

Harnessing graphics processing units for improved neuroimaging statistics

Anders Eklund · Mattias Villani · Stephen M. LaConte

Published online: 28 April 2013
© Psychonomic Society, Inc. 2013

Abstract Simple models and algorithms based on restrictive assumptions are often used in the field of neuroimaging for studies involving functional magnetic resonance imaging, voxel based morphometry, and diffusion tensor imaging. Nonparametric statistical methods or flexible Bayesian models can be applied rather easily to yield more trustworthy results. The spatial normalization step required for multisubject studies can also be improved by taking advantage of more robust algorithms for image registration. A common drawback of algorithms based on weaker assumptions, however, is the increase in computational complexity. In this short overview, we will therefore present some examples of how inexpensive PC graphics hardware, normally used for demanding computer games, can be used to enable practical use of more realistic models and accurate algorithms, such that the outcome of neuroimaging studies really can be trusted.

Keywords Non-parametric statistics · Neuroimaging · Bayesian statistics · Graphics processing units · Spatial normalization · fMRI · VBM · DTI

Introduction

Analysis of neuroimaging data is often computationally demanding. For studies involving functional magnetic resonance imaging (fMRI), voxel-based morphometry (VBM), and diffusion tensor imaging (DTI), it is common to collect data from at least 15 subjects (Friston, Holmes & Worsley, 1999). The size of a single fMRI data set is usually of the order of $64 \times 64 \times 30 \times 200$ elements (200 volumes with 30 slices, each containing 64×64 pixels), and high-resolution volumes for VBM often consist of $256 \times 256 \times 128$ voxels. While increasing statistical power, the large amount of data prevents the use of advanced statistical models, since the calculations required can easily take several weeks. This is especially true for extremely large data sets—for example, the freely available rest data sets in the 1,000 functional connectomes project (Biswal et al., 2010) requiring about 85 GB of storage. With stronger magnetic fields and more advanced sampling techniques for MRI, the spatial and temporal resolution of neuroimaging data will also improve in the near future (Feinberg & Yacoub, 2012), further increasing the computational load.

In this short introductory overview, we will therefore show some examples of how affordable PC graphics hardware, more commonly known as graphics processing units (GPUs), enables the use of more realistic analysis tools. The possibility of performing general computations on GPUs has made it possible to, for example, replace traditional parametric methods with nonparametric alternatives, which would otherwise be prohibitively computationally demanding. Nonparametric methods, such as a Monte Carlo permutation test (Dwass, 1957), make fewer assumptions than do parametric ones and are, therefore, applicable over a wider range of data structures. Kimberg, Coslett and Schwartz (2007) summarized the core of our review: “We can adopt

A. Eklund (✉) · S. M. LaConte
Virginia Tech Carilion Research Institute, Virginia Tech,
2 Riverside Circle,
Roanoke 24016 VA, USA
e-mail: andek034@gmail.com

M. Villani
Division of Statistics, Department of Computer and Information
Science, Linköping University, Linköping, Sweden

S. M. LaConte
School of Biomedical Engineering & Sciences, Virginia Tech-
Wake Forest University, Blacksburg, VA, USA

the perspective that, for many purposes, parametric statistics are a compromise that we have been forced to live with solely due to the cost of computing. That cost has been dropping steadily for the past 50 years, and is no longer a meaningful impediment for most purposes.”

Another route for escaping the shackles of simple parametric models is by using newly developed, flexible semiparametric models from statistics and machine learning. Although often parametric, such models make far fewer restrictive functional and distributional assumptions and, therefore, span a wide array of potential data structures. In this respect, they are similar to nonparametric methods, and we shall refer to both classes of methods as good alternatives to (simple) parametric models. It is common practice to use a Bayesian prior distribution to efficiently regulate these otherwise highly overparametrized models. Bayesian algorithms can be highly computationally demanding, and our review argues that there is a huge potential for using GPUs to speed up Bayesian computations on neuroimaging data.

The review will focus on statistical analysis of fMRI data but will also consider VBM, DTI, and the spatial normalization step required for multisubject studies.

What is a GPU?

A GPU is the computational component of a graphics card used in ordinary computers. The Nvidia GTX 690 graphics card, shown in Fig. 1, contains two GPUs, each consisting of 1,536 processor cores (units that execute program instructions). The physical location of the CPU and two graphics cards in an ordinary PC is shown in Fig. 2. The GPU's large number of cores can be compared with 4 processor cores for a central processing unit (CPU), which normally is used to perform calculations. A GPU core cannot, however, be directly compared with a CPU core. A CPU core is, in general, more powerful due to a higher clock frequency and a much larger cache memory (which stores data just read from the ordinary memory). GPUs can be very fast for a limited number of instructions, while CPUs can handle a much wider range of applications. The CPU is also better at running code with many if-statements, since it has support for so-called branch prediction.



Fig. 1 The Nvidia GTX 690 graphics card, containing two GPUs and a total of 3,072 processor cores



Fig. 2 The physical location of a CPU and two graphics cards in an ordinary PC. By using two Nvidia GTX 690 graphics cards, the user gets a PC equipped with 6,144 processor cores at the price of about \$3,000

Graphics cards were originally designed for computer graphics and visualization. Due to the constant demand for better realism in computer games, the computational performance of a GPU has, during the last 2 decades, increased much more quickly than that of a CPU. The theoretical computational performance can today differ by a factor of ten in favor of the GPU. Graphics cards are also inexpensive, since ordinary consumers must be able to afford them. The Nvidia GTX 690 is one of the most expensive cards and costs about \$1,000.

Why use a GPU?

The main motivation for using a GPU is that one can save time or apply an advanced algorithm, instead of a simple one. In medical imaging, GPUs have been used for a wide range of applications (Eklund, Dufort, Forsberg & LaConte, 2012b). Some examples are to speed up reconstruction of data from magnetic resonance (MR) and computed tomography (CT) scanners and to accelerate algorithms such as image registration, image segmentation, and image denoising. Here, we will focus on how GPUs can be used to lower the processing time of computationally demanding algorithms and methods for neuroimaging.

Some disadvantages of GPUs are their relatively small amount of memory (currently 1–6 GB) and the fact that GPU programming requires deep knowledge about the GPU architecture. Consumer GPUs can also have somewhat limited support for calculations with double precision (64-bit floats), although single precision (32-bit floats) is normally sufficient for most image-processing applications. For the latest Nvidia generation of consumer graphics cards (named Kepler), the performance for double precision can

be as low as 1/24 of the performance for single precision. For professional Kepler graphics cards, the performance ratio can, instead, be 1/3. Professional graphics cards are, however, more expensive. The Nvidia Tesla K20, for example, currently costs about \$3,500.

How can a GPU be used for arbitrary calculations?

