

Plastimatch – An Open Source Software Suite for Radiotherapy Image Processing

Gregory C. Sharp, Rui Li, John Wolfgang, George TY Chen, Marta Peroni, Maria Francesca Spadea, Shinichro Mori, Junan Zhang, James Shackelford, Nagarajan Kandasamy

- (1) Massachusetts General Hospital, Radiation Oncology
- (2) Politecnico di Milano, Biomedical Engineering
- (3) Magna Graecia University, Experimental and Clinical Medicine
- (4) National Institute of Radiological Sciences, Research Center for Charged Particle Therapy
- (5) Oregon Health and Sciences University, Radiation Oncology
- (6) Drexel University, Electrical and Computer Engineering

Abstract

Free and open source software is playing an increasingly important role throughout society. Free software provides a common economic good by reducing duplicated effort, and advances science by promoting the open exchange of ideas. This paper introduces the plastimatch open software suite, which implements a variety of useful tools for radiotherapy research and clinical practice. These tools include cone-beam CT reconstruction, rigid and deformable image registration, digitally reconstructed radiographs, and DICOM-RT file exchange.

Keywords

Open source software, medical imaging, software engineering, radiotherapy.

Introduction

Radiotherapy is a highly technical and rapidly changing field, where increasingly sophisticated software is used to support clinical goals. Commercial software is generally high quality and is used by most clinics for routine treatment planning and delivery. However, the reliance on commercial software leaves several gaps in our ability to deliver cutting-edge treatments. When commercial software has bugs, or is missing features, the clinic is required to implement complicated workarounds. Commercial software also often lacks the flexibility to communicate with complementary software from other vendors, including in-house solutions. Furthermore, it is difficult to do research with commercial medical software. Not only are vendors reluctant to provide open interfaces, purchase and support costs are generally too high for research use. For these reasons, we expect the role of open source software to grow in the radiotherapy clinic.

In this paper, we introduce the plastimatch software suite for radiotherapy image processing [1]. Plastimatch is open-source software, distributed under a BSD-style license. The focus of plastimatch is on high-performance algorithms for deformable image registration, cone-beam computed tomography (CBCT) reconstruction, and generation of digitally reconstructed radiographs (DRRs). Using standard interchange

formats such as DICOM and DICOM-RT, plastimatch can be easily used together with other open source tools, including CERR [2], Conquest DICOM [3], ImageJ [4], and 3D Slicer [5].

Background and History

The origin of the plastimatch project was a collection of image processing software written to support medical physics research. The first utility, DRR generation, was written in 2003. Shortly thereafter, deformable image registration was developed using the Insight Segmentation and Registration Toolkit (ITK) [6], and cone-beam CT reconstruction using the Feldkamp, Davis and Kress (FDK) algorithm [7].

In an attempt to overcome long computation times of ITK's deformable registration algorithms, we began exploring GPU-accelerated algorithms in 2006. The first algorithms targeted were FDK filtered backprojection and demons deformable registration [8]. At the time, CUDA was not yet mature, and therefore these algorithms were deployed using the BrookGPU compiler and runtime library [9]. Working together with our institutional lawyers, we released the first open source version of the GPU Imaging Toolkit (GUIT) in early 2007. Later in 2007, the GUIT registration routines were consolidated with the ITK registration routines to create the plastimatch project.

