2] followed by backprojection onto a 512^3 voxel grid using both the conventional and hierarchical backprojection methods, with a holdoff factor of 1 in the hierarchical method. Figures 4-6 compare slices through the reconstructed images, and show good agreement. A very tight range of values is displayed in order to show the differences between the two algorithms. Backprojection using the conventional algorithm required 67.6 hours whereas the hierarchical algorithm required 4.7 hours, a speedup of 14.4. Further optimization in the code implementations will change the relative speedup between the two. However, the hierarchical algorithm has several parameter selections (filter type, filter length, holdoff factor) that can be adjusted along with code optimizations, which may result in an even faster reconstruction using the hierarchical method.

Another method of comparison is the total number of floating point multiplies and divides in the respective algorithms. As a metric, we define the Multiplicative Effort as the total number of floating point multiplies and divides performed in an algorithm. This aids in isolating the data-driven component of the algorithm from other issues such as processor speed and memory size that are relevant to the execution time. For the $N = 512$ case, the conventional algorithm requires a Multiplicative Effort of $2.27 \cdot 10^{12}$, whereas the hierarchical algorithm requires a Multiplicative Effort of $1.54 \cdot 10^{11}$, a ratio of 14.8. The measured speedup in execution time was similar to this ratio.

The Multiplicative Effort for the conventional algorithm and the hierarchical algorithm are compared in figure 7 for a range of problem sizes of $N = 32$ to $N = 1024$. This plot shows the reduction in order of the hierarchical algorithm which indicates the computational savings become

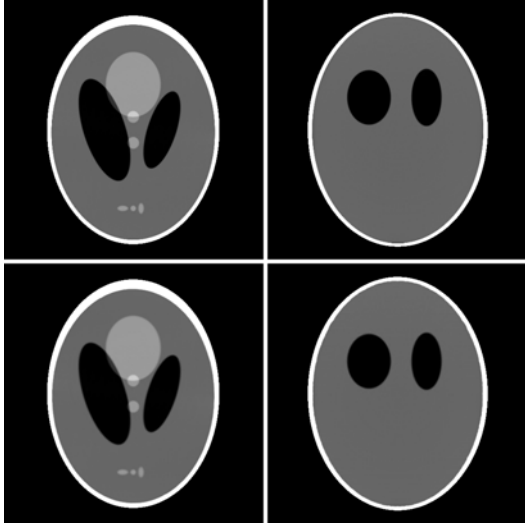


Fig. 4. Reconstruction slices $y = 66$ (left column) and $z = 0$ (right column) for conventional (top row) and hierarchical (bottom row) backprojection methods. The grayscale window is $[1 \ 1.05]$.

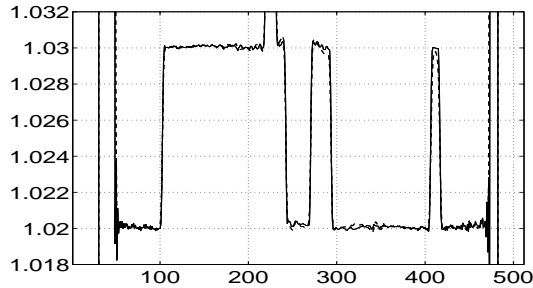


Fig. 5. Cut ($x=0, y=66$) through conventional (solid) and hierarchical (dashed) reconstructions

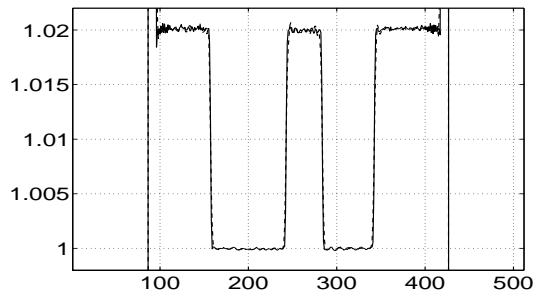


Fig. 6. Cut ($y=-63, z=0$) through conventional (solid) and hierarchical (dashed) reconstructions

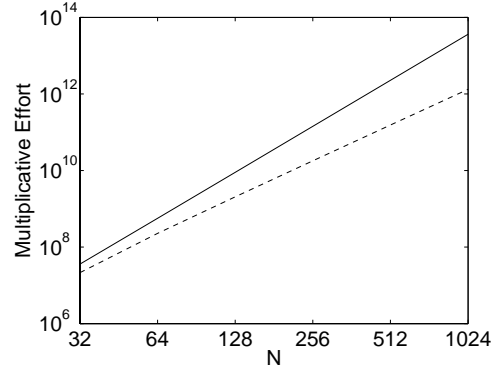


Fig. 7. Multiplicative Effort Plot: Solid = Conventional, Dashed = Hierarchical

even more substantial as the object size increases.

4. CONCLUSION

In this paper, we described a fast hierarchical algorithm for 3-D backprojection in helical cone-beam tomography. The algorithm gives comparable reconstructions to the slow, exact backprojection, and results in a speedup of over an order of magnitude due to its reduction in the computational complexity.

5. REFERENCES

- [1] S. Basu and Y. Bresler. An $O(N^2 \log N)$ filtered backprojection reconstruction algorithm for tomography. In *IEEE Trans. Image Processing*, pages 1760–1773, October 2000.
- [2] H. Kudo, F. Noo, and M. Defrise. Cone-beam filtered-backprojection algorithm for truncated helical data. In *Phys Med Biol*, pages 2885–2909, 1998.
- [3] K. C. Tam, G. Lauritsch, and K. Sourbelle. Exact (spiral + circles) scan region-of-interest cone beam reconstruction via backprojection. In *IEEE Trans. on Med. Imaging*, pages 376–383, May 2000.
- [4] S. Xiao. PhD thesis, University of Illinois at Urbana-Champaign, 2001.
- [5] S. Xiao, Y. Bresler, and D.C. Munson. $O(N^2 \log N)$ native fan-beam tomographic reconstruction. In *Proc. 1st IEEE Int. Symp. Biomedical Imaging, ISBI-2002*, pages 824–827, Washington, DC, July 2002.
- [6] S. Xiao, Y. Bresler, and D.C. Munson. Fast feldkamp algorithm for cone-beam tomographic reconstruction. In *Proc. ICIP 2003*, January 2003.