Initially, a GPU could be programmed only through computer graphics programming languages (e.g., OpenGL, DirectX), which made it hard to use a GPU for arbitrary operations. Despite this fact, using GPUs for general purpose computing (GPGPU) has been popular for several years (Owens et al., 2007). Through the release of the CUDA programming language in 2007, using Nvidia GPUs to accelerate arbitrary calculations has become much easier, since CUDA is very similar to the widely used C programming language. A large number of reports on large speedups, as compared with optimized CPU implementations, have since then been presented (Che, Boyer, Meng, Tarjan, Sheaffer & Skadron, 2008; Garland et al., 2008). A drawback of CUDA is that it only supports Nvidia GPUs, while the open computing language (OpenCL¹) supports any hardware (e.g., Intel CPUs, AMD CPUs, Nvidia GPUs, and AMD GPUs).

Why use a GPU instead of a PC cluster?

As compared with a PC cluster, which often is used for demanding calculations and simulations, an ordinary PC equipped with one or several graphics cards has several advantages. First, PC clusters are expensive, while a powerful PC does not need to cost more than \$2000–3000 and can be bought “off the shelf.” Second, PC clusters can be rather large and use a lot of energy, while GPUs are small and power efficient. Third, it is hard for a single user to take advantage of the full computational power of a PC cluster, since it is normally shared by many users. On the other hand, a PC cluster can have a much larger amount of memory (but a single user normally cannot use more than a fraction of it). Table 1 contains a comparison between a PC cluster (from 2010) and a regular computer with several GPUs (from 2012). A good PC cluster is clearly a major investment, while a computer with several GPUs can be bought by a single researcher.

How fast is a GPU?

A GPU uses its large number of processor cores to process data in parallel (many calculations at the same time), while a CPU normally performs calculations in a serial manner (one

Table 1 A comparison between a GPU supercomputer, shown in Fig. 2, and a PC cluster in terms of cost, computational performance, amount of memory, and power consumption

Property/hardware	GPU supercomputer	PC cluster
Cost	\$3,000	\$400,000
Theoretical computational performance, single precision	11.2 TFLOPS	5.0 TFLOPS
Theoretical computational performance, double precision	0.5 TFLOPS	2.5 TFLOPS
Total amount of memory	8 GB	2.4 TB
Power consumption	1.0 kW	13.0 kW

Especially note the difference in power efficiency for calculations with single precision; the GPU supercomputer is 29 times more power efficient. TFLOPS stands for terra floating point operations per second

at a time). The main difference between serial and parallel processing is illustrated in Fig. 3. Multicore CPUs, which today are standard, can, of course, also perform parallel calculations, but most of the software packages used in the field of neuroimaging do not utilize this property. AFNI² is one of the few software packages that has multicore support for some functions, by using the OpenMP³ (open multiprocessing) library. For many applications, such as image registration, a hybrid CPU–GPU implementation yields the best performance. The GPU can calculate a similarity measure such as mutual information in parallel, while the CPU runs a serial optimization algorithm.

The performance of a GPU implementation greatly depends on how easy it is to run a certain algorithm in parallel. Fortunately, neuroimaging data are often analyzed in exactly the same way for each pixel or voxel. Many of the algorithms commonly used for neuroimaging are therefore well suited for parallel implementations, while algorithms where the result in one voxel depends on the results in other voxels may be harder to run in parallel.

The processing times for some necessary processing steps in fMRI analysis, for three common software packages (SPM,⁴ FSL,⁵ and AFNI), an optimized CPU implementation that uses all cores and a GPU implementation (Eklund, Andersson & Knutsson, 2012a) are stated in Table 2. The size of the fMRI data set used is 180 volumes of the resolution 64×64×33 voxels. The comparison has been done with a Linux-based computer equipped with an Intel Core i7-3770K 3.5 GHz CPU, 16 GB of memory, an OCZ 128 GB SSD drive, and a Nvidia GTX 680 graphics card with 4 GB of video memory. This is not a fair comparison, since the SPM software, for example, often writes

¹ <http://www.khronos.org/opencl/>

² <http://afni.nimh.nih.gov/afni>

³ <http://www.openmp.org>

⁴ <http://www.fil.ion.ucl.ac.uk/spm>

⁵ <http://fsl.fmrib.ox.ac.uk/fsl>

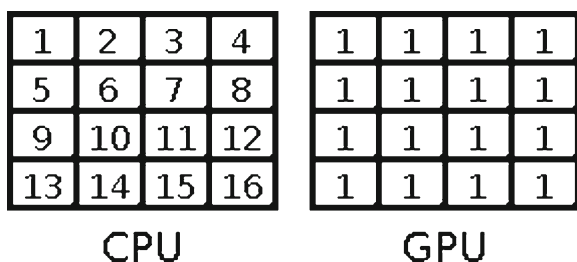


Fig. 3 The figure shows the main difference between a CPU and a GPU for processing of a small image consisting of 16 pixels. The numbers represent in which order the pixels are processed. The CPU processes the pixels one by one, while the GPU processes all the pixels at the same time

intermediate results to file. This is possibly explained by the fact that an fMRI data set could not be fitted into the small memory of ordinary computers when the SPM software was created some 20 years ago. The different software packages and the GPU implementation also use different algorithms for motion correction and model estimation. For example, we use a slightly more advanced algorithm for estimation of head motion (Eklund, Andersson & Knutsson, 2010). Instead of maximizing an intensity-based similarity measure, the algorithm matches structures such as edges and lines. The comparison, however, shows that researchers in neuroimaging can save a significant amount of time by using a multicore CPU implementation that does not involve slow write and read operations to the hard drive. The multicore CPU implementation and the GPU implementation perform exactly the same calculations. Even more time can thus be saved by using one or several GPUs. It should be noted that SPM, FSL, and AFNI are flexible tools that can perform a wide range of analyses and that the mentioned GPU implementations currently can handle only a small subset of these.

This review will not consider any further details about GPU hardware or GPU programming. The interested reader is referred to books about GPU programming (Kirk & Hwu, 2010; Sanders & Kandrot, 2010), the CUDA programming guide and our recent work on GPU accelerated fMRI analysis (Eklund et al., 2012a). The focus will instead be on some types of methods and algorithms that can benefit from higher computational performance.

Methods and algorithms

Nonparametric statistics

In the field of fMRI, the data are normally analyzed by applying the general linear model (GLM) to each voxel time series separately (Friston, Holmes, Worsley, Poline, Frith & Frackowiak, 1995b). The GLM framework is based on a

number of assumptions about the errors—for example, that they are normally distributed and independent. Noise from MR scanners is, however, neither Gaussian nor white but generally follows a Rician distribution (Gudbjartsson & Patz, 1995) and a power spectrum that resembles a $1/f$ function (A. M. Smith et al., 1999). To calculate p -values that are corrected for the large number of tests in fMRI, random field theory (RFT) is frequently used for its elegance and simplicity (Worsley, Marrett, Neelin & Evans, 1992). RFT, however, requires additional assumptions to be met. If any of the assumptions are violated, the resulting brain activity images can have false-positive “active” voxels or be too conservative to detect true positives. A simplistic model of the fMRI noise can, for example, result in biased or erroneous results, as shown in our recent work (Eklund, Andersson, Josephson, Johannesson & Knutsson, 2012c). RFT is also used for VBM (Ashburner & Friston, 2000) and DTI (e.g. (Rugg-Gunn, Eriksson, Symms, Barker & Duncan, 2001)), where the objective is to detect anatomical differences in brain structure. Recent work showed that the SPM software can also yield a high degree of false positives for VBM when a single subject is compared with a group (Scarpazza, Sartori, De Simone & Mechelli, 2013).

