

Combining a Visual Programming and Rapid Prototyping Platform with ITK

Jan Rexilius¹, Julien Jomier², Wolf Spindler¹,
Florian Link¹, Matthias König¹ and Heinz-Otto Peitgen¹

¹MeVis, Center for Medical Diagnostic Systems and Visualization, Bremen, Germany

²CADDLab, University of North Carolina, Department of Radiology, USA

Email: rexilius@mevis.de

Abstract. In the last few years, the Insight Segmentation and Registration Toolkit (ITK) has become one of the most frequently used open-source libraries for filtering, segmentation, and registration of medical image data. However, a major drawback is its lack of visualization and interaction functionality inside a dedicated development platform. We propose an automatic integration of ITK filters into the visual programming and rapid prototyping platform MeVisLab. Based on an XML description of a filter, an automatic code generation is utilized. Different image types and data dimensions are combined in a single MeVisLab module that can be dynamically handled at runtime. An example application using registration, segmentation, and visualization modules is created in order to show the capabilities of our integration concept.

1 Introduction

Medical image analysis and visualization are rapidly growing fields that have gained importance in several medical disciplines. As a result, the amount of images acquired as well as the variety and complexity of algorithms and their handling is increasing. The Insight Segmentation and Registration Toolkit (ITK) [1], with the first official public release in 2002, has since then become one of the most popular libraries for the analysis of medical image data. An important requirement for the development of novel image analysis techniques and their evaluation in a clinical setting is a user-friendly appliance and a dedicated user interface. Today, no prototyping platform is available that satisfactorily combines ITK with a modular visual programming interface.

Several software platforms such as Analyze [2], SCIRun [3], or VolView [4] are offered that contain ITK modules to some extent within their scope of supply. However, a full integration of ITK as well as a simple and user-friendly or even automatic wrapping of ITK functionality is still missing, and an update to a new ITK version that contains interface changes might be tedious.

Beside several software platforms, a number of medical image processing libraries, often called Medical Imaging ToolKit or Medical Imaging Interaction Toolkit (MITK), are available from different research groups [5, 6]. They combine features of ITK and a visualization library such as VTK [7] in a unified

framework. Since any graphical user interface is missing within these tools, a long period of vocational adjustment is usually required for the user. Moreover, the usage of these libraries requires advanced C++ development skills.

In this work, we present a seamless integration of ITK into the flexible research and development platform MeVisLab [8]. Therefore, we develop a code generator that automatically creates MeVisLab modules from ITK filters. In this paper, we especially focus on image registration algorithms and propose a flexible and extendable framework that can be easily integrated into a visual programming environment.

