A unified image registration framework for ITK

Brian B. Avants[1], Nicholas J. Tustison[2], Gang Song[1], Baohua Wu[1], Michael Stauffer[1],
Matthew M. McCormick[3], Hans J. Johnson[4], James C. Gee[1]
and the [5]Insight Software Consortium
[1]Penn Image Computing and Science Lab
Dept. of Radiology
University of Pennsylvania, Philadelphia, PA, 19104
[2]Dept. of Radiology and Medical Imaging,
University of Virginia, Charlottesville, VA 22903
[3]Department of Psychiatry
The University of Iowa, IA 52242
[4]Kitware, Inc.
Clifton Park, NY 12065
[5]http://www.insightsoftwareconsortium.org/

October 21, 2012

1 Abstract

Publicly available scientific resources determine evaluation standards, provide a platform for teaching and may improve reproducibility. The image registration team, centered at the University of Pennsylvania's Penn Image Computing and Science Laboratory, established new standards in publicly available image registration methodology in Version 4 of the Insight ToolKit (ITK[4]). In this work, we provide an overview of the revised toolkit. In total, the ITK[4] 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. [1]

2 Introduction

As image registration methods mature—and their capabilities become more widely recognized—the number of applications increase [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. 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 [13], plays a major role in business models and clinical products (especially in radiation oncology) [6], has led to numerous new findings in studies of brain and behavior (e.g. [14]) and is a critical component in applications in pathology, microscopy, surgical planning and more [3, 4, 15, 5, 6, 8, 10, 12]. Despite the increasing relevance of image registration across application domains, there are relatively few reference algorithm implementations available to the community.

One source of benchmark methodology is the Insight ToolKit (ITK) [16, 17], which marked a significant contribution to medical image processing when it first emerged over 10 years ago. 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). Numerous papers use ITK algorithms as standard references for implementations of Demons registration and mutual information-based affine or B-Spline registration [2, 3, 15, 5, 6]. Multiple toolkits extend ITK registration methods in unique ways. Elastix provides very fast and accurate B-Spline registration [18, 12]. The diffeomorphic

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

demons is a fast/efficient approximation to a diffeomorphic mapping [19]. ANTs provides both flexibility and high average performance [20]. The BrainsFit algorithm is integrated into slicer for user-guided registration [10]. Each of these toolkits has both strengths and weaknesses [18, 12] and was enabled by an ITK core.

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 [21, 22, 1, 4, 8, 20]. To make this technology more accessible, ITK^4 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 ITK^4 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. ITK^4 transforms are also newly applicable to objects such as vectors and tensors and will take into account covariant geometry if necessary. Finally, ITK^4 reconfigures the registration framework to use multi-threading in as many locations as 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.

The remainder of the document will provide an overview of the new framework via the context of a potential general nomenclature. We also establish performance benchmarks for the current ITK^4 registration. Finally, we discuss future developments in the framework.

3 Nomenclature

The nomenclature below designates an image registration algorithm pictorially. This nomenclature is intended to be a descriptive, but also technically consistent, system for visually representing algorithms and applications of registration. Ideally, any standard algorithm can be written in the nomenclature below.

A physical point: $x \in \Omega$ where Ω is the domain, usually of an image.

An image: $I \colon \Omega^d \to \mathbb{R}^n$ where n is the number of components per pixel and d is dimensionality. A second image is J.

Domain map: $\phi \colon \Omega_I \to \Omega_J$ where $\to$ 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: $\underset{b}{\rightsquigarrow}$ e.g. B-Spline field mapping J to I.

Diffeomorphic mapping: $\leftrightsquigarrow$ these maps should have an accurate inverse that is computed in the algorithm or can be computed from the results.

Composite mapping: $\phi = \phi_1(\phi_2(x))$ is defined by $\leftrightsquigarrow\rightarrow$ where ϕ_2 is of type $\leftrightsquigarrow$.

Not invertible: $\nleftrightarrow$ indicates a mapping that is not invertible.

Image warping: For example, $\rightarrow J$ represents applying affine transform $\rightarrow$ on image J. $\rightarrow J = J(A(x))$.

Similarity measure: $\underset{s}{\approx}$ or $\approx_s$ indicates the metric s that compares images.

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.