To complicate things further, multivariate approaches in neuroimaging (Habeck & Stern, 2010) can yield a higher sensitivity than univariate ones (e.g., the GLM), by adaptively combining information from neighboring voxels. Multivariate approaches are especially popular for fMRI (Björnsdotter, Rylander & Wessberg, 2011; Friman, Borga, Lundberg & Knutsson, 2003; Kriegeskorte, Goebel & Bandettini, 2006; LaConte, Strother, Cherkassky, Anderson & Hu, 2005; McIntosh, Chau & Protzner, 2004; Mitchell et al., 2004; Nandy & Cordes, 2003; Norman, Polyn, Detre & Haxby, 2006) but can also be used for VBM (Bergfield et al., 2010; Kawasaki et al., 2007) and DTI (Grigis et al., 2012). It is, however, not always possible to derive a parametric null distribution for these more advanced test statistics, to threshold the resulting statistical maps in an objective way.

A nonparametric test, on the other hand, is generally based on a lower number of assumptions—for example, that the data can be exchanged under the null hypothesis. Permutation tests were rather early proposed for neuroimaging (Brammer et al., 1997; Bullmore et al., 2001; Holmes, Blair, Watson & Ford, 1996; Nichols & Hayasaka, 2003; Nichols & Holmes, 2002) but are generally limited by the increase in computational complexity. To perform all possible permutations of a data set is generally not possible; a time series with only 13 samples can, for example, be permuted in more than 6 billion ways. Fortunately, a random subset of all the possible permutations (e.g., 10,000) is normally sufficient to obtain a good estimate of the null distribution. These subset permutation tests are known as Monte Carlo permutation

Table 2 Processing times for three necessary steps in fMRI analysis, for three common software packages, a multicore CPU implementation, and a GPU implementation

Processing step/software	SPM	FSL	AFNI	Multicore CPU	GPU
Motion correction	52 s	36 s	5 s	37 s	1.2 s
Smoothing	31 s	10 s	0.4 s	0.4 s	0.022 s
Model estimation	25 s	4.8 s	0.5 s	0.011 s	0.0008 s

The three common software packages use different algorithms, while the multicore CPU implementation and the GPU implementation perform exactly the same calculations. The large speedup for the multicore CPU implementation, for smoothing and model estimation, as compared with SPM and FSL, is mainly explained by the fact that it does not involve slow write and read operations to the hard drive. The AFNI software package is clearly fastest of the three common packages

tests (Dwass, 1957) and will here be called *random* permutation tests. For multivariate approaches to brain activity detection, training and evaluation of a classifier may be required in each permutation, which can be very time consuming. In the work by Stelzer, Chen and Turner (2013), 7 h of computation time was required for a classification-based multivoxel approach combined with permutation and bootstrap. Table 3 states the processing times for 10,000 runs of GLM model estimation and smoothing for the different implementations. Here, we have also included processing times for a multi-GPU implementation, which uses the four GPUs in the PC shown in Fig. 2. Each GPU can independently perform a portion of the random permutations. Clearly, the long processing times of standard software packages prevent easy use of nonparametric tests.

The main obstacle for a GPU implementation of a permutation test is the irregularity of the random permutations, which severely limits the performance. Due to this, only two examples of GPU accelerated permutation tests have been reported (Shterev, Jung, George & Owzar, 2010; van Hemert & Dickerson, 2011). Fortunately, a random permutation of several volumes—for example, an fMRI data set or anatomical high-resolution volumes for VBM group analysis, can be performed efficiently on a GPU, if the same permutation is applied to a sufficiently large number of voxels (e.g., 512). In neuroimaging, one normally wishes to apply the same permutation to all voxels, in order to keep the spatial correlation structure. It is thereby rather easy to use GPUs to speedup permutation tests for neuroimaging.

By analyzing 1,484 freely available rest (null) data sets (Eklund et al., 2012c), a random permutation test was

shown to yield more correct results than the parametric approach used by the SPM software. The main reason is that SPM uses a rather simple model of the GLM errors. Performing 10,000 permutations of 85 GB of data is equivalent to analyzing 850 000 GB of data. Table 4 states the processing times for 10,000 permutations of 1,484 fMRI data sets for the different implementations. To compare parametric and nonparametric approaches to fMRI analysis is clearly not possible without the help of GPUs (or a PC cluster). A random permutation test can also, as a bonus, be used to derive null distributions for more advanced test statistics. In our recent work (Eklund, Andersson & Knutsson, 2011a), we took advantage of a GPU implementation to objectively compare activity maps generated by the GLM and canonical correlation analysis based fMRI analysis (Friman et al., 2003), which is a multivariate approach. We have also accelerated the popular searchlight algorithm (Kriegeskorte et al., 2006), making it possible to perform 10,000 permutations including leave-one-out cross validation in 5 min instead of 7 h (Eklund, Björnsdotter, Stelzer & LaConte, 2013).

We have here focused on fMRI, but permutation tests can also be applied to VBM (Bullmore, Suckling, Overmeyer, Rabe-Hesketh, Taylor & Brammer, 1999; Kimberg et al., 2007; Silver, Montana & Nichols, 2011; Thomas, Marrett, Saad, Ruff, Martin & Bandettini, 2009) and DTI (e.g., proposed by Smith et al. (2006) and used by Chung, Pelletier, Sdika, Lu, Berman and Henry (2008) and Cubon, Putukian, Boyer and Dettwiler (2011)) data. Other nonparametric approaches include jackknifing, bootstrapping, and cross-validation. Biswal, Taylor and Ulmer (2001) used jackknife to estimate confidence intervals of fMRI

Table 3 Processing times for 10,000 runs of GLM model estimation and smoothing for a single fMRI data set

Processing step/software	SPM	FSL	AFNI	Multicore CPU	Single GPU	Multi-GPU
Smoothing	86.1 h	27.8 h	1.1 h	1.1 h	220 s	55 s
Model estimation	69.4 h	13.3 h	1.4 h	110 s	8 s	2 s

The long processing times with standard software packages prevent easy use of nonparametric tests

Table 4 Processing times for 10,000 runs of GLM model estimation and smoothing for 1,484 fMRI data sets

SPM	FSL	AFNI	Multicore CPU	Single GPU	Multi-GPU
26.3 years	7 years	154 days	71 days	3.9 days	1 day

GPUs make it possible to apply nonparametric statistical methods to large data sets

parameters, while Wilke (2012) instead used jackknife to assess the reliability and power of fMRI group analysis. Bootstrap has been applied to fMRI (Auffermann, Ngan, & Hu 2002; Bellec, Rosa-Neto, Lyttelton, Benali & Evans, 2010; Nandy & Cordes, 2007) and DTI (Grigis et al., 2012; Jones & Pierpaoli, 2005; Lazar & Alexander, 2005), as well as VBM (Zhu et al., 2007). GPUs can, of course, also be used to speed up these other nonparametric algorithms (see, e.g., the review by Guo, 2012).

