A Flexible Registration and Evaluation Engine (f.r.e.e.)

R Floca, H Dickhaus

Department of Medical Informatics, University of Heidelberg, Heidelberg, Germany

*NOTICE: this is the author’s version of a work that was accepted for publication in Computer Methods and Programs in Biomedicine. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Computer Methods and Programs in Biomedicine [87, 2, (August 2007)] DOI: 10.1016/j.cmpb.2007.04.009*
A Flexible Registration and Evaluation Engine (f.r.e.e.)

R Floca, H Dickhaus
Department of Medical Informatics, University of Heidelberg, Heidelberg, Germany

Summary

We present an image registration framework which offers effective assistance for solving current registration problems. This work was motivated by the huge amount of registration problems in clinical applications and the problem of finding adequate solutions and properly comparing them. We have therefore designed a framework that supports the establishment, evaluation and comparison of registration approaches.

F.r.e.e. (Flexible Registration and Evaluation Engine) achieves a broad basis of algorithms by utilizing the insight segmentation and registration toolkit (ITK). This basis can be extended by virtually any new approach or algorithm, which then becomes seamlessly integrated into the method set of the f.r.e.e. framework. The framework offers suitable tools for an easy integration, optimization and proper evaluation of registration approaches, as well as an efficient utilization of the results in clinical routine.

The framework is currently being evaluated at the Heidelberg University Hospital, Germany. The first results were gathered with an application implemented for the Neurosurgical Department of the hospital. In these tests the framework concept, along with its specific tools, was very promising for establishing clinical applications (e.g. preoperative neurosurgical planning; registration of cardiac images) and therefore motivated further development. The ability to automatically optimize the parameterization of registration methods regarding a given test set also proved useful, allowing more concentration on scientific problems themselves and not on the laborious task of parameter tweaking. Due to implemented abstraction layers, f.r.e.e. also allows a high degree of transparency and thus good comparability of registration approaches and results.

Keywords:

Image registration; matching; evaluation; framework; parameterization; parameter optimization

Contact:

Ralf Floca
University of Heidelberg
Institute for Medical Biometry and Informatics, Department of Medical Informatics
Im Neuenheimer Feld 400, D-69120 Heidelberg, Germany
Telephone: +49 6221 56-7484, Fax: +49 6221 56-4997, Email: ralf.floca@med.uni-heidelberg.de
1 Introduction

Image registration is, as described by Maintz et al. [1], the first step in an information integration process and brings the involved modalities into spatial alignment. Therefore registration approaches are a crucial step in all image analysis tasks combining multimodal data sources [2]. Hence various application scenarios exist and, along with them, a variety of demands; e.g. in the field of neurosurgery, where image registration is employed in the process of preoperative planning, intraoperative alignment and follow-up studies.

In preoperative planning, a set of images from different modalities often has to be matched in order to get more information about vascular morphology and thus allow surgeons to optimize the intervention and minimize harming effects. In these cases accuracy is more important than the duration of the registration. In the case of intraoperative alignment it is different, because the time constraints are very strong and non rigid registration methods (NRR) are sometimes the only possibility. One important reason for intraoperative imaging is the well-known brain shift effect. Studies [3] have shown that a shift of up to 20 mm can be caused by a craniotomy. Thus brain shift can substantially reduce the efficiency of image-guided surgery systems, if it is not compensated for.

As mentioned above, there are often different classes of problems with diverse requirements like accuracy, execution performance, involved modalities or image quality. It is an optimization problem to find the most appropriate solution. A general solution would surely be appreciated, but as formally stated in the “no free lunch” theorem by Wolpert [4]: good performance in one class of problems guarantees a loss of performance in other classes. There is no a priori best solution to all problems. As noted by Avants et al. [5] this theorem can also be transferred to registration problems. Therefore a tool to help establish optimized registration approaches, to evaluate them and to help in choosing among potential solutions is desirable. This work was initially motivated by the various problems in the field of neurosurgery [6]. But the resulting framework with its generic approach is also appropriate to, and used in, other fields, such as the registration of cardiac images.

In addition, a proper evaluation of those approaches is also indispensable, especially in a clinical context [7]. The evaluation should cover two basic points. First, it should measure the quality of approaches in a contextual and reliable way. Second, it should support comparison of different approaches on an equivalent basis to ensure correct judgment and real comparability. A comparison of different approaches is not trivial, especially in the light of the lack of standardization, as pointed out in [8, 9, 10]
The f.r.e.e. framework is designed to contribute to three major topics: first, to the work of easing the optimization process of registration methods; second, to the standardization of evaluation schemes; and third and lastly to offer a reliable framework for application in clinical routine.

