Fast Fat/Water Decomposition Using GPU Computation and Newton's Method

D. H. Johnson[1], S. P. Narayan[2], D. L. Wilson[2,3], C. A. Flask[3]

[1]Ohio State University, [2]Case Western Reserve University, and [3]University Hospitals of Cleveland

ABSTRACT

An improved fat/water estimation technique was developed using Iterative Decomposition of Water and Fat with Echo Asymmetry and Least-squares estimation method and Graphics Computational Units (IDEAL-GPU). The IDEAL-GPU technique produced robust fat and water images quickly and efficiently using a vectorized equation implemented on graphics cards. In addition, Newton's method was used to quickly solve the field inhomogeneity minimization problem. Our initial results show a 2- to 12-fold reduction in processing time when GPU computations are used, which greatly eases the burden of the IDEAL reconstruction time. Fast computation will become even more significant as the trend towards high resolution, whole body mouse and human scanning continues.

INTRODUCTION

We are developing quantitative MRI techniques to quantify fat depots (e.g., visceral, subcutaneous, hepatic, muscular) to determine the role of genetic, environmental, and therapeutic factors on lipid accumulation, metabolism, and disease states. High field MRI scanners (7T-11T) are needed to produce the high resolution images that provide the basis for accurate delineation between visceral and subcutaneous lipid compartments in mice [1,2]. Data processing time is significant because 3-6 image sets at variable echo times must be acquired resulting in >1GB of data. This requires over 1 hour of processing time for each animal. The purpose of this study was to develop a method to more quickly produce fat and water estimates enabling rapid MRI phenotyping.

IDEAL

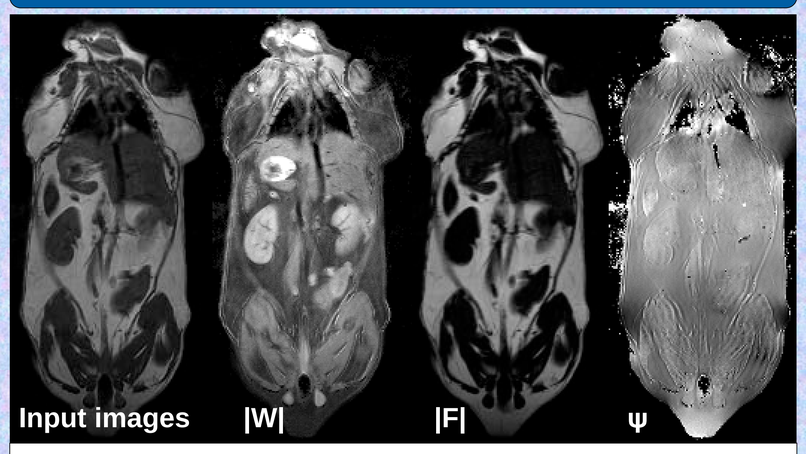

Fig 1. IDEAL provides robust water (|W|) and fat (|F|) estimates from a set of input images. The field inhomogeneity (Ψ) is also found by iterative estimation.

GPU COMPUTATION

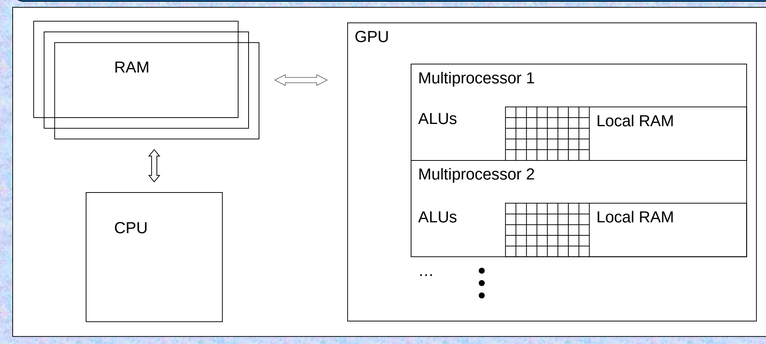

Fig 2. The GPU architecture contains many 'streaming multiprocessors' which process instructions in parallel, which makes it highly efficient for performing the same operations over a large dataset.

