

The ITK Software Guide

Design and Functionality

ITK 4.7

ITK Software Guide: Design and Functionality



4.7



Hans J. Johnson,

Matt McCormick, Luis Ibáñez

and the Insight Software Consortium



©2015 Kitware, Inc. (*cover, preface, postface*)

©2015 Insight Software Consortium (*main text body*)

Published by Kitware, Inc. <http://www.kitware.com>

An electronic version of this document is available from <http://itk.org>.

This work is licensed under a Creative Commons Attribution 3.0 Unported License.



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).

All product names mentioned herein are the trademarks of their respective owners.

Document created with L^AT_EX, 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 9781-930934-28-3

The ITK Software Guide

Book 2: Design and Functionality

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 COVER



The purpose of computing is Insight, not numbers.

Richard Hamming

ABOUT THE COVER

The cover image consists of a photograph of ABS plastic anatomical objects printed with a MakerBot Replicator 2X 3D printer. Mesh STL files were generated from the images with VTK.

Skull. Given that the origins of ITK are with the Visible Human Project, it is appropriate that the skull was derived from the Visible Woman dataset. The skull was segmented with ITK from the Visible Woman head CT images with simple thresholding¹.

Brain. The brain model was segmented with ITK as described in the open science publication:

McCormick M, Liu X, Jomier J, Marion C and Ibanez L. ITK: enabling reproducible research and open science. *Front. Neuroinform.* 8:13. 2014. doi: 10.3389/fninf.2014.00013

¹<https://github.com/XiaoxiaoLiu/3D-printing>

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 second of two companion books. This book covers detailed design and functionality for reading and writing images, filtering, registration, segmentation, and performing statistical analysis. The first book covers building and installation, general architecture and design, as well as the process of contributing in the ITK community.

ABSTRACT

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 $\text{\LaTeX}$ 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 Huhett**, **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

1	Reading and Writing Images	1
1.1	Basic Example	1
1.2	Pluggable Factories	4
1.3	Using ImageIO Classes Explicitly	5
1.4	Reading and Writing RGB Images	6
1.5	Reading, Casting and Writing Images	7
1.6	Extracting Regions	9
1.7	Extracting Slices	11
1.8	Reading and Writing Vector Images	13
1.8.1	The Minimal Example	13
1.8.2	Producing and Writing Covariant Images	14
1.8.3	Reading Covariant Images	16
1.9	Reading and Writing Complex Images	18
1.10	Extracting Components from Vector Images	19
1.11	Reading and Writing Image Series	21
1.11.1	Reading Image Series	21
1.11.2	Writing Image Series	23
1.11.3	Reading and Writing Series of RGB Images	25
1.12	Reading and Writing DICOM Images	27
1.12.1	Foreword	27
1.12.2	Reading and Writing a 2D Image	28

1.12.3	Reading a 2D DICOM Series and Writing a Volume	31
1.12.4	Reading a 2D DICOM Series and Writing a 2D DICOM Series	34
1.12.5	Printing DICOM Tags From One Slice	37
1.12.6	Printing DICOM Tags From a Series	40
1.12.7	Changing a DICOM Header	43
2	Filtering	47
2.1	Thresholding	47
2.1.1	Binary Thresholding	47
2.1.2	General Thresholding	50
2.2	Edge Detection	52
2.2.1	Canny Edge Detection	52
2.3	Casting and Intensity Mapping	53
2.3.1	Linear Mappings	53
2.3.2	Non Linear Mappings	55
2.4	Gradients	58
2.4.1	Gradient Magnitude	58
2.4.2	Gradient Magnitude With Smoothing	60
2.4.3	Derivative Without Smoothing	61
2.5	Second Order Derivatives	63
2.5.1	Second Order Recursive Gaussian	63
2.5.2	Laplacian Filters	67
	Laplacian Filter Recursive Gaussian	67
2.6	Neighborhood Filters	71
2.6.1	Mean Filter	72
2.6.2	Median Filter	73
2.6.3	Mathematical Morphology	75
	Binary Filters	75
	Grayscale Filters	77
2.6.4	Voting Filters	79
	Binary Median Filter	80
	Hole Filling Filter	81
	Iterative Hole Filling Filter	84

