itk.org

ITK Software Guide: Introduction and Development Guidelines

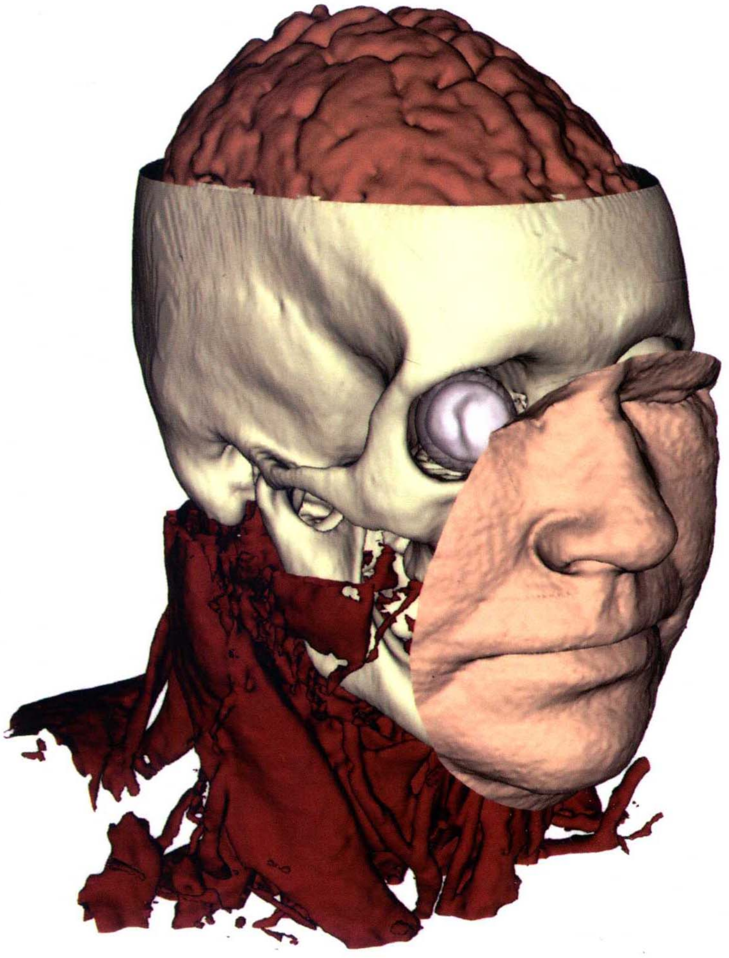
4.7

The

ITK Software Guide

Introduction and Development Guidelines

ITK 4.7



Kitware

Hans J. Johnson,
Matt McCormick, Luis Ibáñez
and the Insight Software Consortium

This project has been funded in whole or in part with Federal funds from the National Institutes of Health (NLM, NIDCR, NIMH, NEI, NINDS, NIDCD, NCI), the NSF, and the DoD (TATRC). Funding has primarily come under the direction of the National Library of Medicine, National Institutes of Health, under numerous contracts. Recently, the major revision to the toolkit, ITKv4, was made possible by NLM directed funds from the American Reinvestment and Recovery Act (ARRA).

Document created with LaTeX, using CMake as configuration manager, with a Python script to extract examples from the `ITK/Examples` directory. All code in this document compiled at the time of publication.

Printed and produced in the United States of America.
ISBN 9781-930934-27-6

The ITK Software Guide
Book 1: Introduction and Development Guidelines

Fourth Edition

Updated for ITK version 4.7

Hans J. Johnson, Matthew M. McCormick, Luis Ibáñez,
and the *Insight Software Consortium*

January 16, 2015

`http://itk.org`
Email: community@itk.org

The purpose of computing is Insight, not numbers.

Richard Hamming

ABOUT THE COVER

Creating the cover image demonstrating the capabilities of the toolkit was a challenging task.[1] Given that the origins of ITK are with the Visible Human Project, it was decided to create the cover image from the Visible Woman dataset, which is one of the more recently acquired datasets of the Visible Human Project. Both the RGB cryosections and the CT scans were combined in the same scene.

Removing the Gel. The body of the Visible Woman was immersed in a block of gel during the freezing process. This gel appears as a blue material in the cryogenic data. To remove the gel, the joint histogram of RGB values was computed. This resulted in an 3D image of $256 \times 256 \times 256$ pixels. The histogram image was visualized in VolView.[2] The cluster corresponding to the statistical distribution of blue values was identified visually, and a separating plane was manually defined in RGB space. The equation of this plane was subsequently used to discriminate pixels in the gel from pixels in the anatomical structures. The gel pixels were zeroed out and the RGB values on the body were preserved.