Bayesian statistics

Bayesian approaches are rather popular for fMRI analysis (Friston, Penny, Phillips, Kiebel, Hinton & Ashburner, 2002; Genovese, 2000; Gössi, Fahrmeir & Auer, 2001; see the review by Woolrich, 2012, for a recent overview). A major advantage of Bayesian methods is that they can incorporate prior information in a probabilistic sense and consider uncertainties in a straightforward manner. Bayesian methods are usually the preferred choice for richly parameterized semiparametric models (see the Introduction). Model selection and prediction are also much more straightforward in a Bayesian setting. To calculate the posterior distribution can, however, be computationally demanding if Markov chain Monte Carlo (MCMC) methods need to be applied. Genovese stated that a day of processing time was required for a single data set, for a simple noise model assuming spatial independence. In the work by Woolrich, Jenkinson, Brady and Smith (2004), fully Bayesian analysis of a single slice took 6 h—that is, about 7 days for a typical fMRI data set with 30 slices. Today, the calculations can perhaps be performed in less than an hour with an optimized CPU implementation. Variational Bayes (VB) can be used, instead, to derive an approximate analytic expression of the posterior distribution—for example, for estimation of autoregressive parameters for fMRI time series (Penny, Kiebel & Friston, 2003)—or in order to include spatial priors in the fMRI analysis (Penny, Trujillo-Barreto & Friston, 2005). A first problem is that a large amount of work may be required to derive the necessary equations, which often is straightforward for MCMC methods. Second, most VB applications assume that the posterior distribution factorizes into several independent factors, to obtain analytic updating equations. Third, tractability typically necessitates a restriction to conjugate priors (Woolrich et

al., 2004). This restriction can be circumvented by instead using approximate VB.

To our knowledge, an unexplored approach to Bayesian fMRI analysis is to perform calculations with large spatio-temporal covariance matrices, in order to properly model nonstationary relationships in space and time. For example, a neighborhood of $5 \times 5 \times 5$ voxels for 80 time samples can be considered as one sample from a distribution with 10,000 dimensions, rather than 10,000 samples from a univariate distribution. The main problem with such an approach is that the error covariance matrices will be of the size $10,000 \times 10,000$. GPUs can be used to speed up the inversion of these large covariance matrices, which is required in order to calculate the posterior distribution of the model parameters. To estimate the covariance matrix itself, a first approach can be to use a Wishart prior. A better prior can be obtained by analyzing large amounts of data—for example, the rest data sets in the 1,000 functional connectomes project (Biswal et al., 2010). Such a prior may, however, require MCMC algorithms for inference.

In the field of statistics, there is a growing literature on using GPUs to accelerate statistical inference (see the work by Guo, 2012, for a review on parallel statistical computing in regression analysis, nonparametric inference, and stochastic processes). Suchard, Wang, Chan, Frelinger, Cron and West (2010) focused on how to use a GPU to accelerate Bayesian mixture models. As a proof of concept, we made a parallel implementation of an MCMC algorithm with a tailored proposal density, described by Chib and Jeliazkov (2001). The processing time for their example in Section 3.1 was reduced from 18 s to 75 ms. The traditionally used MCMC algorithms are sequential and, therefore, not amendable to simple parallelization, except in a few special cases. In fMRI, this can be circumvented by running many serial MCMC algorithms in parallel (Lee, Yau, Giles, Doucet & Holmes, 2010)—for example, one for each voxel time series.

Ferreira da Silva (2011a) implemented a multilevel model for Bayesian analysis of fMRI data and combined MCMC with Gibbs sampling for inference. As was previously proposed, a linear regression model was fitted in parallel for each voxel. Random number generation was performed directly on the GPU, through the freely available CUDA library CURAND, to avoid time-consuming data transportation between the CPU and the GPU (see Ferreira da Silva,

2011b, for more details on the GPU implementation). Processing of a single slice took 452 s on the CPU and 65 s on the GPU. For a data set with 30 slices, this gives a total of 30 min, which still is too long for practical use. The graphics card that was used was somewhat outdated; a more modern card would likely yield an additional speedup of a factor of at least 10, resulting in a processing time of about 3 min, as compared with more than 3.5 h on the CPU.

For DTI, GPUs have been used to accelerate a Bayesian approach to stochastic brain connectivity mapping (McGraw & Nadar, 2007) and a Bayesian framework for estimation of fiber orientations and their uncertainties (Hernandez, Guerrero, Cecilia, Garcia, Inuggi & Sotiropoulos, 2012). This framework normally requires more than 24 h of processing time for a single subject, as compared with 17 min with a GPU. We believe that GPUs are a necessary component to enable regular use of Bayesian methods in neuroimaging, at least for methods that rely on a small number of assumptions.

Spatial normalization

Multisubject studies of fMRI, VBM, and DTI normally require spatial normalization to a brain template (Friston, Ashburner, Frith, Poline, Heather & Frackowiak, 1995a). This image-processing step is generally known as image registration but is often called “normalization” in the neuroimaging literature. A suboptimal registration can lead to artifacts, such as brain activity in the ventricles or artifactual differences in brain anatomy. In general, there is no perfect correspondence between an anatomical volume and a brain template (Roland et al., 1997). The spatial normalization step was early acknowledged as a problem for VBM (Bookstein, 2001), as well as for fMRI (Brett, Johnsrude & Owen, 2002; Nieto-Castanon, Ghosh, Tourville & Guenther, 2003; Thirion, Flandin, Pinel, Roche, Ciuciu & Poline, 2006) and DTI (Jones & Cercignani, 2010; Jones et al., 2002). Another problem is that MR scanners often do not yield absolute measurements, as CT scanners do, but relative ones. A difference in image intensity between two volumes can severely affect the registration performance. This is especially true for registration between T1- and T2-weighted MRI volumes, where the image intensity is inverted in some places (e.g., the ventricles). To solve this problem, one can, for example, take advantage of image registration algorithms that do not depend on the image intensity itself but, rather, try to match image structures such as edges and lines (Eklund, Forsberg, Andersson & Knutsson, 2011b; Heinrich et al., 2012; Hemmendorff, Andersson, Kronander & Knutsson, 2002; Mellor & Brady, 2004, 2005; Wachinger & Navab, 2012). Another approach is to steer the registration through an initial segmentation of brain tissue types. The boundary-based registration algorithm presented by Greve and Fischl (2009) uses such a solution to more robustly register an fMRI volume to an anatomical scan. For DTI, it is possible to instead increase

the accuracy by combining several sources of information, such as a T2-weighted volume and a volume of the fractional anisotropy (Park et al., 2003). Additionally, nonlinear registration algorithms with several thousand parameters can often provide a better match between the subject's brain and a brain template, as compared with linear approaches, which optimize only a few parameters (e.g., translations and rotations).