The remainder of the paper is organized as follows. In Section 2, we provide an overview of registration methodology and previous research on the objective of registration evaluation and standardization. The section also emphasizes the design considerations and requirements for our work. The framework itself and its major concepts will be addressed in Section 3. A short status report on the current use of the framework is given in Section 4. In section 5 the problems mentioned will be discussed and alternative solutions will be suggested. In conclusion, future steps to finalize the framework in the context of clinical environment will be pointed out in section 6. Also information concerning the license and the obtaining of f.r.e.e. can be found in the last section.

2 Background and design considerations

Registration is the process of finding a suitable transformation $T$ which allows the mapping of any point $x$ from one given information space $A$ into another information space $B$.

$$T : x_A \mapsto x_B \iff T(x_A) = x_B$$

In case of medical registration problems these information spaces $A$ and $B$ could be two images from different modalities whose information is to be integrated (e.g. registration of computer tomography (CT) and positron emission tomography (PET)) or $A$ could be an image and $B$ the coordinate system of a treatment device (so-called image to physical registration). $A$, the destination of $T$, is called the reference information space and $B$ is labeled the moving information space. There are a variety of strategies for establishing a suitable $T$. Hence several papers, e.g. by Hill et al. [11], Maintz et al. [1] or Zitová et al. [2], have attempted to categorize and review all trends and basic ideas. Maintz et al. [1] define nine basic criteria (some even subdivided) for classifying registration methods: i. Dimensionality, ii. Nature of registration basis (e.g. based on set of points or image intensities), iii. Nature of transformation (e.g. rigid or elastic), iv. Domain of transformation (local or global), v. Interaction, vi. Optimization procedure, vii. Modalities involved, viii. Subject (e.g. intrasubject registration or atlas), ix. Object (e.g. head, thorax). These criteria – indicative of the large variety of methods – build the three major pillars of a registration procedure: the problem statement, the registration paradigm and the optimization procedure. Maintz et al. also stress that these three major pillars are independent, even though many other articles do not make this distinction.

2.1 Related work
Apart from many approaches which realize specialized registration methods for one or a few given problem statements, several projects have sought to establish a toolset which enables the user to choose from different registration paradigms and optimization procedures. One example of a framework of this kind is the insight segmentation and registration toolkit (ITK) [12]. ITK is an open-source, object-oriented software system for image processing, segmentation and registration, implemented in C++. Aside from basic image processing and segmentation methods, ITK implements a registration pattern as shown in Fig. 1. Within some constraints, ITK allows developers to combine the four parts (interpolator, transformation, metric and optimizer; see Fig. 1) arbitrarily when implementing a new registration method, and thus it allows them to define the registration paradigm and optimization procedure. For the development of our work, the ITK library was used as an algorithmic basis. The concept and philosophy of the ITK library architecture (pluggable registration components and therefore the possibility to separate the three main pillars, smart pointer to enhance memory management, extensibility of input and output file formats, etc.) fit well with our own idea. Furthermore ITK provides a preliminary set of registration components that we were able to test our concept at an early stage of development. So we hope to establish a synergetic effect between the ongoing development of ITK and our own work.

There have been endeavors to tackle the problems of evaluation from several sides, in addition to efforts to develop more efficient registration methods. The spectrum ranges from essays and proposals for the standardization of evaluation [13, 14, 15, 7] or of the test data that should be used [16] to distinct projects which seek to support registration itself (e.g. ITK) or its evaluation. In addition, projects like Phyton for ITK seek to establish fast development cycles through the use of scripting technology, and products like MevisLab© [17] attempt to significantly ease the rapid establishment of new image processing methods.

There have been also efforts to unify the test data basis, e.g. by using in vivo data like those from the Visible Human Project [18] or simulated data (e.g. from the internet project BrainWeb [19]). Whether these approaches are also adequate for real clinical cases is open to debate, but at least the results of evaluations based on them allow comparisons. The question whether there is a possibility or method to derive clinical conclusions from these evaluations is discussed in several papers, e.g. by Fitzpatrick [15] or Lehmann et al. [7]. Furthermore there have been efforts like the online collection of evaluation results based on a preset data basis [8].

The aim of this work is not to compete or supplant the previously mentioned endeavors, but to supplement them and allow a reasonable bundling of their ideas.