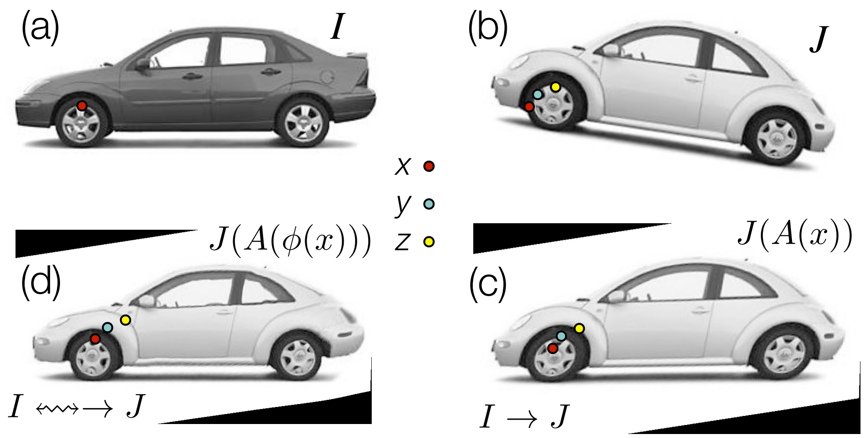

Figure 1: Define x in Ω_I and z in Ω_J as the same material point but existing in different domains. The point y is in a domain that is intermediate between Ω_I and Ω_J. The standard approach in the ITKv4 registration framework is to map image J (b) to image I (a) by first identifying the linear transformation, $\rightarrow$, 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 $I(x) \;\approx_{\text{mi}} J(A(\phi(x))) = J_{\text{Affine}}(y) = J(z)$. The image $J_{\text{Affine}}(y)$ represents J after application of the affine transformation A i.e. $J(A(x))$.

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 arrowhead indicates its range. Finally, we denote the similarity metric as $\approx$ which indicates a sum of squared differences (the default similarity metric). ITK^4 supports metrics such as mutual information, $\underset{\text{mi}}{\approx}$, or cross-correlation, $\underset{\text{cc}}{\approx}$. We will use this nomenclature to write schematics for registration applications in the following sections.

4 Overview of the unified framework

The key ideas for ITK^4 registration are:

1. Registration maps can be applied or optimized through the *itkCompositeTransform* which chains transforms together as in Figure 1.
2. Each ITK^4 transform 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.
3. ITK^4 metrics are applicable to both types of transforms and may optimize over dense or sparse samples from Ω. Metrics may be multi-channel (e.g. for registering RGB or tensor images).
4. The optimization framework is multi-threaded and memory efficient to allow high-dimensional transformations to be optimized quickly on multi-core systems.
5. The ITK^4 optimization framework comes with parameter setting tools that automatically select parameter scales and learning rates for gradient-based optimization schemes. These parameter setting tools use physical units to help provide the user with intuition on the meaning of parameters.

Below we will discuss (1) gradient-based optimization within the framework, (2) techniques to estimate optimization parameters for arbitrary metric and transformation combinations and (3) a generalized diffeomorphic matching approach.

4.1 Optimization Framework

The general ITK[4] optimization criterion is summarized as:

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

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 ITK[4] , 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 $p = (p_1, p_2 \ldots p_n)$. 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),

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

This equation provides the metric gradient specified for sum of squared differences (at point x) but similar forms arise for the correlation and mutual information [23]. Both are implemented in ITK[4] 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 ITK[4] , 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 $I \approx\rightarrow J$, $I \underset{\text{mi}}{\approx} \overset{\rightsquigarrow}{b} \rightarrow J$, $I \underset{\text{cc}}{\approx} \leftrightsquigarrow\rightarrow J$ or, mixing similarity metrics, $I \approx_{\text{cc}}\leftrightsquigarrow\approx_{\text{mi}}\rightarrow J_i$.

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

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

where λ is the overall learning rate and the brackets hold the vector of parameter updates. Note that, as in previous versions of ITK, 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, $\frac{d\phi}{dp}|_x$ corresponds to *ComputeJacobianWithRespectToParameters(mappedFixedPoint, Jacobian);* . Note that it is evaluated at point x not at point $\phi(x,p)$. We then use *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 J as $J_w(x) = J(\phi(x,p))$. Then,

$$\begin{aligned} \frac{dJ_w}{dx} &= \frac{dJ(\phi(x,p))}{d\phi}\frac{d(\phi(x,p))}{dx} \\ \frac{dJ(\phi(x,p))}{d\phi} &= \frac{dJ_w}{dx}\frac{d(\phi(x,p))}{dx}^{-1} \end{aligned} \tag{3}$$

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);* .