IDEAL-GPU

IDEAL was originally formulated for computing one pixel at time, Eqs 1-3.

$$S(TE) = \left(\rho_w + \rho_f e^{j2\pi\Delta f TE}\right) e^{j2\pi\psi TE} \qquad (1)$$

Signal (S) in a pixel at a given TE with unknown water and fat components (ρ_w, ρ_f) and field inhomogeneity (Ψ)

$$\underbrace{\begin{bmatrix} S(TE1) \\ S(TE2) \\ S(TE3) \end{bmatrix}}_{S(x,y)} = \underbrace{\begin{bmatrix} e^{j2\pi\psi TE_1} & 0 & 0 \\ 0 & e^{j2\pi\psi TE_2} & 0 \\ 0 & 0 & e^{j2\pi\psi TE_3} \end{bmatrix}}_{\Psi(x,y)} \underbrace{\begin{bmatrix} 1 & e^{j2\pi\Delta f TE_1} \\ 1 & e^{j2\pi\Delta f TE_2} \\ 1 & e^{j2\pi\Delta f TE_3} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \rho_W \\ \rho_F \end{bmatrix}}_{\rho(x,y)} \qquad (2)$$

Linear system describing the signal at 3 different TEs

$$J(\psi) = \left\| \left(I - AA\dagger\right) \begin{bmatrix} \exp(-j2\pi\psi TE1) & 0 & 0 \\ 0 & \exp(-j2\pi\psi TE2) & 0 \\ 0 & 0 & \exp(-j2\pi\psi TE3) \end{bmatrix} \begin{bmatrix} S(TE1) \\ S(TE2) \\ S(TE3) \end{bmatrix} \right\| \qquad (3)$$

Calculation of the residuals (J) for a given Ψ

However, to take advantage of the parallel processing capabilities of the GPU, these equations must be vectorized. Fortunately, Eq 2 can be expanded column-wise to evaluate every pixel in the image simultaneously. Rewriting Eq 2 after moving Ψ(x,y) to the left hand side gives:

$$\underbrace{\begin{bmatrix} S(x,y,TE_1) & S(x+1,y,TE_1) & \cdots \\ S(x,y,TE_2) & S(x+1,y,TE_2) & \cdots \\ S(x,y,TE_3) & S(x+1,y,TE_3) & \cdots \end{bmatrix}}_{S} .* \underbrace{\begin{bmatrix} e^{-j2\pi\psi(x,y)TE_1} & e^{-j2\pi\psi(x+1,y)TE_1} & \cdots \\ e^{-j2\pi\psi(x,y)TE_2} & e^{-j2\pi\psi(x+1,y)TE_2} & \cdots \\ e^{-j2\pi\psi(x,y)TE_3} & e^{-j2\pi\psi(x+1,y)TE_3} & \cdots \end{bmatrix}}_{\Psi} = \underbrace{\begin{bmatrix} 1 & e^{j2\pi\Delta f TE_1} \\ 1 & e^{j2\pi\Delta f TE_2} \\ 1 & e^{j2\pi\Delta f TE_3} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \rho_W(x,y) & \rho_W(x+1,y) & \cdots \\ \rho_F(x,y) & \rho_F(x+1,y) & \cdots \end{bmatrix}}_{\rho}$$

The critical insight is that the observation matrix (A) does not depend the pixel coordinates. In other words, IDEAL can be implemented as the product of several large matrices corresponding to solving every pixel simultaneously.