2 Methods

2.1 The research and prototyping platform MeVisLab

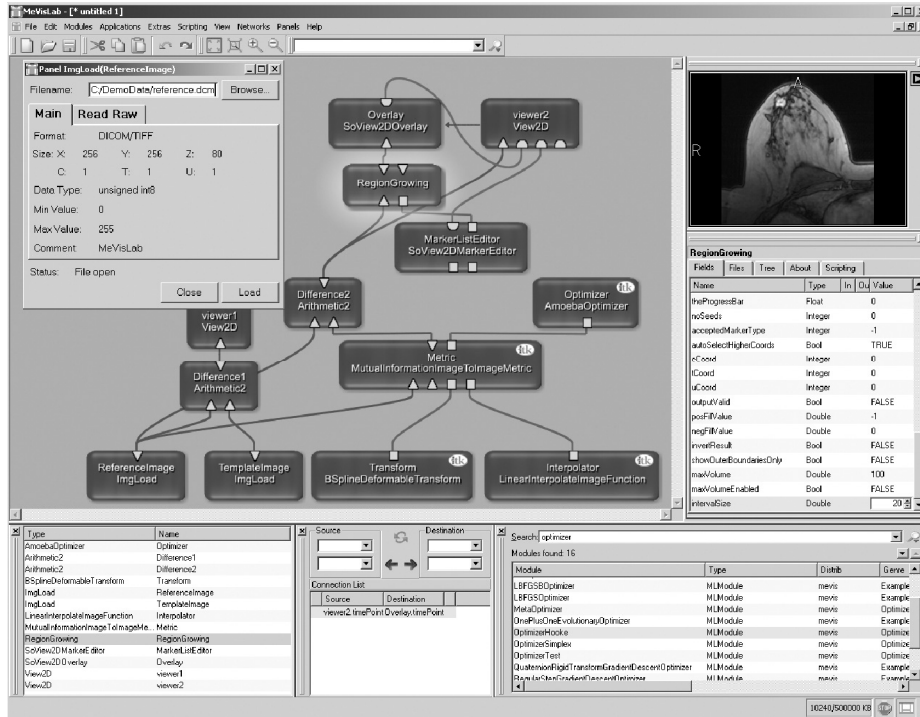
MeVisLab combines a number of design concepts for rapid prototyping. The multiplatform software (MS Windows, Linux) provides a comprehensive suite of general image processing and visualization tools for multi-dimensional medical images as well as advanced methods for specific organs and clinical tasks. Sophisticated paging, caching, and multithreading strategies are used to build a highly efficient image processing library. Furthermore, the Open Inventor 3D visualization toolkit [9] is fully integrated into MeVisLab, offering a comprehensive solution to interactive graphics programming problems. New algorithms and visualization tools can be easily integrated as new modules using a standardized software interface, with data inputs and outputs and parameters controlling the internal operations. Those functional units can be arbitrarily combined to form a network via an intuitive graphical user interface. Macro modules that allow for a hierarchical encapsulation of networks can be used to easily reuse available developments. Figure 1 shows a screenshot of the MeVisLab development environment containing an example network with several ITK modules.

Efficient design of graphical user interfaces can be achieved by an abstract, hierarchical definition language (MDL), hiding the complexity of the underlying module network to the end user (Fig. 2). JavaScript can be used to implement dynamic functionality on both the network and the user interface level. MeVisLab can be fully integrated into the radiological workflow (PACS integration). The MeVisLab environment is based on the Qt application framework.

2.2 The Insight Toolkit (ITK)

The open-source and cross-platform toolkit ITK provides an extensive collection of C++ classes for segmentation and registration of medical image data. One of its core concepts is generic programming, implemented with the template programming mechanism. Thus, a large number of data types can be easily supported and optimized for each algorithm. See [1] for further details. However, code instances for each data type and image dimension have to be generated separately. Furthermore, no support for visualization or user interface design is available.

Fig. 1. Configurable development platform MeVisLab with example network that combines ITK and MeVisLab modules. Several inspectors such as a module search interface and detailed module and connection inspectors are shown. Furthermore, the module panel of a load operator is opened inside the network.



2.3 Integrating ITK into MeVisLab

We propose a framework that integrates the advanced image analysis techniques of ITK into the visual programming environment MeVisLab. Each module can be distinguished from other MeVisLab modules using different color coding schemes. The generation of a new MeVisLab module from an ITK filter consists of two steps.

1. XML description of ITK filter. In a first step, an extensive description of the targeted ITK filter is automatically generated and saved into a XML structure, based on the actual coding styles suggested by the ITK project. The structure can be largely divided into a basic class description and a parameter description section. The general description includes several parameters such as the number of input and output fields. In this work, we focus on image-to-image filters as well as registration-specific data structures, the input/output types, or the global filter type, such as transformations and interpolators. The parameter description mainly includes parameter names, types, default values, and get/set methods.

2. Module generation. In a second step, the XML based filter description is used as input of an automatic code generation process. Depending on the type and implementation of an ITK filter, various paging and caching strategies can be used. In order to cope with different available image types and data dimensions, a single MeVisLab module is created, containing a large set of these settings. Thus, a user can easily select and combine ITK and MeVisLab modules by visual programming and interactively changing parameters, while the correct precompiled ITK filter is dynamically selected at runtime. This adds a new flexibility to ITK and greatly extends the MeVisLab functionality by a new set of algorithms.

3 Results

A large amount of the ITK toolkit functionality is successfully integrated into the MeVisLab environment. Due to a visual programming interface, no specific ITK programming expertise is required in order to use these algorithms. Each algorithm is available as a module via an intuitive user interface inside the software development environment MeVisLab. Combinations of different modules for image processing and visualization can be combined with ITK algorithms to form a network via simple field connections.

