



The Insight ToolKit image registration framework

Brian B. Avants^{1*}, Nicholas J. Tustison², Michael Stauffer¹, Gang Song¹, Baohua Wu¹ and James C. Gee¹

¹ Penn Image Computing and Science Laboratory, Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA

² Department of Radiology and Medical Imaging, University of Virginia, Charlottesville, VA, USA

Edited by:

Hans J. Johnson, The University of Iowa, USA

Reviewed by:

Marcel Prastawa, University of Utah, USA

Andrey Fedorov, Brigham and Women's Hospital, USA

*Correspondence:

Brian B. Avants, Department of Radiology, University of Pennsylvania, 3600 Market St., Suite 370, Philadelphia, PA 19104, USA
e-mail: avants@grasp.cis.upenn.edu

Publicly available scientific resources help establish evaluation standards, provide a platform for teaching and improve reproducibility. Version 4 of the Insight ToolKit (ITK⁴) seeks to establish new standards in publicly available image registration methodology. ITK⁴ makes several advances in comparison to previous versions of ITK. ITK⁴ supports both multivariate images and objective functions; it also unifies high-dimensional (deformation field) and low-dimensional (affine) transformations with metrics that are reusable across transform types and with composite transforms that allow arbitrary series of geometric mappings to be chained together seamlessly. Metrics and optimizers take advantage of multi-core resources, when available. Furthermore, ITK⁴ reduces the parameter optimization burden via principled heuristics that automatically set scaling across disparate parameter types (rotations vs. translations). A related approach also constrains steps sizes for gradient-based optimizers. The result is that tuning for different metrics and/or image pairs is rarely necessary allowing the researcher to more easily focus on design/comparison of registration strategies. In total, the ITK⁴ contribution is intended as a structure to support reproducible research practices, will provide a more extensive foundation against which to evaluate new work in image registration and also enable application level programmers a broad suite of tools on which to build. Finally, we contextualize this work with a reference registration evaluation study with application to pediatric brain labeling.¹

Keywords: registration, MRI, brain, open-source, death

1. INTRODUCTION

As image registration methods mature—and their capabilities become more widely recognized—the number of applications increase (Rueckert et al., 1999; van Dalen et al., 2004; Miller et al., 2005; Shelton et al., 2005; Chen et al., 2008; Baloch and Davatzikos, 2009; Cheung and Krishnan, 2009; Peyrat et al., 2010; Fedorov et al., 2011; Kikinis and Pieper, 2011; Metz et al., 2011; Murphy et al., 2011). Consequently, image registration transitioned from being a field of active research, and few applied results, to a field where the main focus is translational. Image registration is now used to derive quantitative biomarkers from images (Jack et al., 2010), plays a major role in business models and clinical products (especially in radiation oncology) (Cheung and Krishnan, 2009), has led to numerous new findings in studies of brain and behavior (e.g., Bearden et al., 2007) and is a critical component in applications in pathology, microscopy, surgical planning, and more (Miller et al., 2005; Shelton et al., 2005; Flocq and Dickhaus, 2007; Chen et al., 2008; Cheung and Krishnan, 2009; Peyrat et al., 2010; Kikinis and Pieper, 2011; Murphy et al., 2011). Despite the increasing relevance of image registration across application domains, there are relatively few reference algorithm implementations available to the community. Furthermore, these

resources have become critical to setting performance standards in international challenges that evaluate “real world” registration scenarios (see, for instance, the SATA 2013 and BRATS 2013 challenges at MICCAI in Nagoya, Japan).

One source of benchmark methodology is the Insight ToolKit (ITK) (Yoo et al., 2002; Ackerman and Yoo, 2003), which marked a significant contribution to medical image processing when it first emerged at the turn of the millennium. Since that time, ITK has become a standard-bearer for image processing algorithms and, in particular, for image registration methods. In a review of ITK user interests, image registration was cited as the most important contribution of ITK (personal communication with Terry Yoo). Numerous papers use ITK algorithms as standard references for implementations of Demons registration and mutual information-based affine or B-Spline registration (van Dalen et al., 2004; Shelton et al., 2005; Flocq and Dickhaus, 2007; Chen et al., 2008; Cheung and Krishnan, 2009). Multiple toolkits extend ITK registration methods in unique ways. Elastix provides very fast and accurate B-Spline registration (Klein et al., 2010; Murphy et al., 2011). The diffeomorphic demons is a fast/efficient approximation to a diffeomorphic mapping (Vercauteren et al., 2009). ANTs provides both flexibility and high average performance (Avants et al., 2011). The BRAINSFit algorithm is integrated into Slicer for user-guided registration (Kikinis and Pieper, 2011). Each of these toolkits has both strengths and weaknesses (Klein et al., 2010; Murphy et al., 2011) and was enabled by an ITK core.

¹This work is supported by National Library of Medicine sponsored ARRA stimulus funding.

The Insight ToolKit began a major refactoring effort in 2010. The refactoring aimed to both simplify and extend the techniques available in version 3.x with methods and ideas from a new set of prior work (Christensen et al., 1996; Rueckert et al., 1999; Jenkinson and Smith, 2001; Miller et al., 2005; Peyrat et al., 2010; Avants et al., 2011). To make this technology more accessible, ITK⁴ unifies the dense registration framework (displacement field, diffeomorphisms) with the low-dimensional (B-Spline, affine, rigid) framework by introducing composite transforms, deformation field transforms, and specializations that allowed these to be optimized efficiently. A sub-goal set for ITK⁴ was to simplify parameter setting by adding helper methods that use well-known principles of image registration to automatically scale transform components and set optimization parameters. ITK⁴ transforms are also newly applicable to objects such as vectors and tensors and will take into account covariant geometry if necessary. Finally, ITK⁴ reconfigures the registration framework to maximize multi-threading resources where possible. The revised registration framework within ITK is more thoroughly integrated across transform models, is thread-safe and provides broader functionality than in prior releases.

David Donoho once commented (in paraphrase) that academic publications are merely “advertisements” for the real work which is constituted by the “complete instruction set” that produces the results reported in the publication (Buckheit and Donoho, 1995). The first part of the remainder of this document will provide an “advertisement” for the ITK framework and summarize its evolution from ITK³ to ITK⁴. We then detail potential applications of this ITK⁴ framework in the context of a general nomenclature. While this work is indeed incomplete, in the sense of Donoho, we refer to source code and data when relevant. Furthermore, section 3.1 shows a series of reproducible examples of ITK⁴ in action. Several areas relevant to neuroinformatics are highlighted in these examples: optimal template construction, “challenging” registration scenarios involving brain mapping in the presence of lesions or resection, registration when initialization priors are weak, asymmetry analyses, functional MRI, and non-traditional registration strategies are all highlighted. We also establish performance benchmarks for the current ITK⁴ registration, in comparison to a method developed for ITK³, via a standard brain labeling task. Finally, we discuss future developments in the framework.

2. MATERIALS AND METHODS

2.1. OVERVIEW OF THE UNIFIED FRAMEWORK

The overall purpose of the registration refactoring for ITK⁴ was to simplify the user experience and to accelerate and improve performance. Here, we summarize how ITK⁴ works toward these goals.

2.1.1. Core Software Components

Figure 1 sketches the ITK⁴ architecture at a high level. Registration applications are known as “registration methods” as they were in ITK³. The methods, with source contained in ITK⁴’s `RegistrationMethodsv4` directory, hold together the different subcomponents that make a working instantiation of a registration strategy. These subcomponents include the

optimization technique (in the `Optimizersv4` directory), the metric measuring the registration quality² (the `Metricsv4` directory), the images or other data objects that enter the metric and the parameters that are being optimized. The parameters are usually defined by a geometric transformation but may point to other relevant objects. Any of ITK⁴’s transformations may be optimized by the framework. New transformations, relative to ITK³, include the `DisplacementField` transforms that are useful for engendering Demons or B-Spline registration strategies. New `VelocityField` transforms are also available. A typical application developer would employ all of these components. A good starting point for new users who wish to see how these tools work together, in source code, is found in the tests. For instance, see the files `itkTimeVaryingBSplineVelocityFieldImageRegistrationTest.cxx` for an example of a B-Spline diffeomorphism application, `itkSyNImageRegistrationTest.cxx` to see SyN in ITK⁴ and `itkSimpleImageRegistrationTest2.cxx` for a more basic example.

Several usability goals spurred ITK⁴ development. We summarize these here.

2.1.2. Image registration should be achievable in one step

This overarching goal is best illustrated by `RegistrationMethodsv4` in which a user may string together a series of registration tools to perform (for instance) a translation registration, followed by an affine registration, followed by a diffeomorphic mapping each of which might use a different image similarity metric. The different transforms are accumulated in the new `itkCompositeTransform` which chains transforms together as in Figure 2. Thus, this framework provides unprecedented ability to perform complex and staged registration mappings. Furthermore, the framework’s automated parameter scaling, `itkRegistrationParameterScalesEstimator`, vastly reduces the difficulty of tuning parameters for different transform/metric combinations.