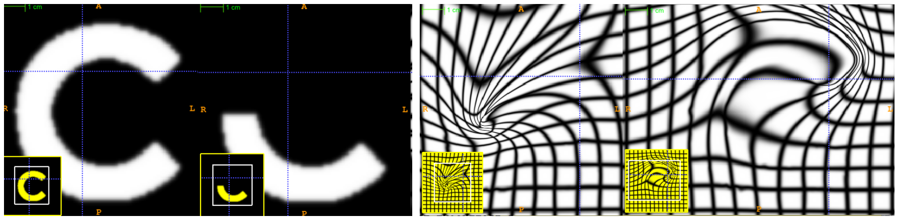

Figure 2: An ITK diffeomorphic mapping of the type $I \leftrightsquigarrow J$. The "C" and 1/2 "C" example illustrate the large deformations that may be achieved with time varying velocity fields. In this case, the moving (deforming) image is the 1/2 "C". The right panels illustrate the deformed grid 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 seconds. The images each were of size 150×150.

4.2 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 [21].

Denote the unscaled gradient descent update to p as $\triangle\, p$. The goal is to rescale $\triangle\, p$ to $q = s \cdot \triangle\, p$, where s is a diagonal matrix diag$(s_1, s_2 \ldots s_n)$, such that a unit change of q_i will have the same impact on deformation for each parameter $i = 1...n$. As an example, we want $\|\phi(x, p_{\text{new}}) - \phi(x, p_{\text{old}})\| = constant$ 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 $[\frac{\partial M}{\partial p_1}, \cdots, \frac{\partial M}{\partial p_n}]$ as $\frac{\partial M}{\partial J}\frac{\partial J(\phi(x,p))}{\partial \phi}[\frac{\partial \phi}{\partial p_1}, \cdots, \frac{\partial \phi}{\partial p_n}]$. 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 diag$(s)\frac{\partial \phi}{\partial p}$. We identify the values of 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 ITK[4] feature. However, a preliminary evaluation is 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 [21]. 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.

4.3 Diffeomorphic mapping with arbitrary metrics

Beg proposed the Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm [4] 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 ITK[4], we implement an alternative to LDDMM that also uses a time varying field and an *ode* but minimizes the following objective function:

$$E(\mathbf{v}) = M(I, J, \phi_{1,0}) + w\int_0^1 \|\mathcal{L}v_t\|^2 \mathrm{d}t\ . \quad (4)$$

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 linear operator $\mathcal{L}$. ITK^4 uses Gaussian smoothing which is the Green's kernel for generalized Tikhonov regularization [24]. 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 [23] with a few adjustments to accomodate diffeomorphic mapping. Figure 2 shows an ITK result on a standard example for large deformation registration. We will evaluate this diffeomorphic mapping, along with parameter estimation, in the following section.

5 Evaluation

We first investigate the ability of our automated parameter estimation to facilitate parameter tuning across metrics. We then compare ITK^4 with an open-source ITK^3 registration application. In the future, the latest evaluation numbers will be available at: ITKv4 latest evaluation results.

Parameter estimation across metrics. ITK^4 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,

$$I \approx \leftrightsquigarrow \rightarrow J, \qquad I \approx_{\text{cc}} \leftrightsquigarrow \rightarrow K, \qquad I \approx_{\text{mi}} \leftrightsquigarrow \rightarrow K$$

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 3 shows the candidate images for this test.

As shown in figure 3, 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.

Comparison against ITK^3. We compare the ITK^4 registration against an ITK^3 registration suite BrainsFit (nitrc.org multimodereg). We present preliminary, encouraging evaluation results for this approach to gradient descent with both affine and deformable registration in Figure 4. The dataset consists of ten elderly and demented subjects with manual labels of brain parenchyma. Of importance is that the ventricles are not included in the parenchyma. Large deformation is required to match ventricles and, as such, this evaluation provides some insight into the benefit of the new ITK^4 diffeomorphic matching.