Regarding evaluation it is important to precisely define the nature of the evaluation study and its objectives. Buvat et al. [13] have defined levels of evaluation in medical image processing based on the work of Fryback et
al. [20]. In this classification, levels range from very basic technical feasibility studies (Level 1) up to full-scale societal efficacy studies (Level 6). The current version of f.r.e.e. focuses on the first two evaluation levels. Level 1 (evaluation of technical efficacy) is a feasibility study for a given component or approach, demonstrating that it produces “reasonable” results (with respect to a priori knowledge regarding the expected results) and determining proper parameters in a specific context. The second level (evaluation of diagnostic accuracy) measures the performance of the approach in relation to the problem it is intended to solve. This includes characterizing and comparing the performance of methods. Testing at both levels ensures that the three main pillars are combined to form a registration method, which in turn provides a sound basis for any clinical applications to build on.

2.2 Design considerations

The problems presented in the introduction (lack of standardization; lack of separation between methods and data; limited impact of research on clinical routine) were the main motivations for designing a flexible and expandable framework, which is also reflected in the name: Flexible Registration and Evaluation Engine (f.r.e.e.).

The following four requirements have been identified and serve as guidelines for the development:

1. Versatility
2. Sustainability
3. Sound test bed and evaluation basis
4. Exchangeability and comparability

To accomplish requirement 1 the system should be able to flexibly combine concepts and algorithms in order to establish registration methods by using independent main pillars as stated by Maintz et al. [1]. Unlike the original ITK, the flexibility does not end with the compilation of the program. This flexibility should not be achieved by using simply a scripting language, in order to suit the individual needs of different user groups. A script-based approach may be well-suited to a scientific environment, but when it comes to integration into daily routine, it would be clearly out of scale and not appropriate for the common clinician. A kind of semantics, helping to find reasonable combinations, is also desirable. The second requirement is an open system with embedding facilities, in order to allow the easy integration of new components and ideas, so as to ensure applications that can more easily evolve with scientific progress. To establish an evaluation basis (requirement 3), protocols and tools are needed that allow analysis, optimization and comparison of the efficiency of components or registration methods. To avoid problems of comparability, as stressed by several authors [8, 9,
10], it should be possible to exchange and import registration methods as well as test data to achieve exchangeability (requirement 4) of ideas and comparability of results.

It is important that scientific results have the highest possible positive impact on daily clinical routine. Therefore, one of the objectives of f.r.e.e. is to assist from development to deployment and to offer support at all steps in between. This was considered in the framework design by defining different target user groups (developers, scientists, clinicians) with individual requirements.

The main focus of the developers is the engineering of new registration algorithms and components. They need an open system for the quick integration of new component ideas. They want to achieve short development cycles and they also need evaluation facilities for the very basic Level 1 evaluations [13] (see section 2.1).

Scientists are focused on solving new matching tasks. They assemble and optimize new registration methods, and finally there has to be the possibility to test, compare and optimize the different approaches.

Clinicians want an application for solving problems in daily routine. Therefore they require a program with good usability within their workflow. It should be a streamlined and simple application, which does not offer confusing options for adjustment but has an intuitive front end and, preferably, evolves with the user’s demands. This would mean only minor changes in workflow when new problems need to be solved.

To serve the idea of versatility, sustainability and comparability, f.r.e.e. is implemented in ISO-C++ [21] and platform/compiler independence was also a design requirement. Till now f.r.e.e. was successfully compiled with Microsoft Visual C++ 6.1©, Microsoft Visual C++ 2003©, Microsoft Visual C++ 2005© and GCC 3.4.2©.

3 System description

F.r.e.e. consists of several main parts. These parts and their interaction are illustrated in Fig. 2. The following sections describe these main parts and their general functionality. The controllers and the generic setup are explained in section 3.1. Section 3.2 addresses the issue of adaptation and sections 3.3 and 3.4 cover evaluation and optimization aspects.

3.1 Controller concept and generic setup

The requirement “versatility” is established by a controller concept which wraps all registration components and serves as a generic interface. This interface defines a unified set of possible procedure calls that allows the framework to interact with the controller and make use of its controlled component. The controller creates its assigned component and manages the data retrieval and data actualization.
Each controller has a specific profile. Among other things, this profile defines the structure of the generic setup part that corresponds to the controller. The profile of a controller (an XML (eXtensible Markup Language) [22] template shown in Fig. 3) is structured into three thematic sections. Section 1 contains the general information (unique ID of the controller, derivation information, etc.) about the component. Section 2 defines the correlated generic setup part (e.g. parameters and their default values, possible sub-components). This information will be used to generate the corresponding generic setup part (Fig. 4). The information in section 3 defines all requirements of a component in relation to other parts of the setup. Hence this section is responsible for establishing an ontology, which describes the semantic coherence between components. For the definition of requirements or constraints, any information from the first two sections can be used as criteria (e.g. the ID of a component, existence or value of a parameter, etc.). An example for such a scenario which uses the ontology is an assisted designing of generic setups (e.g. an editor only offers optimizers that are able to handle quaternion-based rotations, if the user has chosen a quaternion-based transformation).