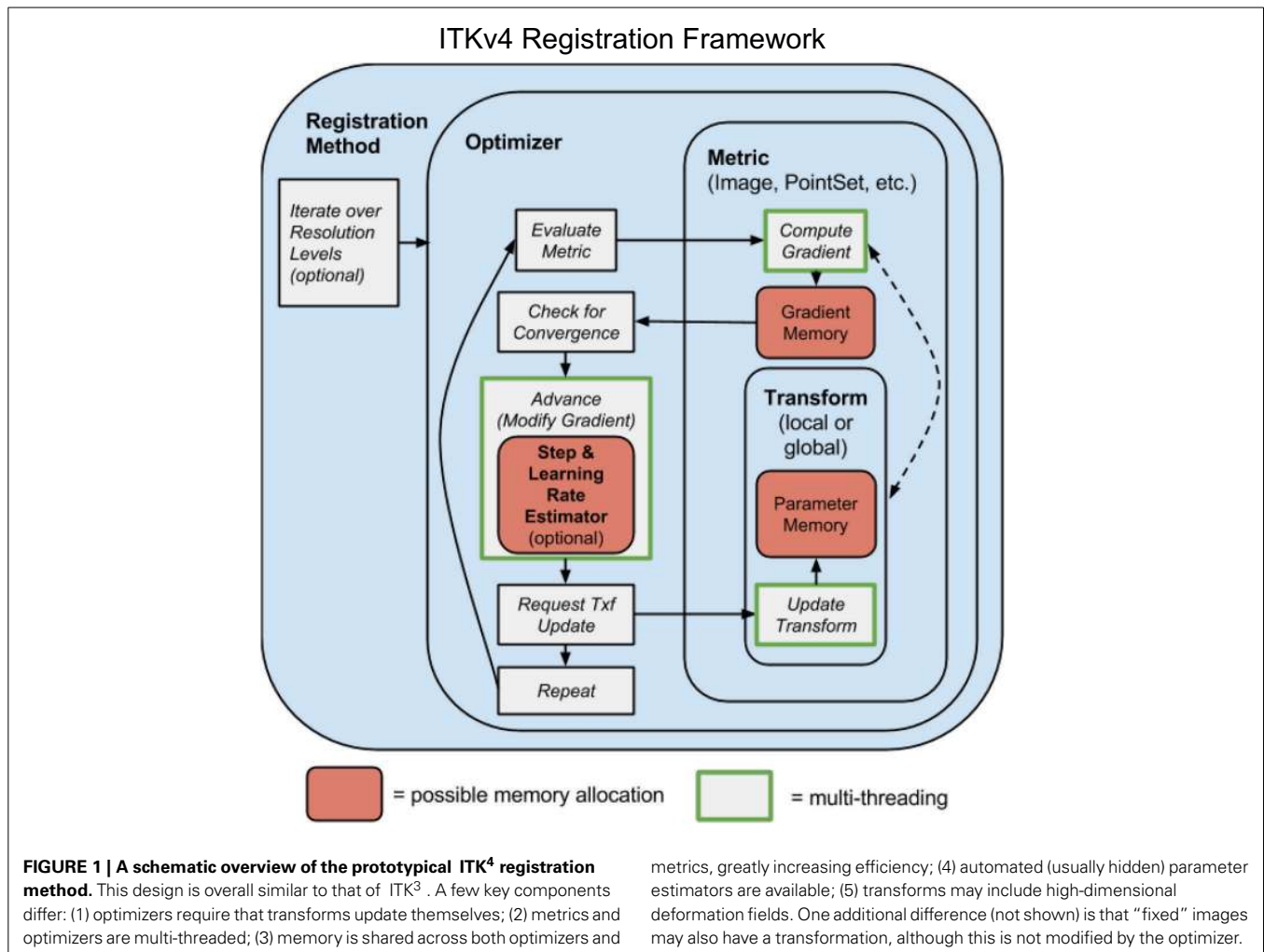
2.1.3. ITK Transforms should be unified

Each ITK⁴ transform now has either global support (affine transform) or local (or compact) support (a displacement field transform). If any map in a composite transform has global support then the composite transform has global support. Both “fixed” and “moving” images may have initial transforms. This allows one to reduce “registration bias” that may be induced by asymmetric interpolation (Yushkevich et al., 2010).

2.1.4. Registration mappings should be applicable to a number of popular data types, including DTI

Our revisions to the ITK³ transform hierarchy validated and extended the ITK³ transforms for thread safety and applicability to not only vectors but also tensors. Reorientation steps necessary for diffusion tensor mappings are now included in ITK⁴.

²All ITK⁴ metrics are set to be minimized. For instance, the `itkMattesMutualInformationImageToImageMetricv4` returns negative mutual information.



metrics, greatly increasing efficiency; (4) automated (usually hidden) parameter estimators are available; (5) transforms may include high-dimensional deformation fields. One additional difference (not shown) is that “fixed” images may also have a transformation, although this is not modified by the optimizer.

2.1.5. Affine and deformable similarity metrics should look as similar as possible

The Metricsv4 framework supports this goal in that it is as trivial to implement a mutual information Demons algorithm as it is to implement a sum of squared differences B-spline or affine registration algorithm. Thus, full plug-and-play support exists across transforms.

2.1.6. Users should be able to combine multiple similarity metrics, some of which may operate on different data types

This is achievable with the existing `itkMultiGradientOptimizerv4` through the multivariate `itkObjectToObjectMultiMetricv4` or through the multi-channel traits (`itkVectorImageToImageMetricTraitsv4`) that allow metrics to deal with multi-channel pixels, all of which were contributed for ITK⁴. The `itkObjectToObjectMultiMetricv4` was used in our winning entry of the SATA 2013 “dog leg” challenge.

2.1.7. Optimizers and transformations should interact flexibly

Optimizersv4 includes optimizers that are applicable to both linear and deformable transformations, which include convergence monitoring and enable 2nd order optimization

(`itkQuasiNewtonOptimizerv4`), multiple objective optimization (`itkMultiGradientOptimizerv4`), or global optimization (`itkMultiStartOptimizerv4`).

2.1.8. GPU and multi-core acceleration will open up new applications for image registration

See `GPUPDEdeformable` for a GPU example. Furthermore, the new metric framework N cores to accelerate metric, gradient and optimization steps. A recent real-world application of the new Insight ToolKit implementation of the symmetric normalization algorithm showed a speed-up of almost a factor of six when comparing single core to eight core execution time. This speed-up is achieved by multi-threading the similarity metric, the gradient descent update, the regularization terms and the composition components of the method. Thus, every essential step exploits intrinsic parallelism in the algorithm. Decreased execution time means more rapid turnaround for users, faster turn-around in testing and higher throughput on large-scale computing tasks.

2.1.9. Improve memory efficiency in optimization framework

Memory optimizations are critical for efficient use of large local transforms. In ITK⁴, transform parameters are no longer copied within the optimizer, but rather left in-place in transform. Metric

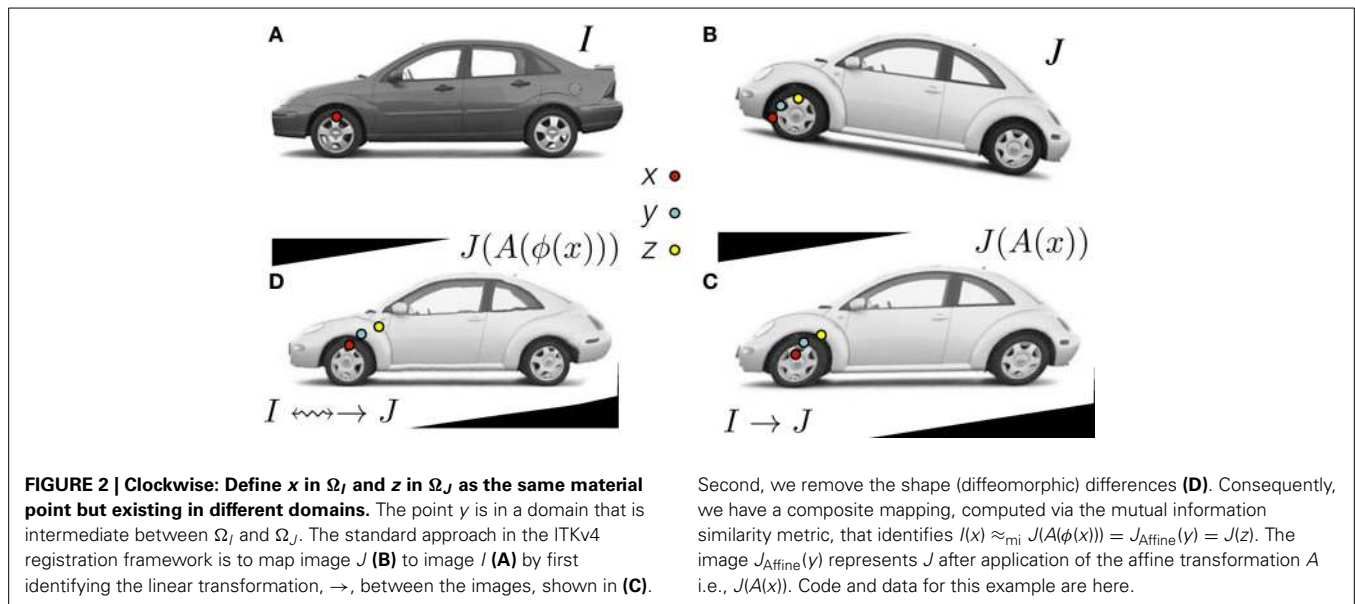


FIGURE 2 | Clockwise: Define x in Ω_I and z in Ω_J as the same material point but existing in different domains. The point y is in a domain that is intermediate between Ω_I and Ω_J . The standard approach in the ITKv4 registration framework is to map image J (B) to image I (A) by first identifying the linear transformation, \rightarrow , between the images, shown in (C).