The Skin. The skin was easy to segment once the gel was removed. A simple region growing algorithm was used requiring seed points in the region previously occupied by the gel and then set to zero values. An anti-aliasing filter was applied in order to generate an image of pixel type float where the surface was represented by the zero set. This dataset was exported to VTK where a contouring filter was used to extract the surface and introduce it in the VTK visualization pipeline.

The Brain. The visible part of the brain represents the surface of the gray matter. The brain was segmented using the vector version of the confidence connected image filter. This filter implements a region growing algorithm that starts from a set of seed points and adds neighboring pixels subject to a condition of homogeneity.

The set of sparse points obtained from the region growing algorithm was passed through a mathematical morphology dilation in order to close holes and then through a binary median filter. The binary median filter has the outstanding characteristic of being very simple in implementation by applying a sophisticated effect on the image. Qualitatively it is equivalent to a curvature flow evolution of the iso-contours. In fact the binary median filter as implemented in ITK is equivalent to the majority filter that belongs to the family of voting filters classified as a subset of the *Larger than Life* cellular automata. Finally,

[1]The source code for the cover is available from SoftwareGuide/Cover/Source/ directory of the ITKSoftwareGuide repository.

[2]VolView is a commercial product from Kitware. It supports ITK plug-ins and is available as a free viewer or may be licensed with advanced functionality. See http://www.kitware.com/products/volview.html for information.

the volume resulting from the median filter was passed through the anti-aliasing image filter. As before, VTK was used to extract the surface.

The Neck Musculature. The neck musculature was not perfectly segmented. Indeed, the resulting surface is a fusion of muscles, blood vessels and other anatomical structures. The segmentation was performed by applying the VectorConfidenceConnectedImageFilter to the cryogenic dataset. Approximately 60 seed points were manually selected and then passed to the filter as input. The binary mask produced by the filter was dilated with a mathematical morphology filter and smoothed with the BinaryMedianImageFilter. The AntiAliasBinaryImageFilter was used at the end to reduce the pixelization effects prior to the extraction of the iso-surface with vtkContourFilter.

The Skull. The skull was segmented from the CT dataset and registered to the cryogenic data. The segmentation was performed by simple thresholding, which was good enough for the cover image. As a result, most of the bone structures are actually fused together. This includes the jaw bone and the cervical vertebrae.

The Eye. The eye is charged with symbolism in this image. This is due in part because the motivation for the toolkit is the analysis of the Visible Human data, and in part because the name of the toolkit is *Insight*.

The first step in processing the eye was to extract a sub-image of $60 \times 60 \times 60$ pixels centered around the eyeball from the RGB cryogenic dataset. This small volume was then processed with the vector gradient anisotropic diffusion filter in order to increase the homogeneity of the pixels in the eyeball.

The smoothed volume was segmented using the VectorConfidenceConnectedImageFilter using 10 seed points. The resulting binary mask was dilated with a mathematical morphology filter with a structuring element of radius one, then smoothed with a binary mean image filter (equivalent to majority voting cellular automata). Finally the mask was processed with the AntiAliasBinaryImageFilter in order to generate a float image with the eyeball contour embedded as a zero set.

Visualization. The visualization of the segmentation was done by passing all the binary masks through the AntiAliasBinaryImageFilter, generating iso-contours with VTK filters, and then setting up a VTK Tcl script. The skin surface was clipped using the vtkClipPolyDataFilter using the implicit function vtkCylinder. The vtkWindowToImageFilter proved to be quite useful for generating the final high resolution rendering of the scene (3000×3000 pixels).

Cosmetic Postprocessing. We have to confess that we used Adobe Photoshop to post-process the image. In particular, the background of the image was adjusted using Photoshop's color selection. The overall composition of the image with the cover text and graphics was also performed using Photoshop.

ABSTRACT

The Insight Toolkit (ITK) is an open-source software toolkit for performing registration and segmentation. *Segmentation* is the process of identifying and classifying data found in a digitally sampled representation. Typically the sampled representation is an image acquired from such medical instrumentation as CT or MRI scanners. *Registration* is the task of aligning or developing correspondences between data. For example, in the medical environment, a CT scan may be aligned with a MRI scan in order to combine the information contained in both.

ITK is a cross-platform software. It uses a build environment known as CMake to manage platform-specific project generation and compilation process in a platform-independent way. ITK is implemented in C++. ITK's implementation style employs generic programming, which involves the use of templates to generate, at compile-time, code that can be applied *generically* to any class or data-type that supports the operations used by the template. The use of C++ templating means that the code is highly efficient and many issues are discovered at compile-time, rather than at run-time during program execution. It also means that many of ITK's algorithms can be applied to arbitrary spatial dimensions and pixel types.