To achieve even more complex registration approaches, any single registration method (e.g. multiresolution rigid registration) can be combined with others to define a series of consecutive registration steps. In this case f.r.e.e. is able to update registration information and media (e.g. reference point sets, regions of interest (ROIs), segmentations, image files) to assure validity in consecutive steps. These media are also components and therefore bound to controllers, which process the update.

The controller concept also handles the integration of new components and registration strategies. Therefore, a controller can be integrated dynamically into the framework at runtime. The framework will look for any controller provided for import of such new elements and, if it finds one, add an instance of this controller to the controller pool. Hence it adds the ability to create and handle a new idea by using the associated controller. Thus applications using the framework do not need to be recompiled to extend their method inventory.

In order to describe a complete registration method, a structured data format was defined and named “generic setup”. A generic setup can be stored in a well defined XML file including an associated schema definition file to ensure validity. Generic setups define the components to be composed, their combination and the setting of their parameters to represent a registration method which can be executed by f.r.e.e. (see Fig. 4). The ontology of the generic setup (combination of components, number and type of parameters) is not predetermined. Its complexity depends on the components used and is handled by the corresponding controller and their profiles.

Every part of a generic setup can be addressed and accessed by f.r.e.e. in an XPath styled manner [23]. This feature is important for example for the controller ontology definition and is utilized by the adaptation facility (see section 3.2).
3.2 Separation of methods and data

A careful separation of case-specific data (e.g. images used, ROIs, segmentation information or coordinates of reference points) and registration information (basic concept described by a generic setup) is also important in order to ensure the design requirements of exchangeability and comparability. Mixing the two would become a problem in terms of exchanging either the test data or the registration concepts individually. This is a problem stressed in several scientific papers, but nonetheless rarely handled properly, as also reported by these groups (cf. [8, 9, 10]).

The generic setup only contains the registration concept. All case-specific information is stored per case in an adaptation item. This adaptation information is used to customize the generic setup to the single case. It is possible for the level of adaptation to range from a simple definition of the images involved to component-sensitive changing of parameter values. Like the generic setup itself, an adaptation is also structured as an XML file and can be evaluated by a schema definition file. After the adaptation of a generic setup, the case-specific adapted setups can be utilized by f.r.e.e. to compute a registration of the corresponding cases.

For ensuring the meaningful use of the framework, the exchange of test cases and algorithms is just as important as the exportation of results. The main result information is a vector field of the image space. It defines the required translation of every image point to achieve the registration of the floating image and the reference image. The standard orientation of the vector field is an inverse transform (from reference to floating image), but it can also be exported reversed. F.r.e.e. also offers a tool to automatically compute/extract other result information (e.g. the matched image itself, difference images, grid images; see Fig. 5 + Fig. 6).

3.3 Test series and evaluation

To provide the basis for a proper evaluation/comparison, f.r.e.e. logs the whole registration process. The statistical information is stored in XML files. The content of the logs (e.g. duration of optimization steps, trend of parameters, etc.) depends on the controllers of all the components used. Each controller is able to log information down to the level of each individual registration step. Using XSLT (EXtensible Stylesheet Language Transformation) [24], these log files can be converted into virtually any file format, e.g. they can be used to produce a file of comma separated values (CSV) or MS Excel files (see Fig. 7).

For testing and evaluation f.r.e.e. provides the possibility to group adaptation items (section 3.2) into adaptation lists. These lists can be used to generate work lists or batch files, which can be utilized to process a huge amount of cases with a given generic setup in an automatic manner, to generate statistics or to perform evaluations.
The current version of f.r.e.e. and its protocols focus on the first two levels of evaluation as mentioned in section 2. The technique of test series is used for different test schemes, such as the validation criteria listed in [25] (e.g. consistency tests (CT) and precision tests (PT)) or for comparison tests and quality tests (QT). Other test concepts are possible and easy to integrate using the registration protocols (or a conversion) and the test series function. CTs and PTs are Level 1 evaluations and measure the integrity of a method and the internal correctness respectively. Fig. 8 shows such a test/comparison between two mutual information (MI) metrics (one based on the work of Wells et al. [26], the other on that of Mattes et al. [27]). But even if the necessary components are known, their proper parameterization creates a vast range of possible combinations. How f.r.e.e. approaches this problem will be explained in section 3.4. QTs are settled within Level 2 evaluations and are used to test the robustness and the accuracy of an approach with a given set of problem relevant data. The accuracy is acquired according to [25] by comparing with a ground truth. This ground truth can be specified either by defining reference points or by a reference transformation field. All tests can also be carried out as comparison tests, thus different approaches are automatically adapted by the same adaptation list, and then the tests are computed. Therefore the results of comparison tests allow the comparison of different properties like robustness, performance or precision, based on identical data. Within f.r.e.e.’s paradigm an evaluation criterion is a metric, but not a metric evaluating the transformation of information, e.g. images of an image-based registration method. It is about evaluating the quality of the whole registration process and computing a measure of that quality. These metrics of process quality are also integrated into all the basic concepts of f.r.e.e. (e.g. the controller concept). Thus the set of evaluation criteria can be extended, as long as you are able to describe evaluation cost functions. A newly added evaluation criterion can be utilized like any other component, by using it within a generic setup (see section 3.4).

