EM+TV for Reconstruction of Cone-beam CT with Curved Detectors using GPU

Jianwen Chen, Ming Yan, Luminita A. Vese,
John Villasenor, Alex Bui, and Jason Cong, *Fellow, IEEE*

***Abstract*—Computerized tomography (CT) plays a critical role in the practice of modern medicine. However, the radiation associated with CT is significant. Methods that can enable CT imaging at reduced radiation exposure without sacrificing image quality are therefore extremely important. This paper introduces a novel method for enabling improved reconstruction at lower radiation exposure levels. The method is based on the combination of 1) expectation maximization (EM), an iterative method used for CT image reconstruction that maximizes the likelihood function under a Poisson noise assumption, and 2) total variation (TV) regularization, which has been used to preserve edges, given the assumption that most images are piecewise constant. While both EM and TV are known, their combination, as described here, is novel. We show that EM+TV can reconstruct a better image using fewer views, thus reducing the overall dose of radiation. Numerical results show the efficiency of the EM+TV method in comparison to classic EM. In addition, the EM+TV algorithm is implemented on the GPU platform; related implementation methods are also discussed.**

***Index Terms*—Expectation Maximization, Computerized Tomography Reconstruction, Total Variation, GPU Implementation**

I. Introduction

As a group of methods for reconstructing two-dimensional and three-dimensional images from the projections of an object, iterative reconstruction has many applications, including computerized tomography (CT), positron emission tomography (PET), and magnetic resonance imaging (MRI). This technique is quite different from the filtered back projection (FBP) method [3], [11], which is the algorithm most commonly used by manufacturers of commercial imaging equipment. The main advantages of the iterative reconstruction technique over FBP are reduced sensitivity to noise and increased data collection flexibility [8]. For example, the data can be collected over any set of lines, the projections do not have to be distributed uniformly in angle, and the projections can be even incomplete.

There are many available algorithms for iterative reconstruction. Most of these algorithms are based on the system of linear equations $Ax = b$ where $x = (x_1, \cdots, x_N)^T \in \mathbf{R}^N$ is the original unknown image represented as a vector, b is the given measurement with $b = (b_1, \cdots, b_M)^T \in \mathbf{R}^M$, and A is a $M \times N$ matrix describing the direct transformation from the original image to the measurements. A depends on the imaging modality used; for example, in CT, A is the discrete Radon transform, with each row describing an integral along one straight line and all the elements of A are nonnegative.

One example of the iterative reconstruction algorithm is expectation maximization (EM) [4], [5]. This is based on the assumption that the noise in b is Poisson noise. If x is given and A is known, the conditional probability of b is

$$P(b|Ax) = \prod_{i=1}^{M} \frac{e^{-(Ax)_i}((Ax)_i)^{b_i}}{b_i!}.$$

Therefore, given b and A, the objective is to find x such that the above probability is maximized. However, instead of maximizing the probability, we can minimize $-\log P(b|Ax) = \sum_i (Ax)_i - b_i \log((Ax)_i) + C$, with C being a constant. Combined with the nonnegative constraint, the problem becomes

$$\begin{aligned} &\underset{x}{\text{minimize}} \sum_{i=1}^{M} (Ax)_i - b_i \log((Ax)_i) \\ &\text{subject to } x \geq 0. \end{aligned} \tag{1}$$

To derive the EM iterative algorithm, we consider the first order optimality condition of the constrained optimization problem (1). Solving the problem is equivalent to solving the Karush-Kuhn-Tucker (KKT) condition [1], [2]:

$$\begin{aligned} \sum_{i=1}^{M} \left(a_{ij}(1 - \frac{b_i}{(Ax)_i}) \right) - y_j = 0, \qquad & j = 1, \cdots, N, \\ y_j \geq 0, \quad x_j \geq 0, \qquad & j = 1, \cdots, N, \\ y^T x = 0. & \end{aligned} \tag{2}$$

Here, y_j is the Lagrange multiplier corresponding to the constraint $x_j \geq 0$. Multiplying (2) by x_j to eliminate y_j, the EM iteration is as follows:

$$x_j^{n+1} = \frac{\sum_{i=1}^{M} (a_{ij}(\frac{b_i}{(Ax^n)_i}))}{\sum_{i=1}^{M} a_{ij}} x_j^n. \tag{3}$$

The total-variation regularization method was originally proposed by Rudin, Osher and Fatemi [7] to remove noise in an image while preserving edges. This technique is widely

This work was supported by the Center for Domain-Specific Computing (CDSC) under the NSF Expeditions in Computing Award CCF-0926127.