Second, we remove the shape (diffeomorphic) differences (D). Consequently, we have a composite mapping, computed via the mutual information similarity metric, that identifies $I(x) \approx_{mi} J(A(\phi(x))) = J_{Affine}(y) = J(z)$. The image $J_{Affine}(y)$ represents J after application of the affine transformation A i.e., $J(A(x))$. Code and data for this example are here.

gradient memory is shared between optimizer and metric, and modifications by the optimizer are done in place when possible.

Finally, we summarize ITK⁴ changes through quantitative metrics:

- Over 12 new multi-threaded image registration metrics are available in v4.
- Four application-level registration methods, with plug-and-play architecture, are available for high-level inclusion in projects such as Slicer and SimpleITK.
- All contributions are unit-tested and have greater than 85% code coverage, in accordance with ITK standards.
- A complete refactoring of the ITK transform hierarchy that makes transforms thread-safe, applicable to high-dimensional optimization and easily used in multi-core computing. Forty-one classes, in total, were impacted by this refactoring.
- We added transparent vector support to two key interpolators that are used pervasively in ITK: the nearest neighbor and linear interpolators. We added two new Gaussian interpolators.
- An example of vector support for image metrics is in `itkMeanSquaresImageToImageMetricv4VectorRegistrationTest.cxx`.

Below we will discuss: (0) an organizing nomenclature matched to the ITK⁴ framework, (1) gradient-based optimization within the framework, (2) techniques to estimate optimization parameters for arbitrary metric and transformation combinations, (3) a ITK⁴ instance implementing generalized diffeomorphic matching, (4) several applications of the updated framework within different neuroinformatics-relevant domains.

2.2. NOMENCLATURE

The nomenclature below designates an image registration algorithm symbolically. This nomenclature is intended to be a descriptive and technically consistent system for visually

begin nomenclature definition

A physical point:	$x \in \Omega$ where Ω is the domain, usually of an image.
An image:	$I: \Omega^d \rightarrow \mathbb{R}^n$ where n is the number of components per pixel and d is dimensionality. A second image is J .
Domain map:	$\phi: \Omega_I \rightarrow \Omega_J$ where \rightarrow may be replaced with any mapping symbol below.
Affine mapping:	\leftrightarrow a low-dimensional invertible transformation: affine, rigid, translation, etc.
Affine mapping:	\rightarrow designates the direction an affine mapping is applied.
Deformation field:	\rightsquigarrow deformation field mapping J to I . May not be invertible.
Spline-based mapping:	$\tilde{\rightsquigarrow}$ e.g., B-Spline field mapping J to I .
Diffeomorphism:	Represented as \rightsquigarrow , these are differentiable maps with differentiable inverse. Ideally, the algorithm should output the inverse and forward mapping.
Composite mapping:	$\phi = \phi_1(\phi_2(x))$ is defined by $\rightsquigarrow \rightarrow$ where ϕ_2 is of type \rightsquigarrow .
Not invertible:	\leftrightarrow indicates a mapping that is not invertible.
Perform image warping:	As an example, $\rightarrow J$ represents the application of an affine transform \rightarrow to image J such that $\rightarrow J = J(A(x))$.
Similarity measure:	\approx_s or \approx_{ms} indicates the metric s that compares a pair of images.

end nomenclature definition

representing algorithms and applications of registration. Ideally, any standard algorithm can be written in the nomenclature below.

We would then write a standard Demons registration application that maps one image, J , into the space of I (presumably a template) as:

$$I \rightsquigarrow \rightarrow J \quad \text{which symbolizes} \quad I \approx J(A(\phi(x))),$$

with A an affine mapping and ϕ a generic deformation. The notation means that the algorithm first optimizes an affine mapping, \rightarrow , between J and I . This is followed by a deformation in the second stage, \rightsquigarrow , from $\rightarrow J$ to I . In terms of transformation composition, we would write $\rightsquigarrow \rightarrow J = J_w(x) = J(\phi_{\text{Affine}}(\phi_{\text{Demons}}(x)))$ where J_w is the result of warping J to I . The ϕ are the specific functions corresponding to the schematic arrows. Note, also, that the tail of the arrow indicates the transform's domain. The arrow-head indicates its range. Finally, we denote the similarity metric as \approx which indicates a sum of squared differences (the default similarity metric). ITK⁴ supports metrics such as mutual information, \approx_{mi} , or cross-correlation, \approx_{cc} . We will use this nomenclature to write schematics for registration applications in the following sections.³

2.3. OPTIMIZATION FRAMEWORK

The general ITK⁴ optimization criterion is summarized as:

$$\text{Find mapping } \phi(x, p) \in \mathcal{T} \text{ such that } M(I, J, \phi(x, p)) \text{ is minimized.} \quad (1)$$

While, for functional mappings, this formulation is not strictly correct, the practical implementation of even high-dimensional continuous transformations involves parameterization. The space \mathcal{T} restricts the possible transformations over which to optimize the mapping ϕ . The arguments to ϕ are its parameters, p , and the spatial position, x . Note that, in ITK⁴, the image I may also contain a mapping, although it is not directly optimized in most cases. As will be seen later in the document, this mapping may also be used within large deformation metrics.

The similarity metric, M , is perhaps the most critical component in image registration. Denote a parameter set as $p = (p_1, p_2 \dots p_n)$. The metric (or comparison function between images) is then defined by $M(I, J, \phi(x, p))$. For instance, $M = \|I(x) - J(\phi(x, p))\|^2$ i.e., the sum of squared differences (SSD) metric. Its gradient with respect to parameter p_i is (using the chain rule),

$$M_{p_i} = \frac{\partial M}{\partial p_i} = \frac{\partial M}{\partial J} \frac{\partial J(\phi(x, p))}{\partial \phi} \frac{\partial \phi}{\partial p_i} \Big|_x. \quad (2)$$

This equation provides the metric gradient specified for sum of squared differences (at point x) but similar forms arise for the

³In the Demons example above, the reader may ask: why does the affine mapping, A , not appear “inside” the deformable mapping, ϕ ? Indeed, this ordering of transformations is feasible. However, this is not what we typically use in our own practice of image registration and is not what we recommend. The reason is that we usually seek a deformable mapping into a template space that is common across many subjects (i.e., the “tail” is in the same domain across subjects). This enables methods such as Jacobian-based morphometry and other groupwise comparison conveniences. For example, see Figures 1 through 4 in (Kim et al., 2008) which explains the classical approach of Jacobian-based morphometry as applied to traumatic brain injury. See also (Gaser et al., 2001; Riddle et al., 2004; Dubb et al., 2005; Lepore et al., 2006; Rohlfing et al., 2006; Studholme and Cardenas, 2007). Additional uses of Jacobian-based morphometry are shown in our examples section in the “C” example and the “asymmetry” example.

correlation and mutual information (Hermosillo et al., 2002). Both are implemented in ITK⁴ for transformations with local and global support. The $\frac{\partial J(\phi(x, p))}{\partial \phi}$ term is the gradient of J at $\phi(x)$ and $\frac{\partial \phi}{\partial p_i}$ is the Jacobian of the transformation taken with respect to its parameter. The transform $\phi(x, p)$ may be an affine map i.e., $\phi(x, p) = Ax + t$ where A is a matrix and t a translation. Alternatively, it may be a displacement field where $\phi(x, p) = x + u(x)$ and u is a vector field. In ITK⁴, both types of maps are interchangeable and may be used in a composite transform to compute registrations that map to a template via a schematic such as $I \approx \rightsquigarrow J, I \approx_{\text{mi}} \rightsquigarrow J, I \approx_{\text{cc}} \rightsquigarrow \rightsquigarrow J$ or, mixing similarity metrics, $I \approx_{\text{cc}} \rightsquigarrow \rightsquigarrow_{\text{mi}} \rightarrow J_i$.

The most commonly used optimization algorithm for image registration is gradient descent, or some variant. In the above framework, the gradient descent takes on the form of

$$\phi(p_{\text{new}}, x) = \phi\left(p_{\text{old}} + \lambda \left[\frac{\partial M}{\partial p_1}, \dots, \frac{\partial M}{\partial p_n} \right], x\right),$$

where λ is the overall learning rate and the brackets hold the vector of parameter updates.