In the context of performance optimizations, a specific plot type is also useful. This plot focuses on the trade-off between a criterion and duration. An example shown in Fig. 9 plots the trade-off between accuracy and duration and shows that a limitation of iteration steps up to 150 would cause an additional inaccuracy in rotation of up to 0.2° along the x axis (less along the other axes) but would also decrease the processing duration by 149 sec (41%). Characteristics of the generic setup used (Variant 1) are demonstrated in Table 1 (see section 5.).

### 3.4 Optimization of parameterization

Finding the proper parameters for a given registration method can often only poorly be covered by generic rules and is more a heuristic process. F.r.e.e. utilizes the adaptation lists to support the search for optimized parameters.
in an automated manner. Finding a proper parameterization of a generic setup involves finding the right way to change a generic setup into a setup for the particular problem statement. In other words, it involves finding the proper transformation of an information space. Thus the problem of proper parameterization can be understood as being very similar to a registration problem in terms of design. This paradigm describes in which way f.r.e.e. approaches the problem. Fig. 10 shows a scheme of the parameterization problem. As in the registration scheme (Fig. 1) a transformation, a metric and an optimizer are needed in order to find the proper parameterization. (The analogy to the ITK concept is intended to facilitate the understanding of the concept.)

The transformation of a generic setup is performed by the component setup transform. Using this component, any value within a generic setup can be defined as a parameter of the transformation. Therefore the dimensionality of the optimizer search space can arbitrarily be defined via the chosen parameters. Changes of the transformation parameters will be mapped to the associated generic setup to generate the transformed setup.

The metric uses a test set specified by a given adaptation list to measure the quality of a transformed generic setup and the registration method respectively. The measures and therefore the optimized criteria depend on the chosen metric. The optimizer uses the measurement of the metric to adjust the setup transform. This repeats until at least one stop criterion is fulfilled. One optimizer available is a simple exhaustive search within a defined search space. This is very useful if the “topology” of the search space or the impact of an isolated parameter (set) on the evaluation criteria is to be analyzed. More sophisticated (and time-saving) optimizers are also available (see section 3.5.). In addition it is possible to define constraints (equality and inequality constraints) for each parameter to sensibly restrain the search space.

All components for the parameter optimization are integrated into the controller concept. Hence a parameterization approach can be described by a generic setup and the set of components can be extended.

### 3.5 Supported approaches and components

In its current form, f.r.e.e. supports nearly all (image-based) registration methods available in ITK version 3.2.0. The field of rigid registration is covered by single and multi-resolution approaches, which offer a variety of transformations (rigid, projective or affine) and can use different optimizers and metrics as components of the registration. Non rigid registration can currently be represented by a finite element model [28] (FEM); a B-Spline model [29]; the “demon” method by Thirion [30]; and by establishing a controller for the poly rigid transformation concept of Arsigny et al. [31]. Available figures of merit are difference metrics, correlation metrics, metrics based on mutual information (MI-metric) (c.f. the work of Viola & Wells et al. [26] or of Mattes et al. [27]), kappa statistic metric [32] and Kullback Leibler metric [33]. Optimizers are represented by a variety
of gradient descent optimizers (e.g. regular step gradient optimizer), two versions of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizers [34], 1+1 evolutionary strategy ((1+1)-ES) optimizer [35], Levenberg Marquart optimizer [36], Powell optimizer [37], simultaneous perturbation stochastic approximation optimizer (SPSA Optimizer) [38] and a downhill simplex optimizer by Nelder and Meade [39]. Each allows a registration in multiple resolutions (coarse to fine) and the definition of setup parameters depending on the resolution level.