6 Discussion and future work

ITK is a community built and maintained toolkit and is a public resource for reproducible methods. The updated ITK^4 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 ITK^4 takes valuable steps toward the direction of automated parameter selection. By the time of the workshop, we intend to have a more extensive series of benchmark performance studies completed on standard datasets and hope that presentation of this work will provide a valuable foundation for future work. The number of possible applications exceeds what can possibly be evaluated via the ITK core. 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. Future work will

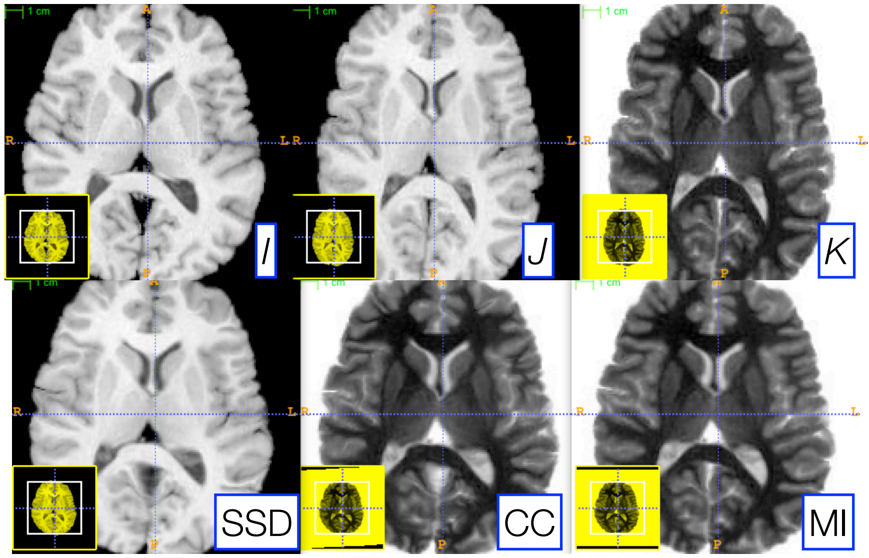

Figure 3: Three references images, I (left), J (middle top), and K (right top), are used to illustrate the robustness of our parameter scale estimation for setting consistent parameters across both metrics and transform types. K is the negation of J and is used to test the correlation and mutual information 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.

enhance the depth and breadth of this documentation as well as seek to optimize the current implementations for speed and memory. With this effort, the user community will be capable of efficiently implementing novel applications and even algorithms based on the ITK^4 framework.

The overall purpose of the registration refactoring for ITKv4 was to simplify the user experience and to accelerate and improve performance. Below, we summarize current progress toward the original goals of the team where bold-faced quotations are taken from our contract application:

Image registration should be achievable in one step: This overarching goal is best illustrated by Modules/Registration/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 (Modules/Core/Transform/include/itkCompositeTransform.h). Thus, this framework provides unprecedented ability to perform complex and staged registration mappings. Furthermore, as shown in a paper accepted to the Workshop on Biomedical Image Registration 2012 (WBIR2012), the frameworks automated parameter scaling (e.g. Modules/Numerics/Optimizersv4/include/itkRegistrationParameterScalesEstimator.h) vastly reduces the difficulty of tuning parameters for different transform/metric combinations (see https://github.com/stnava/ITKv4
A second WBIR 2012 submission by Tustison & Avants shows that the ITKv4 framework also allows a BSpline-based extension to the top-ranked SyN registration method that may outperform the original algorithm. Thus, the contributed technology not only updates v3 but also provides technological advances, including new SyN implementations and an improvement to the Large Deformation Diffeomorphic Metric Mapping (LDDMM) algorithm of Michael I. Miller.

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

Our revisions to the ITK transform hierarchy validated and extended the v3 transforms for thread safety and applicability to not only vectors but also tensors. Reorientation steps necessary for diffusion tensor mappings are also included.

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 BSpline or affine registration algorithm. Thus, full plug-and-play support exists across transforms. BSplines also work with the automated parameter scale estimation that we contributed.