IDEAL-GPU RESULTS

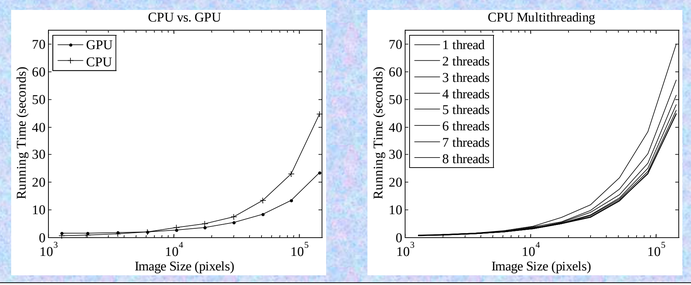

Fig 3. Vectorized IDEAL is faster than CPU computation, even when up to 8 processors are simultaneously used. 2 to 12 fold reductions in execution time were recorded.

OPTIMIZATION METHODS

Minimizing the residuals (J) in Eq 3 is the most time consuming part of IDEAL. Potential algorithms include gradient descent, Golden section search, Brent's method, and VARPRO, which all make different assumptions about whether the desired minimum can be bounded by brackets and whether the function can be approximated by a line or parabola near the minimum. We chose to investigate Newton's method because any order of analytical partial derivatives of J with respect to Ψ are available from Eq 3.

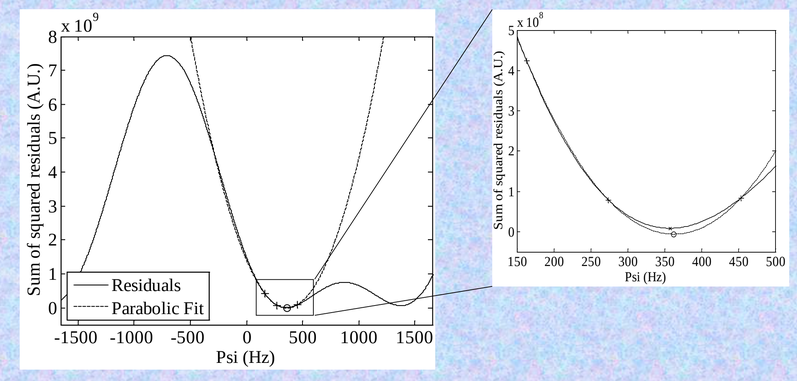

Fig 4. Newton's method uses brackets (+), local inverse quadratic interpolation (○), and the first and second derivatives to quickly find the minimum in the residuals. Newton's method was also be vectorized for GPU implementation. The iterations of Eq 3 required to estimate Ψ in every pixel of the image was reduced from 8-11 iterations down to 3-6 iterations, depending on the initial conditions and local SNR variations. The advantages of Newton's method can be used to reduce execution time or to obtain greater accuracy in Ψ with the same number of iterations.

DISCUSSION

IDEAL provides excellent fat and water estimates, but the processing thrice the data of one acquisition is a burden. Using 6 or more TEs for multi-peak IDEAL [5] is an even greater computational challenge. IDEAL-GPU addresses these problems by reformulating the estimation as a series of efficient matrix multiplications. IDEAL-GPU is not dependent on a specific video card, and we anticipate further speedups with the development of newer video cards with higher clock rates and more processor elements. Also, additional video cards can be used in parallel to reconstruct multiple images simultaneously, allowing for even more scalability.

REFERENCES AND ACKNOWLEDGEMENTS

[1] Johnson et al. JMRI 2008; 28(4):915-927. [2] Johnson et al. JMRI 2010; 31(2):457-465. [3] Reeder et al. MRM 2005; 54(3): 636-644. [4] Johnson et al., ISMRM 2009:2682. [5] Kijowski et al. JMRI 2008; 29(2): 436-442.

This work was supported by NCI R24-CA110943 (Northeast Ohio Animal Imaging Resource Center), NIH 1T32EB007509-01 (Interdisciplinary Biomedical Imaging Training Program), NIH R01EB004070 (Quantitative Image Quality for Optimization of MRI), and NIH 5F30DK082132-02 (Imaging for phenotyping of rodent obesity models). This investigation was conducted in a facility constructed with support from Research Facilities Improvement Program C06 RR12463-01 from the National Center for Research Resources, National Institutes of Health.