The following components are available for parameter optimization: setup transform, downhill simplex optimizer by Nelder and Meade, Powell optimizer, (1+1)-ES optimizer, SPSA optimizer and exhaustive search optimizer (iterating through the whole search space). Three metrics are currently available, all of which are based on the definition in [25] (see also section 3.3). The consistency metric estimates the limits of registration accuracy by evaluating the internal inconsistency of a registration method. The precision metric measures how reproducible the results of a method are, and whether it is prone to random fluctuations. The accuracy metric measures the error between the registration results and given references (these may be whole transformation fields or explicit reference points). All metrics return a measurement vector (consisting e.g. of minimum / maximum registration error, mean error, variance, average registration duration, number of aborted registrations, etc.). The vector elements can be arbitrarily weighted for parameter optimization.

For the export (and import) of image files, f.r.e.e. supports all appropriate data types readable by ITK (e.g. DICOM and ANALYZE formats). Like the components of f.r.e.e. itself, the supported file formats can be extended by using dynamic linked libraries (DLLs).

4 Status report

The controller concept was used to seamlessly integrate over 60 components (thereof 24 own developed components; that are amongst others the components for parameter optimization and special metrics for registration of cardiac images) in order to compose or optimize registration approaches. These components can be directly put into action by means of a generic setup editor we have developed, which utilizes the ontology of the controller concept to better support the design of generic setups. This editor can also be used to define parameter optimization strategies (also a generic setup) and to assemble test series and adaptation lists. A well-documented framework interface with controller examples also helps in the task of establishing new controllers and components.

As a feasibility study, an embedded version of f.r.e.e. was used to implement a matching environment [6] for the Neurosurgery Department of the University Hospital at Heidelberg. The first results gathered with this
application motivated the further development of the framework as presented here. Furthermore f.r.e.e. is currently being used in a scientific environment to support cardiac image analysis and is planned to be used in clinical routine, if the results of the evaluation prove satisfactory. For the neurosurgical application f.r.e.e. was also used to optimize and evaluate the most suitable generic setup. For the optimization, 100 adaptation items/cases were used, and promising generic setups were found. Characteristics of setups (Variant 1 and Variant 2) for preoperative planning registration with rigid transformation are explained in Table 1. For these characteristics 3D DICOM images were used, consisting of more than 100 slices, with a slice resolution of 256x256 pixels. These image couples were taken from clinical practice (pre- and intraoperative images of neurosurgical interventions). To broaden the test sets for rigid registrations, f.r.e.e. was used to generate synthetic image couples from given clinical images, by using random artificial transformations. The allowed scale of transformation was determined by studying the transformations of real test cases.

Table 1 shows the accuracy of Variant 1 and Variant 2. The accuracy was evaluated via quality tests (see section 3.3) utilizing a test set of 30 adaptation items/cases. Manually defined reference points were used as the ground truth in these tests.

The results (e.g. the two setups shown in Table 1) were able to be brought directly into a productive environment by simply adding the setup files to the clinical application. The application front end itself was customized according to the needs of the physicians in the Neurosurgical Department of Heidelberg University Hospital.

5 Discussion

The interfaces and framework design proved to be helpful in integrating and realizing a large number of components on the developer level. For this user group it is planned to enhance the integration support by introducing code templates for easy creation of new controllers, thus facilitating the integration of new components. Furthermore, the communication between elements of f.r.e.e. and embedding applications is to be expanded. This would allow even greater extensibility of existing applications.

On the scientific level the tools and tests presented proved to be very useful in establishing and optimizing proper registration approaches. The ability to automatically optimize a generic setup for specified criteria proved to be a timesaver as expected. Hence it increases the amount of time that can be spent on two of the major tasks of this user group. The first is to develop a broader test bed by incorporating more components. This became most obvious in the field of NRR especially in the context of neurosurgical images. Here approaches like tissue specific constraints should be incorporated to achieve e.g. a satisfying compensation of brain shift near the ventricles. Thus new, highly sophisticated NRR algorithms and optimizing strategies are prioritized, the latter
also being desirable for better parameter optimization. The second task is to establish and test further evaluation protocols within f.r.e.e.; e.g. the strategy introduced by Lehmann et al. [7]. At the current point of development we have ensured high flexibility in data used, methods and evaluation of the results. The quality of test data and the appropriateness of evaluation methods, other then those currently established in f.r.e.e., are the responsibility of the individual user. We are convinced that it is important not only to deliver the concepts that allow the exchange of data and results (where we hope to have made an important first step), but also to ensure real comparability by showing evaluation techniques translated into the “language” of f.r.e.e.