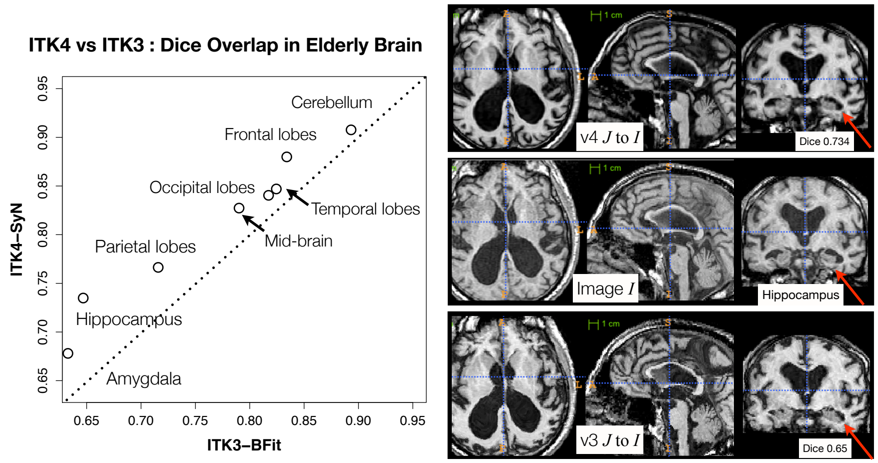

Figure 4: We compare an ITKv4 composite schema as $I \ \approx_{\text{cc}}\leftrightsquigarrow\approx_{\text{mi}}\rightarrow\ J_i$ for mapping a set of $\{J_i\}$ images to a template I to a v3 schema: $I \ \approx_{\text{mi}}\rightsquigarrow_b\approx_{\text{mi}}\rightarrow\ J_i$. We use this schematic in a registration-based segmentation of multiple brain structures in an elderly population as a benchmark for algorithm performance, similar to [18]. Example large-deformation results from the dataset are at right. The largest improvement in performance is within hippocampus, where a 13% improvement in v4 is gained. Overlap improvement from v3 to v4, quantified via paired t-test, is significant. The example pair of images will be included in v4 for regression testing.

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 metric or through the multi-channel metric all of which were contributed in 2012.

Flexible choice of optimizers and transformations: See ITK/Modules/Numerics/Optimizersv4/ for a set of optimizers that are applicable to both linear and deformable transformations, which include convergence monitoring and which enable 2nd order optimization (itkQuasiNewtonOptimizerv4), multiple objective optimization (itkMultiGradientOptimizerv4), or global optimization (itkMultiStartOptimizerv4).

GPU and multi-core acceleration will open up new applications for image registration: See .//Modules/Registration/GPUPDEDeformable and the new metric framework in Modules/Registration/Metricsv4/ which exploits N cores to accelerate metric, gradient and optimization steps.

We now comment on our progress toward overall project themes.

Simplification: The automated parameter scaling routines developed as part of the v4 project have now been tested in a variety of contexts and shown to reliably map the gradient parameters into a flat, linear space with well-defined units (image spacing). For standard registration methods, this means that only one parameter tunes the performance and the range of the parameter exploration space is much narrower than in previous versions of ITK. Typically useful gradient descent parameters take on optimal values lying within the 0.5 - 1 range and, remarkably, this is the case across multiple different transformation models including translation, affine, B-Spline (a significant speed-up and improved performance over what existed previously) and diffeomorphic mappings. This simplification will drastically reduce the user time needed to set up a registration pipeline.

Acceleration: 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. Furthermore, our GPU demons implementation is currently available in v4 for testing and further refinement.

Improved performance: An evaluation of the revised registration pipeline showed considerable improvement in performance in comparison to the best registration available in previous versions of ITK. In a brain mapping study, we found that ITKv4 significantly improves brain overlap in a number of structures when compared to ITKv4.

Other technical advances: A multivariate optimizer was contributed. A multivariate metric and multi-channel extensions to the image metric framework are currently being refined for inclusion to v4 by summer.

The multi-level and multi-stage v4 registration pipeline was fully refactored such that it is easily included in the SimpleITK scripting environment and the NAMIC-sponsored Slicer project.

The symmetric normalization algorithm, in two forms, was contributed and finalized in the v4 registration pipeline. Both a fast version that implements digital diffeomorphisms is available as well as a more time-consuming continuous diffeomorphic implementation. Both methods are easily combined with the host of new similarity metrics available in v4 which include: Mattes mutual information, global correlation, fast neighborhood correlation, and two intensity difference metrics.

We also contributed a novel B-Spline based diffeomorphic method which will be presented, along with a separate paper that provides an overview of the new framework, at the Workshop on Biomedical Image Registration in summer 2012.

Finally, we reiterate some of our contributions through quantitative metrics:

- ⋆ 22 new multi-threaded image registration metrics are available in v4 with a few of these being refactored. A minimum of 3 more implementing multivariate approaches will be contributed by summer.
- ⋆ 4 application-level registration methods are available for high-level inclusion in SimpleITK and related projects.
- ⋆ A new optimization framework that is multi-threaded and includes gradient descent and 2nd order (quasi-Newton) approaches with automated parameter setting specifically designed for registration.
- ⋆ All contributions are unit-tested and have greater than 85
- ⋆ An example-based tutorial for the new framework was presented at SPIE 2012 by Nicholas Tustison.
- ⋆ 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. 41 classes, in total, were impacted by this refactoring.
- ⋆ Transform adaptors that allow all ITK transforms to be used in the same multi-resolution framework.