Jianwen Chen and John Villasenor are with Department of Electrical Engineering, University of California, Los Angeles, CA 90095, USA. (E-mail:jwchen@ee.ucla.edu, villa@ee.ucla.edu).

Ming Yan and Luminita A. Vese are with Department of Mathematics, University of California, Los Angeles, CA 90095, USA. (Corresponding author: Ming Yan, E-mail:yanm@math.ucla.edu, lvese@math.ucla.edu).

Alex Bui is with Department of Radiological Sciences, University of California, Los Angeles, CA 90095, USA. (E-mail:buia@mii.ucla.edu).

Jason Cong is with Department of Computer Science, University of California, Los Angeles, CA 90095, USA. (E-mail:cong@cs.ucla.edu).

used in image processing [10], [13] and can be expressed in terms of minimizing an energy functional of the form

$$\min_x \int_\Omega |\nabla x| + \alpha \int_\Omega F(Ax, b),$$

where x is viewed as a two- or three-dimensional image with spatial domain Ω, A is usually a blurring operator, b is the given noisy-blurry image, and $F(Ax, b)$ is a data-fidelity term. For example, for Gaussian noise, $F(Ax, b) = \|Ax - b\|_2^2$.

In this paper we combine the EM algorithm with the total variation (TV) regularization. While each of these methods has been described individually in the literature, to our knowledge they have never been combined in the context of CT reconstruction. The assumption is that the reconstructed image cannot have a total-variation that is too large (thus noise and reconstruction artifacts are removed). For related relevant work, we refer to the Compressive Sensing Resources [15]. Additionally, as an extension of preliminary work [14], the proposed EM+TV algorithm has been implemented on a GPU platform. The related implementation considerations and methodologies are also described. The 10X times speedup indicates that the proposed algorithm has the potential to be used in practical medical CT systems.

II. METHOD (EM+TV)

In the classic EM algorithm, no *a priori* information about the solution is provided. However, if we are given *a priori* knowledge that the solution has homogeneous regions and sharp edges, the objective is to apply this information in order to reconstruct an image with both minimal total-variation and maximal probability. Thus, we can consider finding a Pareto optimal point by solving a scalarization of these two objective functions and the problem becomes

$$\begin{cases} \underset{x}{\text{minimize}} & \int_\Omega |\nabla x| + \alpha \sum_{i=1}^{M} \left((Ax)_i - b_i \log(Ax)_i\right), \\ \text{subject to} & x_j \geq 0, \quad j = 1, \cdots, N, \end{cases}$$

where $\alpha > 0$ is a parameter for balancing the TV regularization and the fidelity term derived from EM. This is a convex constraint problem and we can find the optimal solution by solving the Karush-Kuhn-Tucker (KKT) conditions [1], [2]:

$$-\text{div}\Big(\frac{\nabla x}{|\nabla x|}\Big)_j + \alpha \sum_{i=1}^{M} \Big(a_{ij}(1 - \frac{b_i}{(Ax)_i})\Big) - y_j = 0, \quad j = 1, \cdots, N,$$
$$y_j \geq 0, \quad x_j \geq 0, \quad j = 1, \cdots, N,$$
$$y^T x = 0.$$

By positivity of $\{x_j\}$, $\{y_j\}$ and the complementary slackness condition $y^T x = 0$, we have $x_j y_j = 0$ for every $j = 1, \cdots, N$. Thus after multiplying x_j, we obtain

$$-\frac{x_j}{\sum_{i=1}^{M} a_{ij}} \text{div}\Big(\frac{\nabla x}{|\nabla x|}\Big)_j + \alpha x_j - \alpha \frac{\sum_{i=1}^{M} \Big(a_{ij}(\frac{b_i}{(Ax)_i})\Big)}{\sum_{i=1}^{M} a_{ij}} x_j = 0, \quad j = 1, \cdots, N.$$

The last term is an EM step (3), which can be replaced as x_j^{EM}, and we finally obtain:

$$-\frac{x_j}{\sum_{i=1}^{M} a_{ij}} \text{div}\Big(\frac{\nabla x}{|\nabla x|}\Big)_j + \alpha x_j - \alpha x_j^{EM} = 0, \quad j = 1, \cdots, N, \tag{4}$$

which is the optimality for the following TV minimization problem

$$\underset{x}{\text{minimize}} \int_\Omega |\nabla x| + \alpha \sum_{j=1}^{N} \sum_{i=1}^{M} a_{ij} \left((x)_j - x_j^{EM} \log x_j\right).$$