While increasing robustness and accuracy, more advanced image registration algorithms often have a higher computational complexity. If a registration algorithm requires several hours of processing time, it does not have much practical value. Here, the GPU can once again be used to improve neuroimaging studies, by lowering the processing time to enable practical use of more robust and accurate image registration algorithms. Using GPUs to accelerate image registration is very popular. One reason is that GPUs can perform translations and rotations of images and volumes very efficiently, which is beneficial for image registration algorithms. Two recent surveys (Fluck, Vetter, Wein, Kamen, Preim & Westermann, 2011; Shams, Sadeghi, Kennedy & Hartley, 2010) mention about 50 publications on GPU accelerated image registration during the last 15 years. By using a GPU, it is not uncommon to achieve a speedup by a factor of 4–20, as compared with an optimized CPU implementation. As an example, Huang, Tang and Ju (2011) accelerated image registration within the SPM software package and obtained a speedup by a factor of 14.

Discussion

We have presented some examples of how affordable PC graphics hardware can be used to improve neuroimaging studies. The speed improvements that can be attained by analyzing fMRI data with one or several GPUs have also been documented. The main focus has been nonparametric and Bayesian methods, but we have also discussed how the spatial normalization step can be improved by taking advantage of more robust and accurate image registration algorithms. Another option is to use GPUs to explore the large space of dynamic casual models (Friston, Harrison & Penny, 2003), which can be very time consuming, or to apply nonparametric or Bayesian methods for brain connectivity analysis. An area not covered here is real-time fMRI (Cox, Jesmanowicz & Hyde, 1995; deCharms, 2008; LaConte, 2011; Weiskopf et al., 2003), where simple models and algorithms are often used to be able to process the constant stream of new data.

GPUs can clearly be used to solve a lot of problems in neuroimaging. The main challenge, as we see it, is how researchers in neuroscience and behavioral science can take advantage of GPUs without learning GPU programming. One option is to develop GPU accelerated versions of the

most commonly used software packages (e.g., SPM, FSL, AFNI), which would make it easy for the users to utilize the computational performance of GPUs. Mathworks recently introduced GPU support for the parallel computing toolbox in MATLAB. Other options for acceleration of MATLAB code include using interfaces such as Jacket⁶ or GPUmat.⁷ For the C and Fortran programming languages, the PGI accelerator model (Wolfe, 2010) or the HMPP workbench compiler (Dolbeau, Bihan & Bodin, 2007) can be used to accelerate existing code. A comparison between such frameworks has been presented by Membarth, Hannig, Teich, Korner and Eckert (2011). There is also a lot of active development of GPU packages for the Python programming language—for example, PyCUDA⁸—which are likely to be used by Python neuroimaging packages like NIPY⁹ in the near future. Recently, an interface between the statistical program R and the software packages SPM, FSL, and AFNI was developed by Boubela et al. (2012). Through this interface, preprocessing can be performed with standard established tools, while additional fMRI analysis can be accelerated with a GPU. As an example, independent component analysis was applied to 300 rest data sets from the 1,000 functional connectomes project (Biswal et al., 2010). The processing time was reduced from 16 to 1.2 h.

To conclude, using GPUs to speed up fMRI analysis that takes only a few minutes is unlikely to be worth the hassle and expense for most researchers. The true power of GPUs is that they practically enable algorithms for statistical analysis that rely on weaker assumptions. GPUs can also be used to take advantage of more robust and accurate algorithms for spatial normalization. Inexpensive PC graphics hardware can thus easily improve neuroimaging studies.

Acknowledgments Anders Eklund owns the company Wanderine Consulting, which has done consulting work for the company Accelereyes (the creators of the MATLAB GPU interface Jacket). The authors would like to thank Gerdien van Eersel for making us aware of the special issue on improved reliability and validity of neuroimaging findings.

References

- Ashburner, J., & Friston, K. J. (2000). Voxel-based morphometry – The methods. *NeuroImage*, 11(6), 805–821. doi:10.1006/nimg.2000.0582
- Aufferman, W. F., Ngan, S.-C., & Hu, X. (2002). Cluster significance testing using the bootstrap. *NeuroImage*, 17(2), 583–591. doi:10.1006/nimg.2002.1223

⁶ <http://www.accelereyes.com>

⁷ <http://sourceforge.net/projects/gpummat>

⁸ <http://mathematician.de/software/pycuda>

⁹ <http://nipy.sourceforge.net/>

- Bellec, P., Rosa-Neto, P., Lyttelton, O. C., Benali, H., & Evans, A. C. (2010). Multi-level bootstrap analysis of stable clusters in resting-state fMRI. *NeuroImage*, 51(3), 1126–1139. doi:10.1016/j.neuroimage.2010.02.082
- Bergfield, K. L., Hanson, K. D., Chen, K., Teipel, S. J., Hampel, H., Rapoport, S. I., & Alexander, G. E. (2010). Age-related networks of regional covariance in MRI gray matter: Reproducible multivariate patterns in healthy aging. *NeuroImage*, 49(2), 1750–1759. doi:10.1016/j.neuroimage.2009.09.051
- Biswal, B. B., Taylor, P. A., & Ulmer, J. L. (2001). Use of jackknife resampling techniques to estimate the confidence intervals of fMRI parameters. *Journal of Computer Assisted Tomography*, 25(1), 113–120.
- Biswal, B. B., Mennes, M., Zuo, X. N., Gohel, S., Kelly, C., Smith, S. M., & Milham, M. P. (2010). Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences of the United States of America*, 107(10), 4734–4739. doi:10.1073/pnas.0911855107
- Björnsdotter, M., Rylander, K., & Wessberg, J. (2011). A Monte Carlo method for locally multivariate brain mapping. *NeuroImage*, 56(2), 508–516. doi:10.1016/j.neuroimage.2010.07.044
- Bookstein, F. L. (2001). “Voxel-based morphometry” should not be used with imperfectly registered images. *NeuroImage*, 14(6), 1454–1462. doi:10.1006/nimg.2001.0770
- Boubela, R. N., Huf, W., Kalcher, K., Sladky, R., Filzmoser, P., Pezawas, L., & Moser, E. (2012). A highly parallelized framework for computationally intensive MR data analysis. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 25(4), 313–320. doi:10.1007/s10334-011-0290-7
- Brammer, M. J., Bullmore, E. T., Simmons, A., Williams, S. C. R., Grasby, P. M., Howard, R. J., & Rabe-Hesketh, S. (1997). Generic brain activation mapping in functional magnetic resonance imaging: A nonparametric approach. *Magnetic Resonance Imaging*, 15(7), 763–770. doi:10.1016/S0730-725X(97)00135-5
- Brett, M., Johnsrude, I. S., & Owen, A. M. (2002). The problem of functional localization in the human brain. *Nature Reviews Neuroscience*, 3, 243–249. doi:10.1038/nrn756
- Bullmore, E. T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., & Brammer, M. J. (1999). Global, voxel, and cluster tests, by theory and permutation, for a difference between two groups of structural MR images of the brain. *IEEE Transactions on Medical Imaging*, 18(1), 32–42. doi:10.1109/42.750253
- Bullmore, E., Long, C., Suckling, J., Fadili, J., Calvert, G., Zelaya, F., & Brammer, M. (2001). Colored noise and computational inference in neurophysiological (fMRI) time series analysis: Resampling methods in time and wavelet domains. *Human Brain Mapping*, 12(2), 61–78. doi:10.1002/1097-0193(200102)12:2<61::AID-HBM1004>3.0.CO;2-W
- Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J. W., & Skadron, K. (2008). A performance study of general-purpose applications on graphics processors using CUDA. *Journal of Parallel and Distributed Computing*, 68(10), 1370–1380. doi:10.1016/j.jpdc.2008.05.014
- Chib, S., & Jeliazkov, J. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, 96(453), 270–281. doi:10.1198/016214501750332848
- Chung, S., Pelletier, D., Sdika, M., Lu, Y., Berman, J. I., & Henry, R. G. (2008). Whole brain voxel-wise analysis of single-subject serial DTI by permutation testing. *NeuroImage*, 39(4), 1693–1705. doi:10.1016/j.neuroimage.2007.10.039
- Cox, R. W., Jesmanowicz, A., & Hyde, J. S. (1995). Real-time functional magnetic resonance imaging. *Magnetic Resonance in Medicine*, 33(2), 230–236. doi:10.1002/mrm.1910330213
- Cubon, V. A., Putukian, M., Boyer, C., & Dettwiler, A. (2011). A diffusion tensor imaging study on the white matter skeleton in individuals with sports-related concussion. *Journal of Neurotrauma*, 28(2), 189–201. doi:10.1089/neu.2010.1430