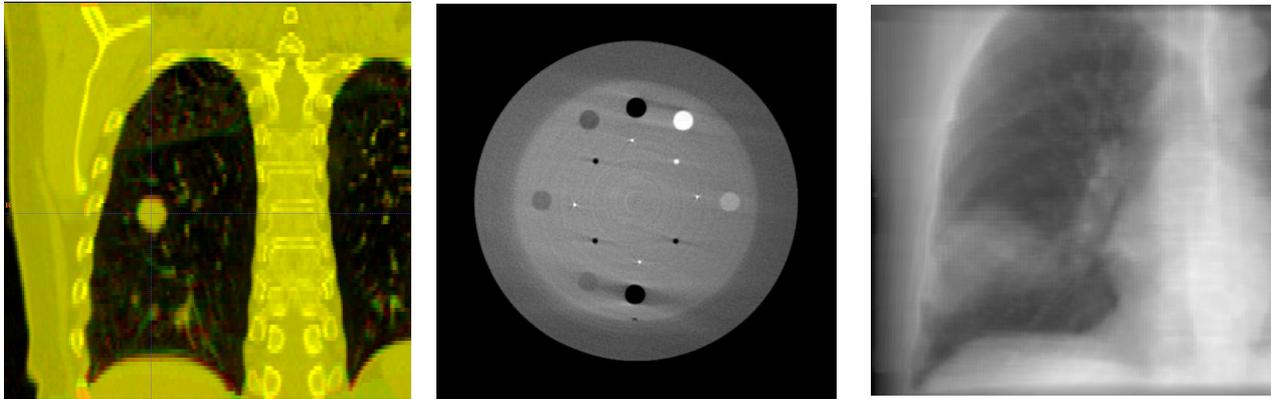


Figure 1: Plastimatch includes utilities for deformable image registration (left), filtered backprojection (center), and DRR generation (right).

In 2008, a native implementation of B-spline registration was designed to replace the ITK implementation, and the FDK reconstruction code was ported from BrookGPU to CUDA. In 2009, we have added GPU-accelerated B-spline registration, landmark-based interactive registration, and DICOM-RT. Our goals for 2010 include support for adaptive radiotherapy, and atlas-based image segmentation.

Overview of Plastimatch

Automatic Image Registration

Plastimatch uses a multi-stage, multi-algorithm framework for automatic image registration. Only pairwise registration is supported. In the initialization stage, the images are loaded, together with any image masks or initial guess. Then, the framework runs a fixed sequence of registration stages, as directed by a parameter file. A registration stage specifies the image resolution (for multi-resolution registration), the transform and metric to be optimized, and the optimization algorithm and parameters. If desired output files can be specified at each stage for saving intermediate results. A typical sequence of stages might include a single rigid alignment stage, followed by two to four deformable registration stages with increasing resolution and decreasing grid spacings.

Table 1 lists the summary of algorithms included in plastimatch, which includes six different core registration methods. Depending on the registration method, you can choose one of four implementations: ITK, single core (SC), multi-core (MC), or GPU. The six registration algorithms can operate on eight different transform types: six ITK transforms and two native transforms. At the end of each stage, the optimal transform propagated to the next stage, and automatically converted to a new transform type by the plastimatch application framework. An example of

| Algorithm | ITK | SC | MC | GPU |
|-----------------------------|-----|----|----|-----|
| Translation registration | X | | | |
| Rigid registration | X | | | |
| Affine registration | X | | | |
| Demons registration | X | X | | X |
| B-spline registration (MSE) | X | X | X | X |
| B-spline registration (MI) | X | X | | |
| Landmark spline (global) | X | | | |
| Landmark spline (local) | | X | | |
| FDK | | | X | X |
| DRR | | | X | X |

Table 1: Algorithm summary. Implementations use either ITK, single core (SC), multi-core (MC), or GPU.

* MSE=mean squared error, MI=mutual information

deformable registration results using the native B-spline method are shown in Figure 1 (left).

Cone-beam CT Reconstruction

A cone-beam CT reconstruction application is provided which implements filtered back projection using the Feldkamp, Davis and Kress (FDK) algorithm. Input images in either raw, pfm, or hnd format are read, filtered, and backprojected into a user defined volume geometry. Images in raw or pfm format must be accompanied by a geometry specification file, whereas files in the Varian hnd format use the geometry specified by the file header. Ramp filtering is performed on the CPU using the FFTW library [10], while backprojection is performed on either CPU or GPU. An example reconstruction is shown in Figure 1 (center).