In addition to basic gradient descent, we implement non-linear gradient descent optimization strategies which combine the conjugate gradient or gradient descent method with line search. In ITK⁴, we implement the classic *golden section* approach to identifying the optimal gradient step-size at each iteration. The generic conjugate gradient approach is performed via:

$$\gamma = \frac{\|\nabla M_t - \|\nabla M_{t-1}\|^2\|^2}{\|\nabla M_{t-1}\|^2}, \quad (3)$$

$$CG_t = \nabla M_t + \gamma CG_{t-1},$$

where CG is the conjugate gradient. The golden section line search determines the specific weight, ϵ_{opt} , of the update to the current parameters such that

$$p_{\text{new}} = p_{\text{old}} + CG_t \epsilon_{\text{opt}}.$$

Note that a naive application of gradient descent will not produce a smooth change of parameters for transformations with mixed parameter types. For instance, a change, Δ , to parameter p_i will produce a different magnitude of impact on ϕ if p_i is a translation rather than a rotation. Thus, we develop an estimation framework that sets “parameter scales” (in ITK parlance) which, essentially, customize the learning rate for each parameter. The update to ϕ via its gradient may also include other steps (such as Gaussian smoothing) that project the updated transform back to space \mathcal{T} . Multi-threading is achieved in the gradient computation, transformation update step and (if used) the regularization by dividing the parameter set into computational units that correspond to contiguous sub-regions of the image domain.

In terms of code, the Jacobian, $\frac{d\phi}{dp} \Big|_x$, is calculated at a physical point using the function `ComputeJacobianWithRespectToParameters(mappedFixedPoint, Jacobian)`. Note that it is evaluated at point x not at point $\phi(x, p)$. We then use the function `ComputeMovingImageGradientAtPoint(mappedMovingPoint,`

mappedMovingImageGradient) to compute the moving image gradient when there is no pre-warping. *ComputeMovingImageGradientAtPoint* uses central differences (or a gradient filter) in the moving image space to compute the image gradient, $\frac{dJ(\phi(x,p))}{d\phi}$.

If one is doing pre-warping, then we have an index access to the warped moving image. We compute the warped image J as $J_w(x) = J(\phi(x,p))$. Then,

$$\frac{dJ_w}{dx} = \frac{dJ(\phi(x,p))}{d\phi} \frac{d(\phi(x,p))}{dx} \quad (4)$$

$$\frac{dJ(\phi(x,p))}{d\phi} = \frac{dJ_w}{dx} \frac{d(\phi(x,p))}{dx}^{-1}$$

In code, we use *ComputeMovingImageGradientAtIndex(index, mappedMovingImageGradient)* to get $\frac{dJ_w}{dx}$ and transform this image gradient via the inverse Jacobian by calling *mappedMovingImageGradient = TransformCovariantVector(mappedMovingImageGradient, mappedMovingPoint)*.

2.4. Diffeomorphic Mapping with Arbitrary Metrics

The framework proposed above, in general form, encompasses both classic affine mapping as well as more recent large deformation strategies. Beg proposed the Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm (Miller et al., 2005) which minimizes the sum of squared differences criterion between two images. LDDMM parameterizes a diffeomorphism through a time varying velocity field that is integrated through an ODE. In ITK⁴, we implement an alternative to LDDMM that also uses a time varying field and an ODE but minimizes a more general objective function:

$$E(\mathbf{v}) = M(I, J, \phi_{1,0}) + w \int_0^1 \|\mathcal{L}v_t\|^2 dt. \quad (5)$$

This is an instance of Equation (1) where w is a scalar weight and $\phi_{1,0}$ is a standard integration of the time-varying velocity field, v_t ,

which is regularized by the linear operator \mathcal{L} . ITK⁴ uses Gaussian smoothing which is the Green's kernel for generalized Tikhonov regularization (Nielsen et al., 1997). This objective is readily optimized using an approach that is similar to that proposed by Beg. Generalization of the LDDMM gradient for other metrics basically follows (Hermosillo et al., 2002) with a few adjustments to accommodate diffeomorphic mapping. **Figure 3** shows an ITK result on a standard example for large deformation registration. We will evaluate this diffeomorphic mapping, along with parameter estimation, in a later section.

2.5. Parameter Scale Estimation

We choose to estimate parameter scales by analyzing the result of a small parameter update on the change in the magnitude of physical space deformation induced by the transformation. The impact from a unit change of parameter p_i may be defined in multiple ways, such as the maximum shift of voxels or the average norm of transform Jacobians (Jenkinson and Smith, 2001). Denote the unscaled gradient descent update to p as Δp . The goal is to rescale Δp to $q = s \cdot \Delta p$, where s is a diagonal matrix $\text{diag}(s_1, s_2 \dots s_n)$, such that a unit change of q_i will have the same impact on deformation for each parameter $i = 1 \dots n$.

As an example, we want $\|\phi(x, p_{\text{new}}) - \phi(x, p_{\text{old}})\| = \text{constant}$ regardless of which of the i parameters is updated by the unit change. The unit is an epsilon value e.g., 1.e-3. Rewrite $\left[\frac{\partial M}{\partial p_1}, \dots, \frac{\partial M}{\partial p_n}\right]$ as $\frac{\partial M}{\partial J} \frac{\partial J(\phi(x,p))}{\partial \phi} \left[\frac{\partial \phi}{\partial p_1}, \dots, \frac{\partial \phi}{\partial p_n}\right]$. To determine the relative scale effects of each parameter, p_i , we can factor out the constant terms on the outside of the bracket. Then the modified gradient descent step becomes $\text{diag}(s) \frac{\partial \phi}{\partial p}$. We identify the values of $\text{diag}(s)$ by explicitly computing the values of $\|\phi(x, p_{\text{new}}) - \phi(x, p_{\text{old}})\|$ with respect to an ϵ change. A critical variable, practically, is which x to choose for evaluation of $\|\phi(x, p_{\text{new}}) - \phi(x, p_{\text{old}})\|$. The corners of the image domain work well for affine transformations. In contrast, local regions of small radius (approximately 5) work well for transformations with local support. Additional work is needed to verify optimal parameters for this new ITK⁴ feature. However, a preliminary evaluation is

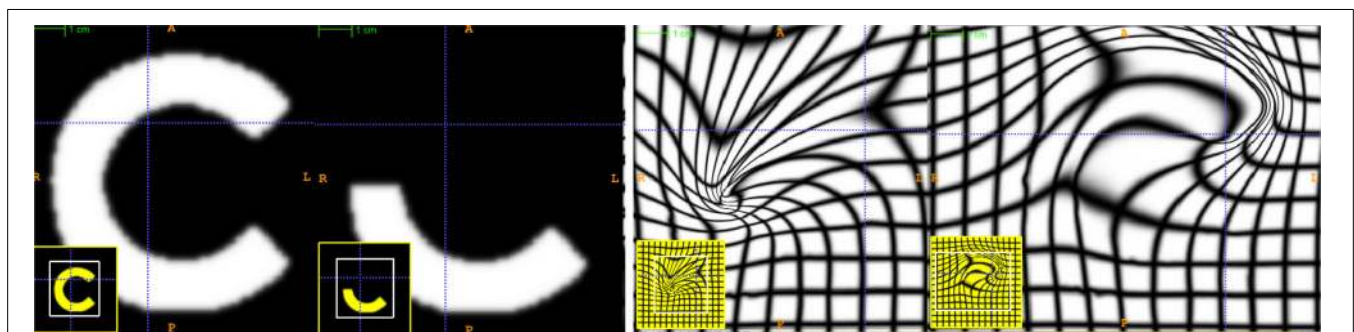


FIGURE 3 | An ITK diffeomorphic mapping of the type $I \leftrightarrow J$. The “C” and 1/2 “C” example illustrate the large deformations that may be achieved with time varying velocity fields. In this case, the moving (deforming) image is the 1/2 “C.” The right panels illustrate the deformed grid for the transformation of the “C” to 1/2 “C” (middle right) and its inverse mapping (far right) which takes the 1/2 “C” to the reference space. The unit time interval is discretized into 15 segments in order to compute this mapping.

15*5 integration steps were used in the Runge-Kutta ODE integration over the velocity field. A two core MacBook Air computed this registration in 110 s. The images each were of size 150 × 150. See C for a reproducible example of this registration and the data. In addition, we provide an example of how the Jacobian determinant is computed from the deformation field resulting from this registration via an ANTs program `CreateJacobianDeterminantImage`.

performed in the results section. The new parameter scale estimation effectively reduces the number of parameters that the user must tune from $k + 1$ (λ plus the scales for each parameter type where there are k types) to only 1, the learning rate.

The learning rate, itself, may not be intuitive for a user to set. The difficulty—across problem sets—is that a good learning rate for one problem may result in a different amount of change per iteration in another problem. Furthermore, the discrete image gradient may become invalid beyond one voxel. Thus, it is good practice to limit a deformation step to one voxel spacing (Jenkinson and Smith, 2001). We therefore provide the users the ability to specify the learning rate in terms of the *maximum physical space change per iteration*. As with the parameter scale estimation, the domain over which this maximum change is estimated impacts the outcome and similar practices are recommended for both cases. This feature is especially useful for allowing one to tune gradient descent parameters without being concerned about which similarity metric is being used. That is, it effectively rescales the term $\lambda \partial M / \partial p$ to have a consistent effect, for a given λ , regardless of the metric choice. In combination with our non-linear conjugate gradient approach (our current optimization of choice for linear registration), this strategy drastically reduces the parameter setting burden for users.