To solve the above TV minimization problem, we can use semi-implicit iteration for several steps. In order to solve the TV minimization problem, we only have to solve the KKT condition (4). Here we change the notation from x_j to $x_{i,j}$ for one pixel in a two dimensional image. The three dimensional case can be easily derived from the two dimensional one. The semi-implicit iteration is as follows:

$$\begin{aligned} &-\frac{x_{i,j}^n}{V_{i,j}} \frac{x_{i+1,j}^n - x_{i,j}^{n+1}}{\sqrt{\epsilon + (x_{i+1,j}^n - x_{i,j}^n)^2 + (x_{i,j+1}^n - x_{i,j}^n)^2}} \\ &+\frac{x_{i,j}^n}{V_{i,j}} \frac{x_{i,j}^{n+1} - x_{i-1,j}^n}{\sqrt{\epsilon + (x_{i,j}^n - x_{i-1,j}^n)^2 + (x_{i-1,j+1}^n - x_{i-1,j}^n)^2}} \\ &-\frac{x_{i,j}^n}{V_{i,j}} \frac{x_{i,j+1}^n - x_{i,j}^{n+1}}{\sqrt{\epsilon + (x_{i+1,j}^n - x_{i,j}^n)^2 + (x_{i,j+1}^n - x_{i,j}^n)^2}} \\ &+\frac{x_{i,j}^n}{V_{i,j}} \frac{x_{i,j}^{n+1} - x_{i,j-1}^n}{\sqrt{\epsilon + (x_{i+1,j-1}^n - x_{i,j-1}^n)^2 + (x_{i,j}^n - x_{i,j-1}^n)^2}} \\ &+\alpha x_{i,j}^{n+1} - \alpha x_{i,j}^{EM} = 0, \end{aligned}$$

where ϵ is very small. For fixed i j, there is only one unknown variable $x_{i,j}^{n+1}$ that can be easily obtained from the linearized equation. Each iteration is called a TV step. Thus the algorithm is as follows [14]:

Input: $x^0 = 1$;
for *Out=1:1:IterMax* **do**
  $x^{0,0} = x^{Out-1}$;
  for *k = 1:1:K* **do**
    $x^{k,0} = EM(x^{k-1,0})$;
  end
  for *l = 1:1:L* **do**
    $x^{K,l} = TV(x^{K,l-1})$;
  end
  $x^{Out} = x^{K,L}$;
end

Algorithm 1: Proposed EM+TV algorithm.

K is the number of EM iterations and L is the number of TV iterations. K is chosen to be 1 to 3, and L is chosen to be 5 to 10 for the numerical implementation.

Actually, the EM+TV algorithm can also be derived from the general EM algorithm with *a priori* information and

alternating minimization. The convergence analysis of this EM+TV algorithm can be easily obtained and it behaves very well in practice.

III. GPU Implementation

In this section, we consider a fast graphics processing unit (GPU)-based implementation of the most computationally challenging aspects of the EMTV algorithm: forward projection and backward projection. While GPUs were originally developed to accelerate graphics computations, they have been applied to a broad variety of more general computational tasks. While traditional central processing unit (CPU) architectures are designed to support a very broad set of tasks and are well suited to handle branching, GPUs are specifically designed for highly parallel mathematical computations. One of the challenges in highly parallel computation lies in the programming model, and in particular, how "traditional" sequential programming languages can be modified to target the specific architectural features of a class of GPUs. To address this, the company NVIDIA has developed the concept of Compute Unified Device Architecture (CUDA), which provides a unified hardware and software solution for parallel computing on CUDA-enabled NVIDIA GPUs. Since the target GPU for the EM+TV algorithm is an NVIDIA GPU, we use the CUDA framework for optimization.