Its technical basis having been established and laid out by the described framework, the plans for the next project phase of f.r.e.e. focus on the evaluation of various further different medical problems to test and improve the flexibility of the concept and to ensure its best possible applicability in the clinical environment. A further evaluation of the given tools and protocols to improve their applicability for the different user groups is also planned.

6 Conclusions

Powerful tools are required in the light of the growing importance of medical imaging for diagnostics and therapy. These tools should be easy to use for clinicians for different application purposes. As a result, the demand for registration capabilities of different image modalities has increased significantly. Though much effort has been invested in this in the scientific environment, the impact on clinical routine remains comparatively small. In order to improve this and take into account the mentioned importance of medical imaging it is crucial that high quality results can be ensured.

Our framework offers a suitable registration platform. It can be used for research and as a basis for clinical routine applications. From our initial trials, the concept of specific tools for different user groups with one underlying framework seems to be a promising solution. It has also successfully served in both scientific ad hoc tests and fully fledged applications. By virtue of its open design, it can be combined with other tools and applications, and the establishing of synergies is both anticipated and welcome. It is up to the user to decide which tasks f.r.e.e. should be utilized for: F.r.e.e. offers assistance from development to deployment, as needed. The objective is to develop f.r.e.e. into a platform for developing, benchmarking and exchanging registration approaches. This should in turn help to ensure that the approaches are useful for research and clinical routine on a broader basis than is often seen today.
In this spirit f.r.e.e. will be available as free software under the terms of the GNU General Public License. Documentation and source code can be obtained via the project website (http://www.klinikum.uni-heidelberg.de/mi/free) and are planned to be online from June 2007.
References


### Tables

Table 1: Characteristics of two generic setups used for rigid registration

<table>
<thead>
<tr>
<th>Registration</th>
<th>$R_{SD}$</th>
<th>$R_{SEM}$</th>
<th>$T_{SD}$</th>
<th>$T_{SEM}$</th>
<th>$D_{Mean}$</th>
<th>$D_{SD}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Var. 1 (real)</td>
<td>0.316</td>
<td>0.091</td>
<td>0.174</td>
<td>0.05</td>
<td>86.4</td>
<td>24.7</td>
</tr>
<tr>
<td>Var. 1 (synthetic)</td>
<td>0.009</td>
<td>0.001</td>
<td>0.354</td>
<td>0.045</td>
<td>60.5</td>
<td>12.2</td>
</tr>
<tr>
<td>Var. 2 (real)</td>
<td>0.431</td>
<td>0.115</td>
<td>0.241</td>
<td>0.064</td>
<td>53.4</td>
<td>5.8</td>
</tr>
<tr>
<td>Var. 2 (synthetic)</td>
<td>0.058</td>
<td>0.009</td>
<td>0.401</td>
<td>0.057</td>
<td>41.2</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Characteristics of rigid registration based on real image couples (real) and generated image couples (synthetic).

Var. 1 needs more computational time than Var. 2, but also exceeds Var. 2 in accuracy. Both rigid registrations consisted of a rigid transformation (rotation + translation), Mattes-MI metric and a gradient optimizer; they differ in parameterization and set up of resolution levels. $R$: Rotation error (in degrees); $T$: Translation error (in mm); $D$: Processing time (in seconds); $SD$: standard deviation; $SEM$: standard error of the mean. Characteristics of processing: Intel Pentium 4 CPU 2.8 GHz, single core; 1 GB DDR-SDRAM; Windows 2000 5.0.2195 SP4; benchmark values: Dhrystone ALU: 6369 MIPS; Whetstone iSSE2: 5265 MFLOPS.
Figures

Fig. 1: General registration scheme of ITK. The floating image is transformed (assisted by an interpolator) into a transformed image. This transformed image is evaluated by a metric using the reference image. The metric measurement is utilized by an optimizer and indicates whether the optimization process should terminate or the transformation needs further adjustments.

Fig. 2: Scheme of the main components of f.r.e.e. and their interactions.
Fig. 3: Example of controller profile saved as an XML file. Three main sections can be distinguished. 1. General information section: identifies the controller ("Controller_1") and its relationship to others ("AncestorController_1", ...). 2. Generic setup section: defines the corresponding part of the generic setup; needed or allowed subcomponents ("SubComponent_1") or component parameters ("Parameter_1"). 3. Requirement section: describes demands ("Controller_2") on other parts of the setup ("./SubComponent_1").
Fig. 4: Example of a generic setup in XML form established by using among others the profile of Fig. 4. The dark box marks the part of the setup determined by the profile of “Controller_1”. The light box selects the part determined by a sub component controller (“Controller_2_Descendant”). The choice of this controller (dashed box) was limited by the requirement section of the “Controller_1” profile.