3. RESULTS

3.1. EXAMPLE APPLICATIONS OF THE ITK⁴ FRAMEWORK

As part of our work in ITK refactoring, we built, in parallel to library programming, an application interface that allows high-level access to the deep layers of ITK registration. These currently exist in the *Advanced Normalization Tools* (ANTs) software (link). While ANTs still serves as intermediate (vs. direct) access-point to these tools, it provides a high-degree of customization possibilities simply through a command line interface and scripting. Therefore, a user is not required to write new low-level (interestingly, C++ is now considered “low-level”) software.

Despite its relative youth, the ANTs wrapping of ITK functionality has been employed with notable success in recent public, unbiased, international evaluation studies. ANTs was instrumental to a first-place finish in SATA 2013 in two of three categories (based on the median performance) where the ANTs approach was considerably simpler than that employed by close finishers. While the evaluation of deep-gray matter registration showed relatively subtle differences, the ANTs solution to the multivariate canine leg data outclassed all other entrants. Notably, the ANTs solution used a multiple metric approach that simultaneously compared two modalities during registration as in Avants et al. (2008). In the cardiac data, the ANTs solution was the only one that was fully automated resulting in a $\approx 15\%$ performance loss which can easily be overcome by a modicum of user intervention. Furthermore, ANTs/ITK-based methods finished a clear first-place in the BRATS 2013 challenge. Our entry used intensity asymmetry as a key feature to segment brain tumors based on multiple modality MRI. Thus, these methods are within the leading ranks of image registration methodologies as evaluated in recent work as well as in the more traditional brain (Klein et al., 2010) and lung CT (Murphy et al., 2011) studies.

The ANTs contribution is valuable, in part, because of the tremendous range of registration problems that exist in neuroinformatics and biomedical imaging in general. While it is not possible to solve all registration problems with a general framework, one cannot afford to invent new solutions for every instance one encounters. Our general optimization-driven strategies have proven to be invaluable to setting performance standards in a variety of application domains. In this section, we highlight some of the lesser known capabilities of ANTs and ITK⁴ with reproducible examples that include data and specific commands to ANTs and/or ITK. A list of these examples follows:

1. The Basic Brain Mapping example shows how one can map two “whole head” T1-weighted MRIs where one is a template that contains a researcher’s prior knowledge defining the “interesting” parts of the image. Within ITK⁴, this domain knowledge is used to focus the `Metricsv4` on only those parts of the image while masking the remainder. Furthermore, a second part of this example shows how the ITK composite transform may be used to initialize new registration solutions as well as how masking functionality may be employed to ignore information that is irrelevant or obstructive to the registration optimization. We have previously employed these strategies in brain mapping with lesions (Avants et al., 2008; Kim et al., 2008, 2010, 2013; Tustison et al., 2011).
2. We use updated ITK methods in template construction with a reproducible example based on face and brain data: ANTs template construction. This work has been employed in different species, age-ranges, and imaging modalities. The resulting template is an image that captures the expected shape and appearance as defined by the population sample, transformation model and intensity comparison metric.
3. A large deformation example implementing the classic “letter C” example provided, originally, by Gary Christensen. While extremely flexible, these algorithms have not found a unique identity in terms of translational applications yet remain of theoretical interest. This example shows a user how to define the parameters of a registration based on optimizing a time varying velocity field.
4. We present a separate example of how to compute landmark-based registration error. ITK uses LPS coordinates to represent physical space. If you need to convert landmarks to physical space, see the discussion here: LPS physical space. We have an example illustrating how to change point coordinates and apply ITK transforms to landmarks here. This exercise can be useful for landmark-based registration or in evaluating registration accuracy.
5. We show how to perform motion correction to time series data here although we do not claim this approach is optimal. The method registers each frame from a 4D time series to a fixed reference image and stores the resampled set in a new 4D image. All transformation parameters are stored in a corresponding csv file.
6. An advanced example with heavy use of statistics via *R* and *ANTsR* is in a study of public test-retest fmri data. This study is not published and may be subject to change.

7. The classic car example shown in ANTs talks is here. This illustrates the benefit of mutual information in deformably mapping wild-type images and highlights the fact that ITK⁴ applications exist outside of medical imaging.
8. A basic multistart example. A more advanced example for brain mapping with a template mask is available in *antsCorticalThickness.sh*. These optimization methods overcome local minima by running registration searches from a variety of starting points and greedily storing the best solution.
9. This asymmetry analysis example uses the “virtual domain” feature to reduce bias caused by mapping an image asymmetrically to a reference. Note that we measure the point-wise asymmetry, in this example, via the Jacobian determinant image as in Kim et al. (2008). If one repeats this analysis across a population—and maps the Jacobian measurements of asymmetry to a common space—then one may perform a statistical analysis of population-level asymmetry. Longitudinal analysis and asymmetry analysis potentially suffer from the same confound Yushkevich et al. (2010). A related longitudinal mapping script is here.
10. We also show how manual labelings can be used to restrict registration in a challenging registration scenario (this example will be improved in the future): registration guided by (crude) labels.
11. A simple orange to apple RGB image registration example for color images is listed at: `itkMeanSquaresImageToImageMetricv4VectorRegistrationTest`. If one compiles the ITK tests, then this example can be run to produce **Figure 4**.

These examples cover many applications for which no “ground truth” evaluation data exists. The next section seeks to add some quantitative reference to these examples. First, we show flexibility and consistency of our framework in a simple example comparing registration with a variety of metrics and a consistent parameter set. Second, we quantify the benefit of ITK⁴ registration in comparison to a method implemented based upon ITK³ registration technology.

3.2. EVALUATION

We first investigate the ability of our automated parameter estimation to facilitate parameter tuning across metrics. Second, we compare ITK⁴ and ITK³ registration implementations with respect to a standard automated brain labeling task.

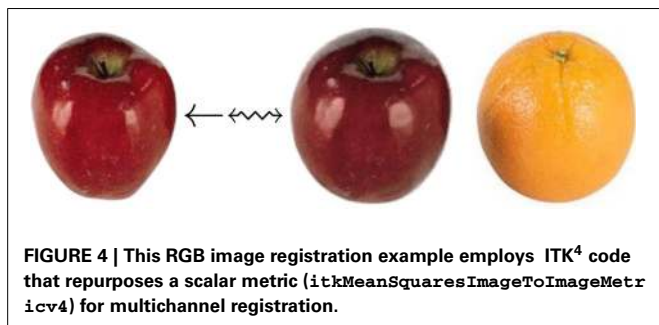


FIGURE 4 | This RGB image registration example employs ITK⁴ code that repurposes a scalar metric (`itkMeanSquaresImageToImageMetricv4`) for multichannel registration.

3.2.1. Parameter estimation across metrics

ITK⁴ provides similarity metrics that may be applied for both deformable and affine registration. In a previous section, we provided a parameter estimation strategy that is applicable to both deformable and affine transformations with arbitrary metrics. Denote images I, J, K , where the latter two are “moving” images, and K is an intensity-inverted version of J . We then evaluate the following schema,

$$I \approx_{\text{mi}} \leftrightarrow J, \quad I \approx_{\text{cc}} \leftrightarrow J, \quad I \approx_{\text{mi}} \leftrightarrow K$$

where, for each schematic, we use the corresponding metric for both affine and diffeomorphic mapping. Furthermore, we keep the same parameters for each registration by exploiting parameter scale estimators. **Figure 5** shows the candidate images for this test.

As shown in **Figure 5**, very similar results are achieved for each schematic without additional parameter tuning. To determine this quantitatively, we perform registration for each schematic and then compare the Dice overlap of a ground-truth three-tissue segmentation. For each result, we have the Dice overlap of dark tissue (cerebrospinal fluid, CSF), medium intensity tissue (gray matter) and bright tissue (white matter). For the mean squares metric, we have: 0.588, 0.816, and 0.90; for CC, we have: 0.624, 0.786, 0.882; for MI, we have: 0.645, 0.779, 0.858. Mutual information does best for the CSF while mean squares does best for other tissues. CC performs in the mid-range for all classes of tissue. Thus, a single set of tuned parameters provides a reasonable result for an affine plus diffeomorphic mapping across three different metrics. While improvement might be gained by further tuning for each metric, this result shows that our parameter estimation method achieves the goal of reducing user burden.

3.2.2. Automated Brain Labeling Task

All *R* and bash analysis scripts for this section are here: <https://github.com/stnava/ITKv4Documentation/tree/frontiers/scripts>. The ITK⁴ core functionality formed the heart of the reference results provided for the SATA2013 challenge at MICCAI 2013. In this sense, these methods have been heavily evaluated on both basic brain mapping challenges (SATA2013’s diencephalon challenge in which ITK⁴-based methods finished first), multivariate registration challenges (the canine MRI / dog leg challenge of SATA2013 in which ITK⁴-based methods were overwhelmingly the top finisher) and in the cardiac challenge (in which ITK⁴-based methods were the only fully automated approach). However, for completeness, we provide an additional evaluation here which focuses on comparison to a ITK³ method, BRAINSFit, in a different dataset than previously used to evaluate ANTs or ITK⁴.