On a GPU platform, the CUDA-based applications are implemented as kernels for different data portions. The CPU acts as the host and can initiate one kernel at one time. In each kernel, there are three different units, which are called thread, block and grid, respectively. The threads are grouped into blocks, and the blocks are logically aggregated into one grid. In the current version of the hardware, only one grid is supported for one GPU card. On a GPU platform, the threads are scheduled in groups of warps. A warp executes one instruction at one time, so the highest efficiency can be achieved when all the threads within a warp share the same instruction path. If the threads in one warp diverge via a conditional branch, the warp will serially execute each branch path; thus the advantages of parallelization are reduced. The communication between the host and the devices occurs by copying data from/to the CPU's memory to/from the GPU global memory. Threads on the GPU devices will work on the global memory by default. For the kernel codes, to achieve the highest performance, the access of global memory should be minimized. When global memory access cannot be avoided, it is important that all the threads in one warp access consecutive address data.

The following steps were followed in the GPU implementation and optimization [9] [12].

Step 1. Analysis of the degree and granularity of parallelism.

Step 2. Workload profiling and tuning. This involves identifying operations that can be performed using single precision and then measuring the complexity of each module of the overall operation.

Step 3. Optimization of memory accesses

Step 4. Optimization of the instruction flow. This involves identifying and implementing optimizations that would not be performed automatically by the compiler.

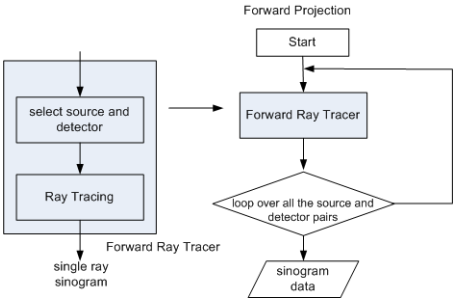

Fig. 1. Forward projection flow chart

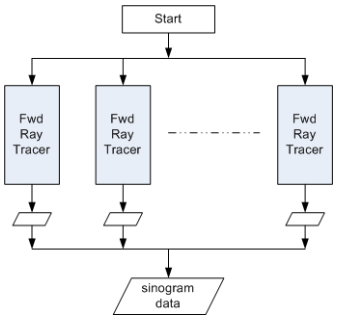

Fig. 2. Forward projection implementation on a GPU

Step 5. Resource balancing. A GPU provides a large amount of shared memory and registers. However, for applications with hundreds of threads, the resources for each thread need to be well-balanced in order to obtain the best overall performance.

Step 6. Optimization of communications between the GPU and host CPU.

The forward projection flow chart is illustrated in Fig. 1. In forward projection, for each pair, it is only necessary to calculate the approximate line integral, without updating the pixels. However, for backward projection, if ray tracing is used, there will be conflict when it is parallelized. Different threads may update the same pixel at the same time, because for a given source-detector pair, all the pixels intersecting with the ray will have to be updated.

The forward projection can be parallelized on a GPU platform as illustrated in Fig. 2. A large number of threads will operate on the forward ray tracer simultaneously for different source and detector pairs. For backward projection, since there are dependencies and conflicts when two threads access one pixel, parallelization is possible, but more challenging. CUDA provides atomic functions to guarantee the mutual exclusion for one same address in memory, and can be used to address potential data conflicts. For backward projection, we use a method similar to the forward projection, the only difference being that all the memory updating operations in a backward projection are atomic operations. The EM+TV algorithm has been implemented on a GPU platform as illustrated in Fig. 3.

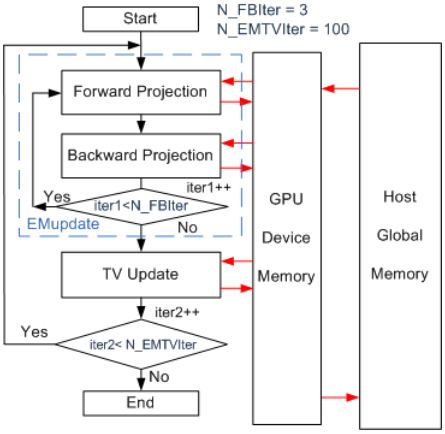

Fig. 3. EM+TV GPU implementation diagram

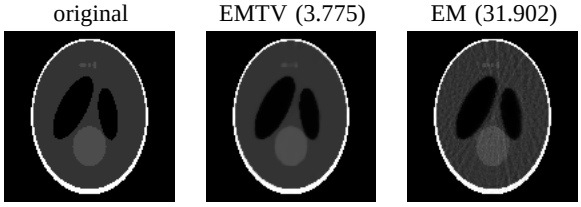

Fig. 4. The middle slices of original image, reconstructed images from EMTV and EM. RMSEs for these two results are provided.