- deCharms, R. C. (2008). Applications of real-time fMRI. *Nature Reviews Neuroscience*, 9, 720–729. doi:10.1038/nrn2414
- Dolbeau, R., Bihan S., & Bodin, F. (2007). HMPP: A hybrid multi-core parallel programming environment. *Proceedings of the Workshop on general-purpose processing on graphics processing units*
- Dwass, M. (1957). Modified randomization tests for nonparametric hypotheses. *Annals of Mathematical Statistics*, 28(1), 181–187. doi:10.1214/aoms/1177707045
- Eklund, A., Andersson, M., & Knutsson, H. (2010). Phase based volume registration using CUDA. *International conference on acoustics, speech and signal processing (ICASSP)*, 658–651. doi:10.1109/ICASSP.2010.5495134
- Eklund, A., Andersson, M., & Knutsson, H. (2011a). Fast random permutation tests enable objective evaluation of methods for single-subject fMRI analysis. *International Journal of Biomedical Imaging*. doi:10.1155/2011/627947. Article ID 627947.
- Eklund, A., Andersson, M., & Knutsson, H. (2012a). fMRI analysis on the GPU – possibilities and challenges. *Computer Methods and Programs in Biomedicine*, 105(2), 145–161. doi:10.1016/j.cmpb.2011.07.007
- Eklund, A., Dufort, P., Forsberg, D., & LaConte, S. M. (2012b). *Medical image processing on the GPU—Past, present and future*. Manuscript submitted for publication.
- Eklund, A., Björnsdotter, M., Stelzer, J., & LaConte, S.M. (2013). Searchlight goes GPU – Fast multi-voxel pattern analysis of fMRI data. *International society for magnetic resonance in medicine (ISMRM)*
- Eklund, A., Forsberg, D., Andersson, M., & Knutsson, H. (2011b). Using the local phase of the magnitude of the local structure tensor for image registration. *Lecture notes in computer science, Scandinavian conference on image analysis (SCLIA)*, 6688, 414–423. doi:10.1007/978-3-642-12227-7_39
- Eklund, A., Andersson, M., Josephson, C., Johannesson, M., & Knutsson, H. (2012c). Does parametric fMRI analysis with SPM yield valid results? - An empirical study of 1484 rest datasets. *NeuroImage*, 61(3), 565–578. doi:10.1016/j.neuroimage.2012.03.093
- Feinberg, D. A., & Yacoub, E. (2012). The rapid development of high speed, resolution and precision in fMRI. *NeuroImage*, 62(2), 720–725. doi:10.1016/j.neuroimage.2012.01.049
- Ferreira da Silva, A. R. (2011a). A Bayesian multilevel model for fMRI data analysis. *Computer Methods and Programs in Biomedicine*, 102(3), 238–252. doi:10.1016/j.cmpb.2010.05.003
- Ferreira da Silva, A. R. (2011b). cudaBayesreg: Parallel implementation of a Bayesian multilevel model for fMRI data analysis. *Journal of Statistical Software*, 44(4), 1–24.
- Fluck, O., Vetter, C., Wein, W., Kamen, A., Preim, B., & Westermann, R. (2011). A survey of medical image registration on graphics hardware. *Computer Methods and Programs in Biomedicine*, 104(3), e45–e57. doi:10.1016/j.cmpb.2010.10.009
- Friman, O., Borga, M., Lundberg, P., & Knutsson, H. (2003). Adaptive analysis of fMRI data. *NeuroImage*, 19(3), 837–845. doi:10.1016/S1053-8119(03)00077-6
- Friston, K. J., Harrison, L., & Penny, W. (2003). Dynamic casual modelling. *NeuroImage*, 19(4), 1273–1302. doi:10.1016/S1053-8119(03)00202-7
- Friston, K. J., Holmes, A. P., & Worsley, K. J. (1999). How many subjects constitute a study? *NeuroImage*, 10(1), 1–5. doi:10.1006/nimg.1999.0439
- Friston, K. J., Ashburner, J., Frith, C. D., Poline, J. B., Heather, J. D., & Frackowiak, R. S. J. (1995a). Spatial registration and normalization of images. *Human Brain Mapping*, 3(3), 165–189. doi:10.1002/hbm.460030303
- Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J. P., Frith, C. D., & Frackowiak, R. S. J. (1995b). Statistical parametric maps in functional imaging: A general linear approach. *Human Brain Mapping*, 2(4), 189–210. doi:10.1002/hbm.460020402
- Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G., & Ashburner, J. (2002). Classical and Bayesian inference in neuroimaging: Theory. *NeuroImage*, 16(2), 465–483. doi:10.1006/nimg.2002.1090
- Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., ... Volkov, V. (2008). Parallel computing experiences with CUDA. *IEEE Micro*, 28(4), 13–27. doi:10.1109/MM.2008.57
- Genovese, C. R. (2000). A Bayesian time-course model for functional magnetic resonance imaging data. *Journal of the American Statistical Association*, 95(451), 691–703. doi:10.1080/01621459.2000.10474253
- Greve, D. N., & Fischl, B. (2009). Accurate and robust brain image segmentation using boundary-based registration. *NeuroImage*, 48(1), 63–72. doi:10.1016/j.neuroimage.2009.06.060
- Grigis, A., Noblet, V., Heitz, F., Blanc, F., de Seze, F., Kremer, S., & Armspach, J.-P. (2012). Longitudinal change detection in diffusion MRI using multivariate statistical testing on tensors. *NeuroImage*, 60(4), 2206–2221. doi:10.1016/j.neuroimage.2012.02.049
- Gudbjartsson, H., & Patz, S. (1995). The Rician distribution of noisy MRI data. *Magnetic Resonance in Medicine*, 34(6), 910–914. doi:10.1002/mrm.1910340618
- Guo, G. (2012). Parallel statistical computing for statistical inference. *Journal of Statistical Theory and Practice*, 6, 536–565. doi:10.1080/15598608.2012.695705
- Gössi, C., Fahrmeir, L., & Auer, D. P. (2001). Bayesian modeling of the hemodynamic response function in BOLD fMRI. *NeuroImage*, 14(1), 140–148. doi:10.1006/nimg.2001.0795
- Habeck, C., & Stern, Y. (2010). Multivariate data analysis for neuroimaging data: Overview and application to Alzheimer's disease. *Cell Biochemistry and Biophysics*, 58(2), 53–67. doi:10.1007/s12013-010-9093-0
- Heinrich, M. P., Jenkinson, M., Bhushan, M., Matin, T., Gleeson, F. V., Brady, M., & Schnabel, J. A. (2012). MIND: Modality independent neighbourhood descriptor for multi-modal deformable registration. *Medical Image Analysis*, 16(7), 1423–1435. doi:10.1016/j.media.2012.05.008
- Hemmenndorff, M., Andersson, M. T., Kronander, T., & Knutsson, H. (2002). Phase-based multidimensional volume registration. *IEEE Transactions on Medical Imaging*, 21(12), 1536–1543. doi:10.1109/TMI.2002.806581
- Hernandez, M., Guerrero, G.D., Cecilia, J.M., Garcia, J.M., Inuggi, A., & Sotiropoulos, S.N. (2012). Accelerating fibre orientation estimation from diffusion weighted resonance imaging using GPUs. *Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 622–626. doi:10.1109/PDP.2012.46
- Holmes, A. P., Blair, R. C., Watson, J. D. G., & Ford, I. (1996). Nonparametric Analysis of Statistic Images from Functional Mapping Experiments. *Journal of Cerebral Blood Flow & Metabolism*, 16, 7–22. doi:10.1097/00004647-199601000-00002
- Huang, T., Tang, Y., & Ju, S. (2011). Accelerating image registration of MRI by GPU-based parallel computation. *Magnetic Resonance Imaging*, 29(5), 712–716. doi:10.1016/j.mri.2011.02.027
- Jones, D. K., & Cercignani, M. (2010). Twenty-five pitfalls in the analysis of diffusion MRI data. *NMR in Biomedicine*, 23(7), 803–820. doi:10.1002/nbm.1543
- Jones, D. K., & Pierpaoli, C. (2005). Confidence mapping in diffusion tensor magnetic resonance imaging tractography using a bootstrap approach. *Magnetic resonance in medicine*, 53(5), 1143–1149. doi:10.1002/mrm.20466
- Jones, D. K., Griffin, L. D., Alexander, D. C., Catani, M., Horsfield, M. A., Howard, R., & Williams, S. C. R. (2002). Spatial normalization and averaging of diffusion tensor MRI data sets. *NeuroImage*, 17(7), 592–617. doi:10.1006/nimg.2002.1148
- Kawasaki, Y., Suzuki, M., Kherif, F., Takahashi, T., Zhou, S.-Y., Nakamura, K., & Kurachi, M. (2007). Multivariate voxel-based morphometry successfully differentiates schizophrenia patients