As ground truth, we use T1 MRI data from 33 2-year old subjects as described in Gousias et al. (2008) and available at <http://www.brain-development.org>. Each subject’s brain is manually parcellated into 83 distinct regions that include ventricles, cortical areas, white matter and deep gray matter regions such as the amygdala, hippocampus and thalamus. One benefit of this data is that some of these anatomical regions are relatively easy to align (the caudate) whereas others are relatively difficult to align due to their small size (amygdala) or inconsistent shape across subjects (the inferior frontal gyrus). Thus,

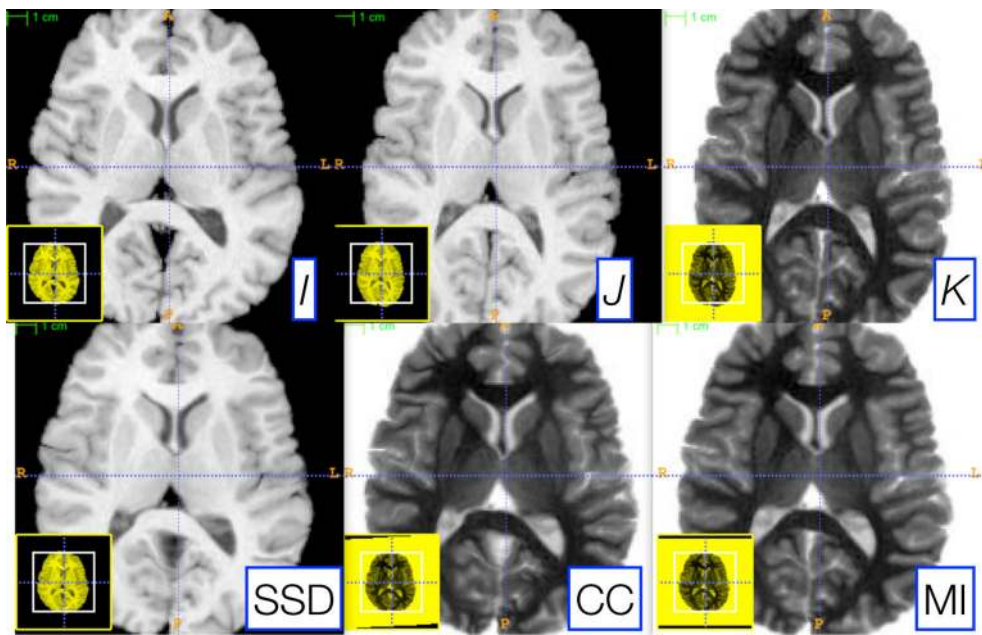


FIGURE 5 | Three reference images, *I* (left), *J* (middle top), and *K* (right top), are used to illustrate the robustness of our parameter scale estimation for setting consistent parameters across both metrics and transform types. *K* is the negation of *J* and is used to test the correlation and mutual information registrations. We optimized, by hand, the step-length parameters for one metric (the sum of squared differences) for both the

affine and deformable case. Thus, two parameters had to be optimized. We then applied these same parameters to register *I* and *K* via both correlation and mutual information. The resulting registrations (bottom row) were all of similar quality. Further, the same metric is used for both affine and diffeomorphic mapping by exploiting the general optimization process given in Equation (1).

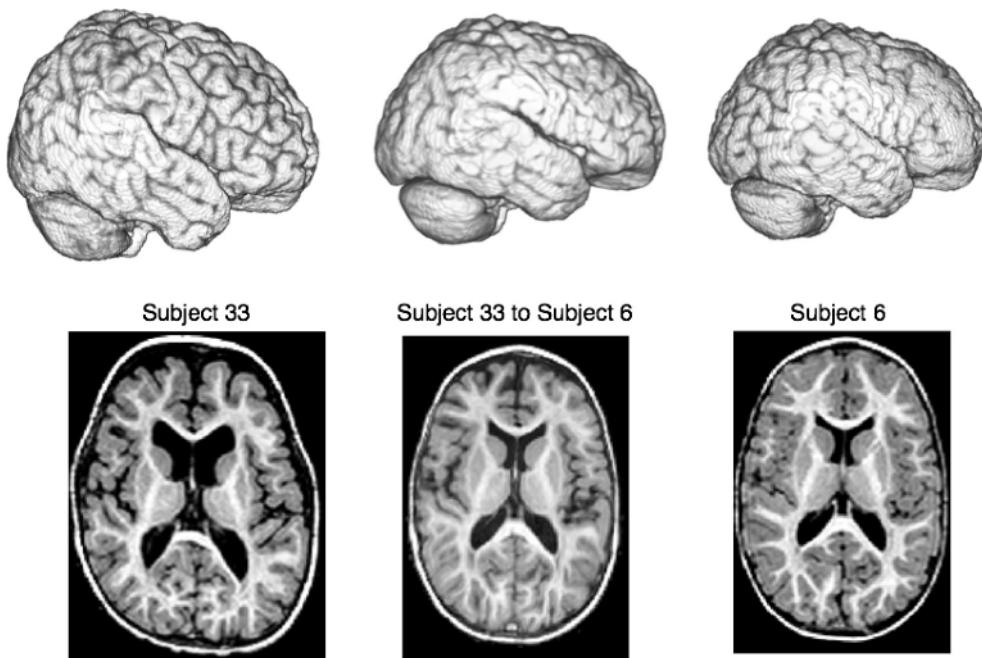


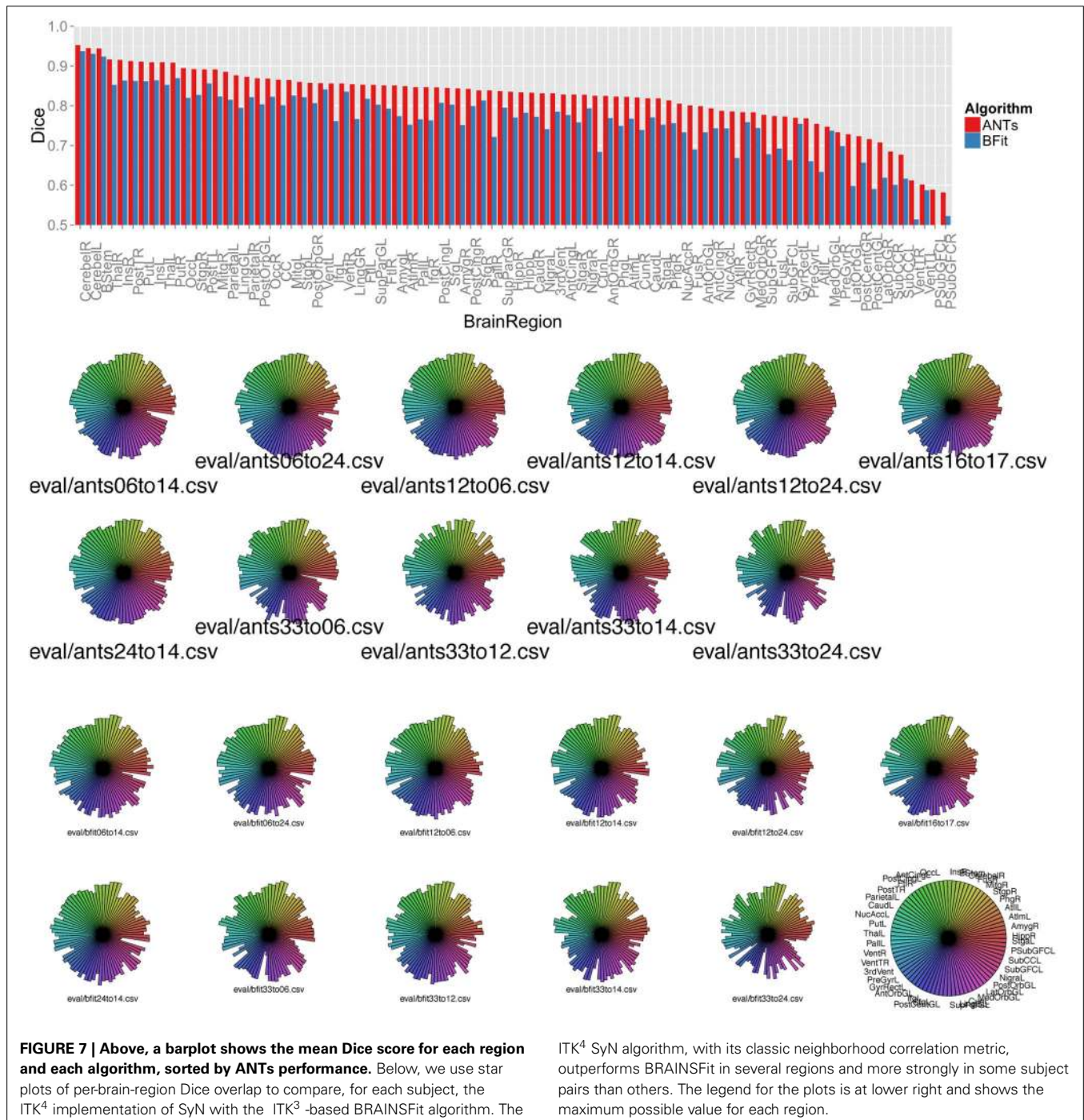
FIGURE 6 | We compare a ITK⁴ composite schema as $I \approx_{cc} \rightsquigarrow \approx_{mi} \rightarrow J_i$ for mapping a set of $\{J_i\}$ images to a template *I* to a ITK³ schema: $I \approx_{mi} \rightsquigarrow b \approx_{mi} \rightarrow J_j$. We use this schematic in a registration-based segmentation of multiple brain structures in a pediatric population as a benchmark for algorithm performance, similar to Klein et al. (2010). An

example ANTs-based large-deformation result from the dataset is shown for illustration where we render the extracted brains as well as show select axial slices. All registrations were run on the original MRI data with no preprocessing except what is done by ANTs or BRAINSFit internally. Overlap improvement from v3 to v4, quantified via paired *t*-test, is highly significant.