IV. Experimental Results

The EMTV algorithm was tested on a three-dimensional 128x128x128 Shepp-Logan phantom. First, we obtained the projections using Siddon's algorithm [6]. Only 36 views were taken (every 10 degrees), and for each view there were 301x255 measurements. The code is implemented on the GPU platform (Tesla C1060) with single-precision floating point data type. The inner loop of EMupdate has three iterations, and the EMupdate and TVupdate will repeat for 100 iterations. For forward projection and backward projection, 512x64 blocks were used, and for each block there were 288 threads. Compared with the single-thread implementation on a CPU platform (Intel i7-920, 2.66G), implementation on the GPU provides more than 26x speed up for forward projection. For backward projection, because of atomic operations, only 4x speedup can be achieved. And the overall reconstruction time is about 870 seconds, which is about 12x speedup when compared with the CPU implementation. The reconstructed image of the EMTV algorithm on a GPU platform is provided in Fig. 4. For this numerical example, 100 iterations are used for EM+TV, compared with 1000 iterations for EM without regularization. According to the root-mean-square-error (RMSE) between the original and reconstructed images, scaled between 0 and 255, we can see that the result of EM+TV with only 36 views delivers very good quality compared to the EM method without TV regularization.

V. Conclusion

In this paper, we propose a method that use convergence analysis to combine EM and TV; for CT image reconstruction. This method can provide very good results using fewer views. It requires fewer measurements to obtain a comparable image, which results in a decrease of radiation. The method is extended to three dimensions and can be used for real data. One of the challenges in EM+TV is computation time. We have demonstrated that by implementing this method on a GPU platform, execution time can be reduced by well over an order of magnitude. In addition, we believe there are opportunities for further optimizations in areas such as memory access, instruction flow, and parallelization of the backward algorithm that can further improve execution time. In summary, we believe that the combination of algorithms and optimized implementation on appropriate platforms as demonstrated has the potential to enable high-quality image reconstruction with reduced radiation exposure, while also enabling relatively fast image reconstruction times. Future work will focus on an alternative easily parallelized backward projection algorithm and the high performance implementation on an FPGA hardware platform.

References

[1] W. Karush, Minima of functions of several variables with inequalities as side constraints, M.Sc. Dissertation Department of Mathematics, University of Chicago, Chicago, Illinois, 1939.
[2] H. Kuhn and A. Tucker, Nonlinear programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1951, 481-492.
[3] L. Shepp and B. Logan, The Fourier Reconstruction of a Head Section, IEEE Transaction on Nuclear Science, 21 (1974), 21-34.
[4] A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society Series B, 39 (1977), 1-38.
[5] L. Shepp and Y. Vardi, Maximum likelihood reconstruction for emission tomography, IEEE Transaction on Medical Imaging, 1 (1982), 113-122.
[6] R. Siddon, Fast calculation of the exact radiological path for a three-dimensional CT array, Medical Physics, 12 (1986), 252-255.
[7] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physics D, 60 (1992), 259-268.
[8] A. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, Society of Industrial and Applied Mathematics, 2001.
[9] H. Scherl, B. Keck, M. Kowarschik, J. Hornegger, Fast GPU-Based CT reconstruction using the Common Unified Device Architecture (CUDA), IEEE Nuclear Science Symposium Conference Record, M26-280, 2007.
[10] E. Sidky and X. Pan, Image reconstruction in circular cone-beam computed tomography by total variation minimization, Physics in Medicine and Biology, 53 (2008), 4777-4807.
[11] X. Pan, E.Y. Sidky, and M. Vannier, Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction? Inverse Problem 25, (2009), 123009.
[12] Y. Xiao, Z. Chen, L. Zhang, Accelerated CT reconstruction using GPU SIMD parallel computing with bilinear warping method,Information Science and Engineering (ICISE), 2009.
[13] X. Jia, Y. Lou, R. Li, W.Y. Song, and S.B. Jiang, GPU-based fast cone beam CT reconstruction from undersampled and noisy projection data via total variation, Medical Physics 37 (2010), 1757-1760.
[14] M. Yan and L. A. Vese, Expectation maximization and total variation based model for computed tomography reconstruction from undersampled data, In: Proceedings of SPIE Vol. 7961 Medical Imaging 2011: Physics of Medical Imaging, edited by N. J. Pelc, E. Samei and R. M. Nishikawa, 79612X.
[15] Compressive Sensing Resources, http://dsp.rice.edu/cs.