2.7	Smoothing Filters	87
2.7.1	Blurring	87
	Discrete Gaussian	87
	Binomial Blurring	90
	Recursive Gaussian IIR	91
2.7.2	Local Blurring	93
	Gaussian Blur Image Function	94
2.7.3	Edge Preserving Smoothing	94
	Introduction to Anisotropic Diffusion	94
	Gradient Anisotropic Diffusion	96
	Curvature Anisotropic Diffusion	98
	Curvature Flow	100
	MinMaxCurvature Flow	101
	Bilateral Filter	105
2.7.4	Edge Preserving Smoothing in Vector Images	107
	Vector Gradient Anisotropic Diffusion	108
	Vector Curvature Anisotropic Diffusion	109
2.7.5	Edge Preserving Smoothing in Color Images	110
	Gradient Anisotropic Diffusion	110
	Curvature Anisotropic Diffusion	112
2.8	Distance Map	115
2.9	Geometric Transformations	118
2.9.1	Filters You Should be Afraid to Use	118
2.9.2	Change Information Image Filter	118
2.9.3	Flip Image Filter	119
2.9.4	Resample Image Filter	120
	Introduction	120
	Importance of Spacing and Origin	125
	A Complete Example	131
	Rotating an Image	134
	Rotating and Scaling an Image	136
	Resampling using a deformation field	137
	Subsampling and image in the same space	139

Resampling an Anisotropic image to make it Isotropic	142
2.10 Frequency Domain	146
2.10.1 Computing a Fast Fourier Transform (FFT)	146
2.10.2 Filtering on the Frequency Domain	149
2.11 Extracting Surfaces	151
2.11.1 Surface extraction	151
3 Registration	155
3.1 Registration Framework	155
3.2 "Hello World" Registration	157
3.3 Features of the Registration Framework	166
3.4 Monitoring Registration	169
3.5 Multi-Modality Registration	173
3.5.1 Mattes Mutual Information	173
3.6 Centered Transforms	180
3.6.1 Rigid Registration in 2D	180
3.6.2 Initializing with Image Moments	188
3.6.3 Similarity Transform in 2D	193
3.6.4 Rigid Transform in 3D	195
3.6.5 Centered Affine Transform	202
3.7 Multi-Resolution Registration	204
3.7.1 Fundamentals	206
3.8 Multi-Stage Registration	211
3.8.1 Fundamentals	212
3.8.2 Cascaded Multistage Registration	221
3.9 Transforms	225
3.9.1 Geometrical Representation	225
3.9.2 Transform General Properties	228
3.9.3 Identity Transform	229
3.9.4 Translation Transform	229
3.9.5 Scale Transform	230
3.9.6 Scale Logarithmic Transform	231
3.9.7 Euler2DTransform	231

3.9.8	CenteredRigid2DTransform	233
3.9.9	Similarity2DTransform	234
3.9.10	QuaternionRigidTransform	235
3.9.11	VersorTransform	236
3.9.12	VersorRigid3DTransform	237
3.9.13	Euler3DTransform	238
3.9.14	Similarity3DTransform	239
3.9.15	Rigid3DPerspectiveTransform	241
3.9.16	AffineTransform	241
3.9.17	BSplineDeformableTransform	242
3.9.18	KernelTransforms	243
3.10	Interpolators	245
3.10.1	Nearest Neighbor Interpolation	246
3.10.2	Linear Interpolation	246
3.10.3	B-Spline Interpolation	246
3.10.4	Windowed Sinc Interpolation	247
3.11	Metrics	250
3.11.1	Mean Squares Metric	252
	Exploring a Metric	252
3.11.2	Normalized Correlation Metric	255
3.11.3	Mutual Information Metric	255
	Parzen Windowing	256
	Mattes et al. Implementation	256
3.11.4	Normalized Mutual Information Metric	257
3.11.5	Demons metric	257
3.11.6	ANTS neighborhood correlation metric	257
3.12	Optimizers	259
3.12.1	Registration using the One plus One Evolutionary Optimizer	261
3.12.2	Registration using masks constructed with Spatial objects	263
3.12.3	Rigid registrations incorporating prior knowledge	264
3.13	Deformable Registration	267
3.13.1	FEM-Based Image Registration	267
3.13.2	BSplines Image Registration	271