we anticipate that performance gains due to new technology in ITK⁴ will be most prominent in the more variable and challenging regions. **Figure 6** summarizes the study nomenclature and shows a single image pair selected from this data along with the registration result given by ITK⁴. **Figure 7** summarizes these evaluation results. The scripts for running this study are available at <https://github.com/stnava/ITKv4Documentation/tree/frontiers/scripts>. The git hashtag for the ANTs version used in this evaluation <https://github.com/stnava/ANTs> is ce8b5a7414ae9e389071d756c5f36ee6cecbcf8d. The associated ITK tag is

contained within the ANTs repository. The git hashtag for the BRAINSFit version <https://github.com/BRAINSia/BRAINSTools> used in this evaluation is ad7e114ab1c92bd800819b80e0548259398931c8. Both programs were run on a MacBook Pro running OS X 10.9 (13A3028) with a 2.6 GHz Intel Core i7 and 16 GB of RAM.

To help isolate which subject pairs to deformably register, we first clustered the initial dataset based on an all pairs affine registration which revealed five representative subject clusters. The subject pairings used during evaluation were chosen such



that each subject pair contained the most representative subject for one cluster paired with the most representative subject from another cluster; thus, the study design allows us to focus, in a principled manner, on a set of representative shape comparisons where the comparisons are made across different image types.

The new methods in ITK⁴ show enhanced performance within all registration pairs. The mean overall performance gain was approximately 6.3% with a standard deviation of 5% and *T*-statistic/*p*-value, over all structures, of 12.6 / $p < 1.e - 16$. We also identified which regions were most improved in ITK⁴ vs. ITK³. These regions include the left and right insula, the brainstem, the superior temporal gyrus, parahippocampal gyrus, putamen, and the substantia nigra. **Table 1** lists all structures and the mean Dice score for each algorithm, along with the *p*-value. **Figure 7** summarizes all of these findings by using star plots to visualize the Dice results for every region in every subject.

We also recorded the amount of processing time spent on each subject, for each algorithm. Noting that the ITK⁴ algorithm also provides a dense and high-resolution forward and inverse transform and does explicit transformation regularization to guarantee a diffeomorphism, the algorithm takes, on average, 5 times as long as the ITK³ BRAINSFit algorithm (≈ 10 min), assuming default settings. Much of this time, in ANTs, is taken up by full resolution image registration. If this fine level is avoided, then the disparity in timing reduces to less than a factor of two, without much loss in accuracy. Note also that ANTs and BRAINSFit each use a different multithreading strategies, similarity metric implementations, rigid/affine registration mechanisms and optimizers making this overall comparison less than ideal.

4. DISCUSSION

ITK is a community built and maintained toolkit and is a public resource for reproducible methods. The updated ITK⁴ registration framework provides a novel set of user-friendly parameter setting tools and benchmark implementations of both standard and advanced algorithms. Robustness with respect to parameter settings has long been a goal of image registration and ITK⁴ takes valuable steps toward the direction of automated parameter selection. The primary decision left up to the user, currently, is the feature scale at which registration should be performed. E.g., whether the registration should focus on coarse features, fine features, etc and the different resolutions at which this should be done. While we have provided a reproducible reference comparison of registration-based brain labeling in this paper, we intend to have a more extensive series of benchmark performance studies completed on datasets beyond the brain. However, the number of possible applications exceeds what can possibly be evaluated by the core ITK developer community. Community involvement is needed in order to increase the number of possible registration applications and metric/transform/optimizer/data combinations that have been evaluated. At the same time, documentation, usability and examples must be provided by the development team in order to improve user involvement.

4.1. FUTURE WORK

Future work will enhance the depth and breadth of documentation as well as seek to further optimize the current

Table 1 | Dice overlap for ANTs and BRAINSFit where only regions with *q*-value < 0.01 are shown.

	meanants	meanbfit	FDR-adjusted <i>p</i> -value
R Hippocampus	0.835	0.770	0.0083
L Hippocampus	0.834	0.782	0.0063
R Amygdala	0.843	0.752	0.0027
L Amygdala	0.851	0.773	0.0037
L AnteriorTemporalLobeMedialPart	0.849	0.752	0.0005
R AnteriorTemporalLobeLateralPart	0.754	0.633	0.0002
L AnteriorTemporalLobeLateralPart	0.786	0.668	0.0083
R Gyri parahippocampalis et ambiens	0.813	0.756	0.0037
L Gyri parahippocampalis et ambiens	0.823	0.749	0.0007
R Superior temporal gyrus posterior	0.892	0.827	0.0001
R Medial and inferior temporal gyri	0.891	0.823	0.0007
R Lateral occipitotemporal gyrus (gyrus fusiformis)	0.801	0.690	0.0023
R Cerebellum	0.953	0.937	0.0066
Brainstem	0.944	0.923	0.0001
R Insula	0.910	0.864	0.0001
L Insula	0.915	0.863	0.0001
L Occipital lobe	0.895	0.819	0.0049
L Posterior temporal lobe	0.912	0.862	0.0001
L Parietal lobe	0.885	0.815	0.0052
R Putamen	0.909	0.869	0.0002
L Putamen	0.911	0.862	0.0003
R Pallidum	0.838	0.721	0.0037
L Pallidum	0.846	0.766	0.0049
L Lingual gyrus	0.876	0.795	0.0037
L Cuneus	0.825	0.684	0.0083
L Lateral orbital gyrus	0.728	0.598	0.0037
L Substantia nigra	0.831	0.741	0.0023
L Superior temporal gyrus anterior part	0.818	0.752	0.0043

implementations for speed and memory. In time, it may be possible to extend the design philosophy used here to GPU implementations. However, our ability to interface low and high-dimensional transformations depends heavily on generic programming. This style is less well-developed (and less well understood) in GPU applications which depend, to some extent, on specialization. The current framework is amenable to groupwise registration strategies when used in combination with a computing cluster. However, single core groupwise strategies are not currently implemented although one may consider basing an implementation on existing multi-metric/multivariate registration tools within the current code base. While ITK⁴ does contain a statistics infrastructure, we currently prefer using *R* and *ANTsR* for analyzing our data.

However, the lack of visualization methods in ITK means that one must still move to another package to look at one's results. Therefore, direct interfaces to *R* remain useful. SimpleITK also has a promising *R* interface that is similar to *ANTsR*.

A primary challenge to the future of ITK⁴ includes, beyond documentation, reduced C++ fluency. As ITK⁴ leverages several advanced features of C++, even experienced developers may find it difficult to contribute meaningfully to the ITK software base. Therefore, the ITK⁴ community must also seek to educate potential future contributors not only on ITK but also, at times, on the fundamentals or advanced extensions of C++. A second major hurdle is that ITK⁴ includes a host of generic registration ingredients. However, many of the most compelling new application domains require specialization. Specialization may be needed for a specific imaging modality, via hardware interface or in the use of domain-specific prior knowledge. Therefore, we envision the next phase of ITK⁴ development may focus on using the toolkit to support its specialization in solving high-impact and translational applications. Hopefully, this transition will occur in the near future.