  Vector support for the nearest neighbor and linear interpolators as well as two new Gaussian interpolators.
- ⋆ An example of vector support for image metrics is in : itkMeanSquaresImageToImageMetricv4VectorRegistrationTe

Future work will extend the approaches here with additional support for diffusion tensor registration and additional evaluation.

Deliverables

ANTSNeighborhoodCorrelationImageToImageMetricv4
ANTSNeighborhoodCorrelationImageToImageMetricv4DenseGetValueAndDerivativeThreader
BSplineSmoothingOnUpdateDisplacementFieldTransform
BSplineSmoothingOnUpdateDisplacementFieldTransformParametersAdaptor
BSplineSyNImageRegistrationMethod
BSplineTransform
BSplineTransformInitializer
BSplineTransformParametersAdaptor
ComposeDisplacementFieldsImageFilter
CompositeTransform
CompositeTransformIOHelper
ConvergenceMonitoringFunction
CSVArray2DDataObject
CSVArray2DFileReader
CSVFileReaderBase
CSVNumericObjectFileWriter
DemonsImageToImageMetricv4
DemonsImageToImageMetricv4DenseGetValueAndDerivativeThreader
DisplacementFieldToBSplineImageFilter

DisplacementFieldTransform
DisplacementFieldTransformParametersAdaptor
EuclideanDistancePointSetToPointSetMetricv4
ExpectationBasedPointSetToPointSetMetricv4
GaussianInterpolateImageFunction
GaussianSmoothingOnUpdateDisplacementFieldTransform
GaussianSmoothingOnUpdateDisplacementFieldTransformParametersAdaptor
GaussianSmoothingOnUpdateTimeVaryingVelocityFieldTransform
GradientDescentOptimizerBasev4
GradientDescentOptimizerBasev4ModifyGradientByLearningRateThreader
GradientDescentOptimizerBasev4ModifyGradientByScalesThreader
GradientDescentOptimizerv4
ImageRegistrationMethodv4
ImageToImageMetricv4
ImageToImageMetricv4GetValueAndDerivativeThreader
ImageToImageMetricv4GetValueAndDerivativeThreaderBase
InvertDisplacementFieldImageFilter
JensenHavrdaCharvatTsallisPointSetToPointSetMetricv4
JointHistogramMutualInformationComputeJointPDFThreader
JointHistogramMutualInformationComputeJointPDFThreaderBase
JointHistogramMutualInformationGetValueAndDerivativeThreader
JointHistogramMutualInformationImageToImageMetricv4
LabelImageGaussianInterpolateImageFunction
ManifoldParzenWindowsPointSetFunction
MattesMutualInformationImageToImageMetricv4
MattesMutualInformationImageToImageMetricv4GetValueAndDerivativeThreader
MeanSquaresImageToImageMetricv4
MeanSquaresImageToImageMetricv4GetValueAndDerivativeThreader
OptimizerParameters
OptimizerParameterScalesEstimator
OptimizerParametersHelper
PointSetToPointSetMetricv4
QuasiNewtonOptimizerv4
QuasiNewtonOptimizerv4EstimateNewtonStepThreader
RegistrationParameterScalesEstimator
RegistrationParameterScalesFromJacobian
RegistrationParameterScalesFromShift
SyNImageRegistrationMethod
TimeVaryingBSplineVelocityFieldImageRegistrationMethod
TimeVaryingBSplineVelocityFieldTransform
TimeVaryingBSplineVelocityFieldTransformParametersAdaptor
TimeVaryingVelocityFieldImageRegistrationMethodv4
TimeVaryingVelocityFieldIntegrationImageFilter
TimeVaryingVelocityFieldTransform

Major Fixes

KdTree
PointsLocator
CentralDifferenceImageFunction

References

[1] D. Rueckert et al. “Nonrigid registration using free-form deformations: application to breast MR images.” In: *IEEE Trans Med Imaging* 18.8 (1999), pp. 712–721.

[2] J. A. van Dalen et al. “Accuracy of rigid CT-FDG-PET image registration of the liver.” In: *Phys Med Biol* 49.23 (2004), pp. 5393–5405.