3.13.3	Level Set Motion for Deformable Registration	273
3.13.4	BSplines Multi-Grid Image Registration	276
3.13.5	BSplines Multi-Grid Image Registration in 3D	279
3.13.6	Image Warping with Kernel Splines	281
3.13.7	Image Warping with BSplines	281
3.14	Demons Deformable Registration	285
3.14.1	Asymmetrical Demons Deformable Registration	286
3.14.2	Symmetrical Demons Deformable Registration	289
3.15	Visualizing Deformation fields	292
3.15.1	Visualizing 2D deformation fields	292
3.15.2	Visualizing 3D deformation fields	293
3.16	Model Based Registration	299
3.17	Point Set Registration	308
3.17.1	Point Set Registration in 2D	308
3.17.2	Point Set Registration in 3D	312
3.17.3	Point Set to Distance Map Metric	314
3.18	Registration Troubleshooting	315
3.18.1	Too many samples outside moving image buffer	315
3.18.2	General heuristics for parameter fine-tuning	316
4	Segmentation	319
4.1	Region Growing	319
4.1.1	Connected Threshold	319
4.1.2	Otsu Segmentation	322
4.1.3	Neighborhood Connected	325
4.1.4	Confidence Connected	328
	Application of the Confidence Connected filter on the Brain Web Data	331
4.1.5	Isolated Connected	333
4.1.6	Confidence Connected in Vector Images	334
4.2	Segmentation Based on Watersheds	337
4.2.1	Overview	337
4.2.2	Using the ITK Watershed Filter	340
4.3	Level Set Segmentation	343

4.3.1	Fast Marching Segmentation	345
4.3.2	Shape Detection Segmentation	350
4.3.3	Geodesic Active Contours Segmentation	358
4.3.4	Threshold Level Set Segmentation	362
4.3.5	Canny-Edge Level Set Segmentation	367
4.3.6	Laplacian Level Set Segmentation	370
4.3.7	Geodesic Active Contours Segmentation With Shape Guidance	373
4.4	Feature Extraction	383
4.4.1	Hough Transform	383
	Line Extraction	383
	Circle Extraction	387
5	Statistics	391
5.1	Data Containers	391
5.1.1	Sample Interface	392
5.1.2	Sample Adaptors	394
	ImageToListSampleAdaptor	394
	PointSetToListSampleAdaptor	396
5.1.3	Histogram	399
5.1.4	Subsample	401
5.1.5	MembershipSample	404
5.1.6	MembershipSampleGenerator	406
5.1.7	K-d Tree	408
5.2	Algorithms and Functions	413
5.2.1	Sample Statistics	413
	Mean and Covariance	414
	Weighted Mean and Covariance	416
5.2.2	Sample Generation	419
	SampleToHistogramFilter	419
	NeighborhoodSampler	420
5.2.3	Sample Sorting	422
5.2.4	Probability Density Functions	424
	Gaussian Distribution	425

5.2.5	Distance Metric	426
	Euclidean Distance	426
5.2.6	Decision Rules	427
	Maximum Decision Rule	427
	Minimum Decision Rule	428
	Maximum Ratio Decision Rule	429
5.2.7	Random Variable Generation	430
	Normal (Gaussian) Distribution	430
5.3	Statistics applied to Images	431
5.3.1	Image Histograms	431
	Scalar Image Histogram with Adaptor	431
	Scalar Image Histogram with Generator	433
	Color Image Histogram with Generator	435
	Color Image Histogram Writing	438
5.3.2	Image Information Theory	440
	Computing Image Entropy	441
	Computing Images Mutual Information	444
5.4	Classification	449
5.4.1	k-d Tree Based k-Means Clustering	450
5.4.2	K-Means Classification	456
5.4.3	Bayesian Plug-In Classifier	458
5.4.4	Expectation Maximization Mixture Model Estimation	464
5.4.5	Classification using Markov Random Field	468