REFERENCES

- Ackerman, M. J., and Yoo, T. S. (2003). The visible human data sets (VHD) and insight toolkit (ITk): experiments in open source software. *AMIA Annu. Symp. Proc.* 2003:773.
- Avants, B. B., Tustison, N. J., Song, G., Cook, P. A., Klein, A., and Gee, J. C. (2011). A reproducible evaluation of ANTs similarity metric performance in brain image registration. *Neuroimage* 54, 2033–2044. doi: 10.1016/j.neuroimage.2010.09.025
- Avants, B., Duda, J. T., Kim, J., Zhang, H., Pluta, J., Gee, J. C., et al. (2008). Multivariate analysis of structural and diffusion imaging in traumatic brain injury. *Acad. Radiol.* 15, 1360–1375. doi: 10.1016/j.acra.2008.07.007
- Baloch, S., and Davatzikos, C. (2009). Morphological appearance manifolds in computational anatomy: groupwise registration and morphological analysis. *Neuroimage* 45(1 Suppl), S73–S85. doi: 10.1016/j.neuroimage.2008.10.048
- Bearden, C. E., van Erp, T. G. M., Dutton, R. A., Tran, H., Zimmermann, L., Sun, D., et al. (2007). Mapping cortical thickness in children with 22q11.2 deletions. *Cereb. Cortex* 17, 1889–1898. doi: 10.1093/cercor/bhl097
- Buckheit, J. B., and Donoho, D. L. (1995). "WaveLab and reproducible research," in *Wavelets and statistics*, eds A. Antoniadis and G. Oppenheim (New York, NY: Springer), 55–81.
- Chen, M., Lu, W., Chen, Q., Ruchala, K. J., and Olivera, G. H. (2008). A simple fixed-point approach to invert a deformation field. *Med. Phys.* 35, 81–88. doi: 10.1118/1.2816107
- Cheung, M. R., and Krishnan, K. (2009). Interactive deformation registration of endorectal prostate mri using itk thin plate splines. *Acad. Radiol.* 16, 351–357. doi: 10.1016/j.acra.2008.09.011
- Christensen, G. E., Rabbitt, R. D., and Miller, M. I. (1996). Deformable templates using large deformation kinematics. *IEEE Trans. Image Process.* 5, 1435–1447. doi: 10.1109/83.536892
- Dubb, A., Xie, Z., Gur, R., Gur, R., and Gee, J. (2005). Characterization of brain plasticity in schizophrenia using template deformation. *Acad. Radiol.* 12, 3–9. doi: 10.1016/j.acra.2004.06.009
- Fedorov, A., Li, X., Pohl, K. M., Bouix, S., Styner, M., Addicott, M., et al. (2011). Atlas-guided segmentation of vervet monkey brain MRI. *Open Neuroimag. J.* 5, 186–197. doi: 10.2174/1874440001105010186
- Floca, R., and Dickhaus, H. (2007). A flexible registration and evaluation engine (f.r.e.e.). *Comput. Methods Programs Biomed.* 87, 81–92. doi: 10.1016/j.cmpb.2007.04.009
- Gaser, C., Nenadic, I., Buchsbaum, B. R., Hazlett, E. A., and Buchsbaum, M. S. (2001). Deformation-based morphometry and its relation to conventional volumetry of brain lateral ventricles in mri. *Neuroimage* 13(6 pt 1), 1140–1145. doi: 10.1006/nimg.2001.0771
- Gousias, I. S., Rueckert, D., Heckemann, R. A., Dyet, L. E., Boardman, J. P., Edwards, A. D., et al. (2008). Automatic segmentation of brain mris of 2-year-olds into 83 regions of interest. *Neuroimage* 40, 672–684. doi: 10.1016/j.neuroimage.2007.11.034
- Hermosillo, G., Chefed'Hotel, C., and Faugeras, O. (2002). A variational approach to multi-modal image matching. *Intl. J. Comp. Vis.* 50, 329–343. doi: 10.1023/A:1020830525823
- Jack, C. R. Jr., Knopman, D. S., Jagust, W. J., Shaw, L. M., Aisen, P. S., Weiner, M. W., et al. (2010). Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade. *Lancet Neurol.* 9, 119–128. doi: 10.1016/S1474-4422(09)70299-6
- Jenkinson, M., and Smith, S. (2001). A global optimisation method for robust affine registration of brain images. *Med. Image Anal.* 5, 143–156. doi: 10.1016/S1361-8415(01)00036-6
- Kikinis, R., and Pieper, S. (2011). 3d slicer as a tool for interactive brain tumor segmentation. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2011, 6982–6984. doi: 10.1109/IEMBS.2011.6091765
- Kim, J., Avants, B., Patel, S., Whyte, J., Coslett, B. H., Pluta, J., et al. (2008). Structural consequences of diffuse traumatic brain injury: a large deformation tensor-based morphometry study. *Neuroimage* 39, 1014–1026. doi: 10.1016/j.neuroimage.2007.10.005
- Kim, J., Avants, B., Whyte, J., and Gee, J. C. (2013). Methodological considerations in longitudinal morphometry of traumatic brain injury. *Front. Hum. Neurosci.* 7:52. doi: 10.3389/fnhum.2013.00052
- Kim, J., Whyte, J., Patel, S., Avants, B., Europa, E., Wang, J., et al. (2010). Resting cerebral blood flow alterations in chronic traumatic brain injury: an arterial spin labeling perfusion fmri study. *J. Neurotrauma* 27, 1399–1411. doi: 10.1089/neu.2009.1215
- Klein, S., Staring, M., Murphy, K., Viergever, M. A., and Pluim, J. P. W. (2010). elastix: a toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* 29, 196–205. doi: 10.1109/TMI.2009.2035616
- Lepore, N., Brun, C. A., Chiang, M.-C., Chou, Y.-Y., Dutton, R. A., Hayashi, K. M., et al. (2006). Multivariate statistics of the jacobian matrices in tensor based morphometry and their application to hiv/aids. *Med. Image Comput. Comput. Assist. Interv.* 9(pt 1), 191–198.
- Metz, C. T., Klein, S., Schaap, M., van Walsum, T., and Niessen, W. J. (2011). Nonrigid registration of dynamic medical imaging data using nd + t b-splines and a groupwise optimization approach. *Med. Image Anal.* 15, 238–249. doi: 10.1016/j.media.2010.10.003
- Miller, M. I., Beg, M. F., Ceritoglu, C., and Stark, C. (2005). Increasing the power of functional maps of the medial temporal lobe by using large deformation diffeomorphic metric mapping. *Proc. Natl. Acad. Sci. U.S.A.* 102, 9685–9690. doi: 10.1073/pnas.0503892102
- Murphy, K., van Ginneken, B., Reinhardt, J. M., Kabus, S., Ding, K., Deng, X., et al. (2011). Evaluation of registration methods on thoracic CT: the EMPIRE10 challenge. *IEEE Trans. Med. Imaging* 30, 1901–1920. doi: 10.1109/TMI.2011.2158349
- Nielsen, M., Florack, L., and Deriche, R. (1997). Regularization, scale-space, and edge detection filters. *J. Math. Imaging Vis.* 7, 291–307. doi: 10.1023/A:1008282127190
- Peyrat, J.-M., Delingette, H., Sermesant, M., Xu, C., and Ayache, N. (2010). Registration of 4d cardiac ct sequences under trajectory constraints with multi-channel diffeomorphic demons. *IEEE Trans. Med. Imaging* 29, 1351–1368. doi: 10.1109/TMI.2009.2038908
- Riddle, W. R., Li, R., Fitzpatrick, J. M., DonLevy, S. C., Dawant, B. M., and Price, R. R. (2004). Characterizing changes in mr images with color-coded jacobians. *Magn. Reson. Imaging* 22, 769–777. doi: 10.1016/j.mri.2004.01.078
- Rohlfing, T., Sullivan, E. V., and Pfefferbaum, A. (2006). Deformation-based brain morphometry to track the course of alcoholism: differences between intra-subject and inter-subject analysis. *Psychiatry Res.* 146, 157–170. doi: 10.1016/j.psychres.2005.12.002
- Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L., Leach, M. O., and Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: application to breast mr images. *IEEE Trans. Med. Imaging* 18, 712–721. doi: 10.1109/42.796284

- Shelton, D., Stetten, G., Aylward, S., Ibáñez, L., Cois, A., and Stewart, C. (2005). Teaching medical image analysis with the insight toolkit. *Med. Image Anal.* 9, 605–611. doi: 10.1016/j.media.2005.04.011
- Studholme, C., and Cardenas, V. (2007). Population based analysis of directional information in serial deformation tensor morphometry. *Med. Image Comput. Comput. Assist. Interv.* 10(pt 2), 311–318.
- Tustison, N., Avants, B., Cook, P., Kim, J., Whyte, J., Gee, J., et al. (2011). “Multivariate analysis of diffusion tensor imaging and cortical thickness maps in a traumatic brain injury (tbi) cohort using advanced normalization tools (ants) Journal of Neurotrauma 28 A111,” in *29th Annual National Neurotrauma Symposium* (Hollywood Beach, FL).
- van Dalen, J. A., Vogel, W., Huisman, H. J., Oyen, W. J. G., Jager, G. J., and Karssemeijer, N. (2004). Accuracy of rigid CT-FDG-PET image registration of the liver. *Phys. Med. Biol.* 49, 5393–5405. doi: 10.1088/0031-9155/49/23/014
- Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. (2009). Diffeomorphic demons: efficient non-parametric image registration. *Neuroimage* 45(1 Suppl), S61–S72. doi: 10.1016/j.neuroimage.2008.10.040
- Yoo, T. S., Ackerman, M. J., Lorensen, W. E., Schroeder, W., Chalana, V., Aylward, S., et al. (2002). Engineering and algorithm design for an image processing api: a technical report on itk—the insight toolkit. *Stud. Health Technol. Inform.* 85, 586–592.
- Yushkevich, P. A., Avants, B. B., Das, S. R., Pluta, J., Altinay, M., Craige, C., et al. (2010). Bias in estimation of hippocampal atrophy using deformation-based morphometry arises from asymmetric global normalization: an illustration in adni 3 t mri data. *Neuroimage* 50, 434–445. doi: 10.1016/j.neuroimage.2009.12.007

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 01 November 2013; accepted: 30 March 2014; published online: 28 April 2014.

Citation: Avants BB, Tustison NJ, Stauffer M, Song G, Wu B and Gee JC (2014) The Insight ToolKit image registration framework. *Front. Neuroinform.* 8:44. doi: 10.3389/fninf.2014.00044

This article was submitted to the journal *Frontiers in Neuroinformatics*.

Copyright © 2014 Avants, Tustison, Stauffer, Song, Wu and Gee. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.