Fig. 5: Results of a non rigid registration of a preoperative T1 MR image and an intraoperative T1 MR image in the context of neurosurgery (top row: preoperative T1 MR image; intraoperative T1 MR image; mid row: registration result; visualized transformation field via grid image; bottom row: difference between preoperative and interim result before NRR; difference between preoperative image and final image).
Fig. 6: Results of a rigid pre-registration of two angiography images; this case has mainly a translation to the left and a slight counterclockwise rotation which both are visualized by the grid image (top row: floating image; reference image; bottom row: registration result with superimposed grid image to allow quick recognition of the transform topology; checkerboard blending between the registration result and reference image).

<table>
<thead>
<tr>
<th>Iter</th>
<th>Dur. (sec)</th>
<th>Metric value</th>
<th>Rx</th>
<th>Ry</th>
<th>Rz</th>
<th>Cx</th>
<th>Cy</th>
<th>Cz</th>
<th>Tx</th>
<th>Ty</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.4</td>
<td>-0.58637</td>
<td>0.075</td>
<td>0.388</td>
<td>0.497</td>
<td>135.7</td>
<td>128.95</td>
<td>79.585</td>
<td>9.6279</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.4</td>
<td>-0.59601</td>
<td>0.081</td>
<td>0.107</td>
<td>0.513</td>
<td>134.47</td>
<td>129.75</td>
<td>79.655</td>
<td>9.2406</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.4</td>
<td>-0.63131</td>
<td>0.046</td>
<td>0.109</td>
<td>-0.556</td>
<td>134.04</td>
<td>129.17</td>
<td>79.493</td>
<td>9.3594</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.5</td>
<td>-0.62955</td>
<td>0.008</td>
<td>0.093</td>
<td>-0.570</td>
<td>133.23</td>
<td>129.7</td>
<td>79.030</td>
<td>0.5774</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.4</td>
<td>-0.70739</td>
<td>0.072</td>
<td>0.104</td>
<td>-0.606</td>
<td>133.45</td>
<td>130.95</td>
<td>79.603</td>
<td>7.4629</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.4</td>
<td>-0.74196</td>
<td>0.102</td>
<td>0.186</td>
<td>-0.6</td>
<td>132.91</td>
<td>130.81</td>
<td>79.742</td>
<td>7.9339</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.4</td>
<td>-0.72499</td>
<td>0.006</td>
<td>0.141</td>
<td>-0.675</td>
<td>133.11</td>
<td>130.34</td>
<td>79.030</td>
<td>0.2624</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.5</td>
<td>-0.75973</td>
<td>0.004</td>
<td>0.138</td>
<td>-0.693</td>
<td>132.92</td>
<td>130.7</td>
<td>79.746</td>
<td>7.6574</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.4</td>
<td>-0.73019</td>
<td>0.017</td>
<td>0.127</td>
<td>-0.625</td>
<td>132.52</td>
<td>130.95</td>
<td>79.727</td>
<td>7.1557</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.4</td>
<td>-0.73252</td>
<td>0.048</td>
<td>0.135</td>
<td>-0.576</td>
<td>132.59</td>
<td>130.2</td>
<td>79.603</td>
<td>0.6812</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.4</td>
<td>-0.77794</td>
<td>0.030</td>
<td>0.172</td>
<td>-0.692</td>
<td>132.89</td>
<td>129.53</td>
<td>79.741</td>
<td>0.9193</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7: Example of a registration protocol (a rigid transformation) converted into an MS Excel file (table in the background that contains iteration count, duration in seconds per step, metric value and transformation parameters) using XSLT. This MS Excel file is used to plot transformation parameters (translation) against the iteration count (plot in the foreground).
Fig. 8: Comparison of component performances demonstrated for two metrics based on mutual information plotted against the iteration steps. This illustrates the difference of smoothness between the two cost functions regarding the same data basis.

Fig. 9: Trade-off estimation: Rx, Ry, Rz: Average deviation of the rotation angle R derived from the final registration result. D: duration. Both plotted against the iteration steps.

Fig. 10: Scheme of the parameter optimization process of f.r.e.e.: A generic setup serves as template. The parameter to be optimized will be adjusted by a setup transform which leads to a transformed generic setup. This setup will be evaluated by a metric using a test case set (described as adaptation list). The arbitrarily weighted measurement vector of the metric will be used by an optimizer to adjust the setup transformation for the next optimization cycle, if needed.