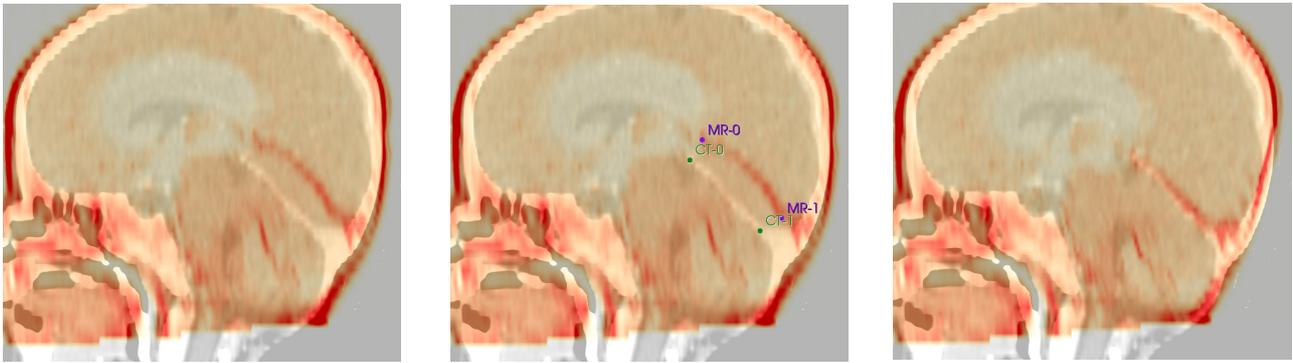


Figure 2: Automatic registration using B-splines was used to warp the MRI of a 6-month old infant onto the CT of the same patient at age 2. The registration properly matches the skull, but features within the brain are not properly aligned (left). The registration results are partially corrected by placing two landmarks (center), which improve the registration (right).

DRR Generation

The plastimatch DRR generator implements three variants of the Siddon ray tracing method [11]. The fastest and most popular method uses the original “exact pathlength” method based on the intersection of rays with the image voxels. In addition, two voxel interpolation methods are included, which can be used to increase the apparent resolution of the DRR construction. Both multi-core and GPU versions are available, and an example of the DRR application output is shown in Figure 1 (right).

Interactive (Landmark-based) Image Registration

While automatic registration yields acceptable results in many cases, we are often confronted with difficult registration problems where automatic registration fails. For this purpose, plastimatch includes two manual registration tools: a “global” landmark-based tool based on thin plate splines, and a “local” landmark-based tool based on radial basis functions.

The global tool is implemented as an ITK wrapper, takes a list of corresponding points in 3D, and generates a complete vector that interpolates all of the input landmarks. This method requires a minimum of six landmarks, which are used to find a global affine transform superimposed with a minimum energy deformation field [12]. The global landmark registration results can be used as a standalone method, or to initialize the automatic registration.

In contrast, the local tool is a native radial basis function warper, and does not perform global rigid or affine mapping. Instead, it uses a small number of landmark pairs to correct failed deformable registration results. An example of this idea is shown in Figure 2, where the failed registration (left), is corrected using two pairs of landmarks (center, right).

Data Interchange

Plastimatch supports a wide variety of file input types for data interchange. Using ITK wrappers, most image formats are supported, including DICOM, Analyze, Metimage, and NRRD. In addition, partial support exists for DICOM-RT, XiO, and RTOG formats. Plastimatch is capable of rasterizing DICOM-RT structure sets into images, as well as converting images back into DICOM-RT structure sets. In addition, a utility is provided for attaching existing DICOM-RT structure sets onto arbitrary DICOM series.

User Interface

The standard interface to plastimatch is a set of command line utilities, which are configured using configuration files and command line options. However, graphical user interfaces for plastimatch do exist using plugin methods. Currently Plastimatch runs as a plugin for Aqualizer [13] and 3D Slicer [5]. Aqualizer is a specialized research software for four-dimensional treatment planning. Deformable image registration is used to map radiation dose from all breathing phases onto a reference phase, and accumulate the time-averaged dose. 3D Slicer is a general purpose research software for medical image computing. Plastimatch plugins are available for automatic registration, landmark-based registration, and DICOM-RT import.

Software Engineering

As of January 2010, plastimatch comprises 98,000 lines of code. The code is cross-platform, and makes heavy use of external libraries, including ITK, dcmtk, fftw, CUDA, sqlite. Therefore, we manage our project complexity and growth through a comprehensive set of software engineering tools.