- from healthy controls. *NeuroImage*, 34(1), 235–242. doi:10.1016/j.neuroimage.2006.08.018
- Kimberg, D. Y., Coslett, H. B., & Schwartz, M. F. (2007). Power in voxel-based lesion-symptom mapping. *Journal of Cognitive Neuroscience*, 19(7), 1067–1080. doi:10.1162/jocn.2007.19.7.1067
- Kirk, D. B., & Hwu, W. W. (2010). *Programming massively parallel processors: A hands-on approach*. Morgan Kaufmann
- Kriegeskorte, N., Goebel, R., & Bandettini, P. (2006). Information based functional brain mapping. *Proceedings of the National Academy of Sciences*, 103(10), 3863–3868. doi:10.1073/pnas.0600244103
- LaConte, S. M., Strother, S., Cherkassky, V., Anderson, J., & Hu, X. (2005). Support vector machines for temporal classification of block design fMRI data. *NeuroImage*, 26(2), 317–329. doi:10.1016/j.neuroimage.2005.01.048
- LaConte, S. M. (2011). Decoding fMRI brain states in real-time. *NeuroImage*, 56(2), 440–454. doi:10.1016/j.neuroimage.2010.06.052
- Lazar, M., & Alexander, A. L. (2005). Bootstrap white matter tractography (BOOT-TRAC). *NeuroImage*, 24(2), 524–532. doi:10.1016/j.neuroimage.2004.08.050
- Lee, A., Yau, C., Giles, M. B., Doucet, A., & Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of computational and graphical statistics*, 19(4), 769–789. doi:10.1198/jcgs.2010.10039
- McGraw, T., & Nadar, M. (2007). Stochastic DT-MRI connectivity mapping on the GPU. *IEEE Transactions on visualization and computer graphics*, 13(6), 1504–1511. doi:10.1109/TVCG.2007.70597
- McIntosh, A. R., Chau, W. K., & Protzner, A. B. (2004). Spatiotemporal analysis of event-related fMRI data using partial least squares. *NeuroImage*, 23(2), 764–775. doi:10.1016/j.neuroimage.2004.05.018
- Mellor, M., & Brady, M. (2004). Non-rigid multimodal image registration using local phase. *Lecture Notes in Computer Science: Vol 3216, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 789–796. doi:10.1007/978-3-540-30135-6_96
- Mellor, M., & Brady, M. (2005). Phase mutual information as similarity measure for registration. *Medical Image Analysis*, 9(4), 330–343. doi:10.1016/j.media.2005.01.002
- Membarth, R., Hannig, F., Teich, J., Komer, M., & Eckert, W. (2011). Frameworks for GPU accelerators: A comprehensive evaluation using 2D/3D image registration. *IEEE Symposium on Application specific processors (SASP)*, 78–71. doi:10.1109/SASP.2011.5941083
- Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., & Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, 57(1–2), 145–175. doi:10.1023/B:MACH.0000035475.85309.1b
- Nandy, R. R., & Cordes, D. (2003). Novel nonparametric approach to canonical correlation analysis with applications to low CNR functional MRI data. *Magnetic Resonance in Medicine*, 50(2), 354–365. doi:10.1002/mrm.10537
- Nandy, R., & Cordes, D. (2007). A semi-parametric approach to estimate the family-wise error rate in fMRI using resting-state data. *NeuroImage*, 34(4), 1562–1576. doi:10.1016/j.neuroimage.2006.10.025
- Nichols, T. E., & Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1), 1–25. doi:10.1002/hbm.1058
- Nichols, T. E., & Hayasaka, S. (2003). Controlling the familywise error rate in functional neuroimaging: A comparative review. *Statistical Methods in Medical Research*, 12(5), 419–446. doi:10.1191/0962280203sm341ra
- Nieto-Castanon, A., Ghosh, S. S., Tourville, J. A., & Guenther, F. H. (2003). Region of interest based analysis of functional imaging data. *NeuroImage*, 19(4), 1303–1316. doi:10.1016/S1053-8119(03)00188-5
- Norman, K. A., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: Multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, 10(9), 424–430. doi:10.1016/j.tics.2006.07.005
- Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Kruger, J., Lefohn, A. E., & Purcell, T. J. (2007). A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1), 80–113. doi:10.1111/j.1467-8659.2007.01012.x
- Park, H.-J., Kubicki, M., Shenton, M. E., Guimond, A., McCarley, R. W., Maier, S. E., & Westin, C.-F. (2003). Spatial normalization of diffusion tensor MRI using multiple channels. *NeuroImage*, 20(4), 1995–2009. doi:10.1016/j.neuroimage.2003.08.008
- Penny, W., Kiebel, S., & Friston, K. J. (2003). Variational Bayesian inference for fMRI time series. *NeuroImage*, 19(3), 727–741. doi:10.1016/S1053-8119(03)00071-5
- Penny, W., Trujillo-Barreto, N. J., & Friston, K. J. (2005). Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2), 350–362. doi:10.1016/j.neuroimage.2004.08.034
- Roland, P. E., Geyer, S., Amunts, K., Schormann, T., Schleicher, A., Malikovic, A., & Zilles, K. (1997). Cytoarchitectural maps of the human brain in standard anatomical space. *Human Brain Mapping*, 5(4), 222–227. doi:10.1002/(SICI)1097-0193(1997)5:4<222::AID-HBM3>3.0.CO;2-5
- Rugg-Gunn, F. J., Eriksson, S. H., Symms, M. R., Barker, G. J., & Duncan, J. S. (2001). Diffusion tensor imaging of cryptogenic and acquired partial epilepsies. *Brain*, 124(3), 627–636. doi:10.1093/brain/124.3.627
- Sanders, J., & Kandrot, E. (2010). *CUDA by example: An introduction to General-Purpose GPU Programming*. Addison-Wesley Professional
- Scarpazza, C., Sartori, G., De Simone, M. S., & Mechelli, A. (2013). When the single subject matters more than the group: Very high false positive rates in single case voxel based morphometry. *NeuroImage*, 70, 175–188. doi:10.1016/j.neuroimage.2012.12.045
- Shams, R., Sadeghi, P., Kennedy, R., & Hartley, R. (2010). A survey of medical image registration on multicore and the GPU. *IEEE Signal Processing Magazine*, 27(2), 50–60. doi:10.1109/MSP.2009.935387
- Shterev, I. D., Jung, S.-H., George, S. L., & Owzar, K. (2010). permGPU: Using graphics processing units in RNA microarray association studies. *BMC Bioinformatics*, 11, 329. doi:10.1186/1471-2105-11-329
- Silver, M., Montana, G., & Nichols, T. E. (2011). False positives in neuroimaging genetics using voxel-based morphometry data. *NeuroImage*, 54(2), 992–1000. doi:10.1016/j.neuroimage.2010.08.049
- Smith, A. M., Lewis, B. K., Ruttimann, U. E., Ye, F. Q., Sinnwell, T. M., Yang, Y., & Frank, J. A. (1999). Investigation of low frequency drift in fMRI signal. *NeuroImage*, 9(5), 526–533. doi:10.1006/nimg.1999.0435
- Smith, S. M., Jenkinson, M., Johansen-Berg, H., Rueckert, D., Nichols, T. E., Mackay, C. E., & Behrens, T. E. J. (2006). Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. *NeuroImage*, 31(4), 1487–1505. doi:10.1016/j.neuroimage.2006.02.024
- Stelzer, J., Chen, Y., & Turner, R. (2013). Statistical inference and multiple testing correction in classification-based multi-voxel pattern analysis (MVPA): Random permutations and cluster size control. *NeuroImage*, 65, 69–82. doi:10.1016/j.neuroimage.2012.09.063
- Suchard, M. A., Wang, Q., Chan, C., Frelinger, J., Cron, A., & West, M. (2010). Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 19(2), 419–438. doi:10.1198/jcgs.2010.10016
- Thirion, B., Flandin, G., Pinel, P., Roche, A., Ciuciu, P., & Poline, J.-B. (2006). Dealing with the shortcomings of spatial normalization: Multi-subject parcellation of fMRI datasets. *Human brain mapping*, 27(8), 678–693. doi:10.1002/hbm.20210
- Thomas, A. G., Marrett, S., Saad, Z. S., Ruff, D. A., Martin, A., & Bandettini, P. A. (2009). Functional but not structural changes