In order to show the potential of this approach, an exemplary application was developed that incorporates both algorithms of ITK and MeVisLab.

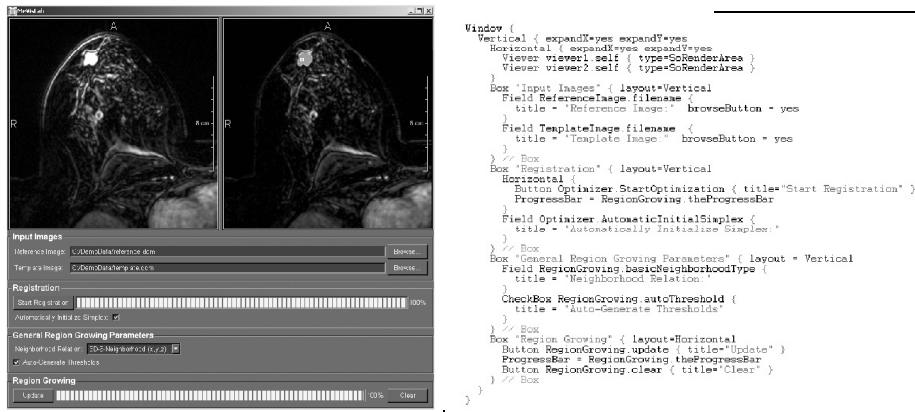
Figure 2 shows the graphical user interface as well as the complete corresponding interface description via abstract module definition language (MDL). The underlying network is shown in Figure 1. The proposed application can be used for the analysis of dynamic functional MR data sets. In a first step, a spatial b-spline based registration is performed in order to estimate the inflow and outflow of a contrast agent into tissue [10]. Then, an interactive region growing algorithm is used to segment a selected lesion.

4 Discussion and Conclusion

For the first time, a widely automatic integration of the open-source toolkit ITK into a flexible visual programming platform for medical image analysis is proposed. Our approach allows for an arbitrary combination of different modules without any recompilation, and the ability of efficient graphical user interface generation. Different parameter settings can be interactively explored. In this work, we especially focused on the integration of image registration functionality available in ITK. Our integration will be made publicly available inside the free version of MeVisLab.

Future work will focus on an automatic integration of all currently distributed ITK modules. Efficient utilization of available imaging concepts in MeVisLab will add further flexibility and performance.

Fig. 2. Example application that combines both ITK and MeVisLab algorithms based on the network shown in Figure 1. (LEFT) Screenshot of graphical user interface. The left viewer shows the difference image before and the right viewer after b-spline based registration. Furthermore, the segmentation result of a region-growing process is shown as overlay on the right viewer; (RIGHT) complete description of corresponding user interface.



References

1. Ibanez L, Schroeder W, Ng L, et al. The ITK Software Guide. Kitware Inc.; 2003.
2. Augustine K, Holmes D, Robb R. ITK and Analyze: A synergistic integration. In: SPIE, Medical Imaging 2004; 2004. p. 6–15.
3. Johnson CR, MacLeod RS, Parker SG, Weinstein DM. Biomedical Computing and Visualization Software Environments. Communications of the ACM 2004;47(11):64–71.
4. Kitware. VolView homepage <http://www.kitware.com/products/volview.html>; 2004.
5. Wolf I, Vetter M, Wegner I, et al. The medical imaging interaction toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK. In: SPIE, Medical Imaging 2004; 2004. p. 16–27.
6. Zhao M, Tian J, Zhu X, et al. The Design and Implementation of a C++ Toolkit for Integrated Medical Image Processing and Analysis. In: SPIE, Medical Imaging 2004; 2004. p. 39–47.
7. Schroeder W, Martin K, Lorensen B. The Visualization Toolkit, 3rd edition. Kitware Inc.; 2003.
8. MeVis. MeVisLab 1.0, Available at <http://www.mevislab.de>; 2004.
9. Wernecke J. The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2. Addison-Wesley Professional; 1994.
10. Kohle S, Preim B, Wiener J, Peitgen HO. Exploration of time-varying data for medical diagnosis. In: VMV; 2002. p. 31–38.