An automated wrapping system integrated with ITK generates an interface between C++ and a high-level programming language Python. This enables rapid prototyping and faster exploration of ideas by shortening the edit-compile-execute cycle. In addition to automated wrapping, the SimpleITK project provides a streamlined interface to ITK that is available for C++, Python, Java, CSharp, R, Tcl and Ruby.

Developers from around the world can use, debug, maintain, and extend the software because ITK is an open-source project. ITK uses a model of software development known as Extreme Programming. Extreme Programming collapses the usual software development methodology into a simultaneous iterative process of design-implement-test-release. The key features of Extreme Programming are communication and testing. Communication among the members of the ITK community is what helps manage the rapid evolution of the software. Testing is what keeps the software stable. An extensive testing process supported by the system known as CDash measures the quality of ITK code on a daily basis. The ITK Testing Dashboard is updated continuously, reflecting the quality of the code at any moment.

The most recent version of this document is available online at http://itk.org/ItkSoftwareGuide.pdf. This book is a guide to developing software with ITK; it is the first of two companion books. This book covers building and installation, general architecture and design, as well as the process of contributing in the ITK community. The second book covers detailed design and functionality for reading and writing images, filtering, registration, segmentation, and performing statistical analysis.

CONTRIBUTORS

The Insight Toolkit (ITK) has been created by the efforts of many talented individuals and prestigious organizations. It is also due in great part to the vision of the program established by Dr. Terry Yoo and Dr. Michael Ackerman at the National Library of Medicine.

This book lists a few of these contributors in the following paragraphs. Not all developers of ITK are credited here, so please visit the Web pages at http://itk.org/ITK/project/parti.html for the names of additional contributors, as well as checking the GIT source logs for code contributions.

The following is a brief description of the contributors to this software guide and their contributions.

Luis Ibáñez is principal author of this text. He assisted in the design and layout of the text, implemented the bulk of the LaTeX and CMake build process, and was responsible for the bulk of the content. He also developed most of the example code found in the `Insight/Examples` directory.

Will Schroeder helped design and establish the organization of this text and the `Insight/Examples` directory. He is principal content editor, and has authored several chapters.

Lydia Ng authored the description for the registration framework and its components, the section on the multiresolution framework, and the section on deformable registration methods. She also edited the section on the resampling image filter and the sections on various level set segmentation algorithms.

Joshua Cates authored the iterators chapter and the text and examples describing watershed segmentation. He also co-authored the level-set segmentation material.

Jisung Kim authored the chapter on the statistics framework.

Julien Jomier contributed the chapter on spatial objects and examples on model-based registration using spatial objects.

Karthik Krishnan reconfigured the process for automatically generating images from all the examples. Added a large number of new examples and updated the Filtering and Segmentation chapters for the second edition.

Stephen Aylward contributed material describing spatial objects and their application.

Tessa Sundaram contributed the section on deformable registration using the finite element method.

YinPeng Jin contributed the examples on hybrid segmentation methods.

Celina Imielinska authored the section describing the principles of hybrid segmentation methods.

Mark Foskey contributed the examples on the AutomaticTopologyMeshSource class.

Mathieu Malaterre contributed the entire section on the description and use of DICOM readers and writers based on the GDCM library. He also contributed an example on the use of the VTKImageIO class.

Gavin Baker contributed the section on how to write composite filters. Also known as minipipeline filters.

Since the software guide is generated in part from the ITK source code itself, many ITK developers have been involved in updating and extending the ITK documentation. These include **David Doria**, **Bradley Lowekamp**, **Mark Foskey**, **Gaëtan Lehmann**, **Andreas Schuh**, **Tom Vercauteren**, **Cory Quammen**, **Daniel Blezek**, **Paul Hughett**, **Matthew McCormick**, **Josh Cates**, **Arnaud Gelas**, **Jim Miller**, **Brad King**, **Gabe Hart**, **Hans Johnson**.

Hans Johnson, **Kent Williams**, **Constantine Zakkaroff**, **Xiaoxiao Liu**, **Ali Ghayoor**, and **Matthew McCormick** updated the documentation for the initial ITK Version 4 release.

Luis Ibáñez and **Sébastien Barré** designed the original Book 1 cover. **Matthew McCormick** and **Brad King** updated the code to produce the Book 1 cover for ITK 4 and VTK 6. **Xiaoxiao Liu**, **Bill Lorensen**, **Luis Ibáñez**,and **Matthew McCormick** created the 3D printed anatomical objects that were photographed by **Sébastien Barré** for the Book 2 cover. **Steve Jordan** designed the layout of the covers.

Lisa Avila, **Hans Johnson**, **Matthew McCormick**, **Sandy McKenzie**, **Christopher Mullins**, **Katie Osterdahl**, and **Michka Popoff** prepared the book for the 4.7 print release.

CONTENTS

LIST OF FIGURES