- associated with learning: An exploration of longitudinal Voxel-Based Morphometry (VBM). *NeuroImage*, 48(1), 117–125. doi:10.1016/j.neuroimage.2009.05.097
- van Hemert, J. L., & Dickerson, J. A. (2011). Monte Carlo randomization tests for large-scale abundance datasets on the GPU. *Computer Methods and Programs in Biomedicine*, 101(1), 80–86. doi:10.1016/j.cmpb.2010.04.010
- Wachinger, C., & Navab, N. (2012). Entropy and Laplacian images: Structural representations for multi-modal registration. *Medical Image Analysis*, 16(1), 1–17. doi:10.1016/j.media.2011.03.001
- Weiskopf, N., Veit, R., Erb, M., Mathiak, K., Grodd, W., Goebel, R., & Birbaumer, N. (2003). Physiological self regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI): methodology and exemplary data. *NeuroImage*, 19(3), 577–586. doi:10.1016/S1053-8119(03)00145-9
- Wilke, M. (2012). An iterative jackknife approach for assessing reliability and power of fMRI group analyses. *PLoS ONE*, 7(4), e35578. doi:10.1371/journal.pone.0035578
- Wolfe, M. (2010). Implementing the PGI accelerator model. *Proceedings of the workshop on general-purpose computation on graphics processing units*, 43–50. doi:10.1145/1735688.1735697
- Woolrich, M. W. (2012). Bayesian inference in fMRI. *NeuroImage*, 62(2), 801–810. doi:10.1016/j.neuroimage.2011.10.047
- Woolrich, M. W., Jenkinson, M., Brady, J. M., & Smith, S. M. (2004). Fully Bayesian spatio-temporal modeling of FMRI data. *IEEE Transactions on Medical Imaging*, 23(2), 213–231. doi:10.1109/TMI.2003.823065
- Worsley, K. J., Marrett, S., Neelin, P., & Evans, A. C. (1992). A three-dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, 12(6), 900–918. doi:10.1038/jcbfm.1992.127
- Zhu, H., Ibrahim, J. G., Tang, N., Rowe, D. B., Hao, X., Bansal, R., & Peterson, B. S. (2007). A statistical analysis of brain morphology using wild bootstrapping. *IEEE Transactions on Medical Imaging*, 26(7), 954–966. doi:10.1109/TMI.2007.897396