[3] Damion Shelton et al. “Teaching medical image analysis with the Insight Toolkit.” In: *Med Image Anal* 9.6 (2005), pp. 605–611.

[4] Michael I. Miller et al. “Increasing the power of functional maps of the medial temporal lobe by using large deformation diffeomorphic metric mapping.” In: *Proc Natl Acad Sci U S A* 102.27 (2005), pp. 9685–9690.

[5] Mingli Chen et al. “A simple fixed-point approach to invert a deformation field.” In: *Med Phys* 35.1 (2008), pp. 81–88.

[6] M Rex Cheung and Karthik Krishnan. “Interactive deformation registration of endorectal prostate MRI using ITK thin plate splines.” In: *Acad Radiol* 16.3 (2009), pp. 351–357.

[7] Sajjad Baloch and Christos Davatzikos. “Morphological appearance manifolds in computational anatomy: groupwise registration and morphological analysis.” In: *Neuroimage* 45.1 Suppl (2009), S73–S85.

[8] Jean-Marc Peyrat et al. “Registration of 4D cardiac CT sequences under trajectory constraints with multichannel diffeomorphic demons.” In: *IEEE Trans Med Imaging* 29.7 (2010), pp. 1351–1368.

[9] C. T. Metz et al. “Nonrigid registration of dynamic medical imaging data using nD + t B-splines and a groupwise optimization approach.” In: *Med Image Anal* 15.2 (2011), pp. 238–249.

[10] Ron Kikinis and Steve Pieper. “3D Slicer as a tool for interactive brain tumor segmentation.” In: *Conf Proc IEEE Eng Med Biol Soc* 2011 (2011), pp. 6982–6984.

[11] Andriy Fedorov et al. “Atlas-guided segmentation of vervet monkey brain MRI.” In: *Open Neuroimag J* 5 (2011), pp. 186–197.

[12] Keelin Murphy, Bram van Ginneken, and et al. “Evaluation of registration methods on thoracic CT: the EMPIRE10 challenge.” In: *IEEE Trans Med Imaging* 30.11 (2011), pp. 1901–1920.

[13] Clifford R Jack Jr et al. “Hypothetical model of dynamic biomarkers of the Alzheimer’s pathological cascade.” In: *Lancet Neurol* 9.1 (2010), pp. 119–128.

[14] Carrie E. Bearden et al. “Mapping cortical thickness in children with 22q11.2 deletions.” In: *Cereb Cortex* 17.8 (2007), pp. 1889–1898.

[15] Ralf Floca and Hartmut Dickhaus. “A flexible registration and evaluation engine (f.r.e.e.).” In: *Comput Methods Programs Biomed* 87.2 (2007), pp. 81–92.

[16] Terry S. Yoo et al. “Engineering and algorithm design for an image processing Api: a technical report on ITK–the Insight Toolkit.” In: *Stud Health Technol Inform* 85 (2002), pp. 586–592.

[17] Michael J. Ackerman and Terry S. Yoo. “The Visible Human data sets (VHD) and Insight Toolkit (ITk): experiments in open source software.” In: *AMIA Annu Symp Proc* (2003), p. 773.

[18] Stefan Klein et al. “elastix: a toolbox for intensity-based medical image registration.” In: *IEEE Trans Med Imaging* 29.1 (2010), pp. 196–205.

[19] Tom Vercauteren et al. “Diffeomorphic demons: efficient non-parametric image registration.” In: *Neuroimage* 45.1 Suppl (2009), S61–S72.

[20] Brian B. Avants et al. “A reproducible evaluation of ANTs similarity metric performance in brain image registration.” In: *Neuroimage* 54.3 (2011), pp. 2033–2044.

[21] M. Jenkinson and S. Smith. “A global optimisation method for robust affine registration of brain images.” In: *Med Image Anal* 5.2 (2001), pp. 143–156.

[22] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. “Deformable templates using large deformation kinematics.” In: *IEEE Trans Image Process* 5.10 (1996), pp. 1435–1447.

[23] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. "A variational approach to multi-modal image matching". In: *Intl. J. Comp. Vis.* 50.3 (2002), pp. 329–343.

[24] Mads Nielsen, Luc Florack, and Rachid Deriche. "Regularization, Scale-Space, and Edge Detection Filters". In: *J. Math. Imaging Vis.* 7 (1997), pp. 291–307. ISSN: 0924-9907.