Configuration Management

Software configuration management is managed using CMake. The CMake system uses a custom build description language to generate platform-specific makefiles for the project. At configuration time, the software tests for the presence of required and optional libraries and compiler features, and configures plastimatch accordingly. The overall philosophy of our project is that external libraries are optional, and the absence of the library results in reduced functionality. As an example, if CUDA is not installed on the target system, plastimatch will still build and run, but cannot use CUDA-accelerated implementations.

Version Control and Defect Tracking

Software version control and defect tracking are managed through the use of a web-enabled, custom install of GForge Advanced Server. Dissemination of software and documentation are through a subversion repository, which allows anonymous access. Tarballs and windows binaries are only made sporadically upon request. Defect tracking is also accomplished using the web-based portal.

Automated Testing

As a measure to maintain and improve software quality, we have recently (2009) initiated a automated software testing using CTest. Test cases are included with the software distribution, and can be executed using the native build system (such as Makefiles or Visual Studio). As of January 2010, plastimatch contains 76 tests. Prior to committing changes, the developer executes a short battery of tests as a sanity check to ensure that basic functionality has not been broken. At the current time, the full sequence of tests is run manually after major updates. Our goal in 2010 is to implement to an overnight testing strategy.

Licensing

The plastimatch software is licensed under a custom BSD-style license. The license specifically allows royalty-free non-exclusive license to use, modify, and redistribute our software. The primary restrictions on licensing are that (1) attribution and copyright notices be retained, (2) modified versions must be clearly marked, and (3) names, logos, and trademarks of our institutions are not used for promotion. Our software is provided "AS IS," without warranty. In addition, our license clearly states that the software has been designed for research purposes only, and that clinical applications are neither recommended nor advised. A complete copy of the license is available on-line [1].

Concluding Remarks

This paper has introduced the plastimatch project, including its history, scope, engineering infrastructure, and license. We would like to thank Steve Jiang for help getting the project started, Denise Lagasse for help with licensing, and Steve Pieper for help with 3D Slicer integration. Funding for this project has been provided by NIH/ NCI 6-PO1 CA 21239, the Federal share of program income earned by MGH on C06CA059267 and the Progetto Rocca Foundation, a collaboration between MIT and Politecnico di Milano.

References

- [1] Plastimatch.
<http://plastimatch.org>
- [2] JO Deasy, AI Blanco, VH Clark. 2003. CERR: a computational environment for radiotherapy research. *Medical Physics* 30, no. 5: 979-985.
<http://radium.wustl.edu/CERR>
- [3] Conquest DICOM.
<http://www.xs4all.nl/~ingenium/dicom.html>
- [4] WS Rasband. ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA, 1997-2009.
<http://rsb.info.nih.gov/ij>
- [5] 3D Slicer.
<http://slicer.org>
- [6] The Insight Segmentation and Registration Toolkit.
<http://itk.org>
- [7] LA Feldkamp, LC Davis, JW Kress. 1984. Practical cone-beam algorithms. *J. Opt. Soc. Am.* 6: 612-619.
- [8] J-P Thirion. 1998. Image matching as a diffusion process: an analogy with Maxwell's demons. *Med Image Anal* 2, no. 3: 243-260.
- [9] I Buck, T Foley, D Horn, J Sugerman, K Fatahalian, M Houston, P Hanrahan. 2004. Brook for GPUs: stream computing on graphics hardware. *ACM Trans Graph* 23, no. 3: 777-786.
- [10] M Frigo, SG Johnson. 2005. The Design and Implementation of FFTW3. *Proceedings of the IEEE* 93, no. 2: 216-231.
- [11] RL Siddon. 1985. Fast calculation of the exact radiological path for a three-dimensional CT array. *Medical Physics* 12, no. 2: 252-255.
- [12] FL Bookstein. 1989. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans Pat Anal Mach Intell*, 11, no. 6: 567-585.
- [13] B Hoffman, K Martin. 2006. *Mastering Cmake 2.2 Edition*. Kitware, Inc.
<http://cmake.org>
- [14] S Mori, GTY Chen. 2008. Quantification and visualization of charged particle range variations. *Int J Rad Onc Bio Phys* 72, no. 1: 268-277.