
Importing Contours from DICOM-RT Structure Sets

Release 1.00

Jason Dowling^{1*}, Mathieu Malaterre², Peter B. Greer^{3,4} and Olivier Salvado¹

October 6, 2009

¹Australian e-Health Research Centre, CSIRO ICT Centre, Australia

²Mathieu Malaterre Consulting, France

³Calvary Mater Newcastle Hospital, Newcastle, Australia

⁴University of Newcastle, Australia

* Jason.Dowling@csiro.au

Abstract

This paper presents parsing software for importing contours from DICOM RT structure set files into binary image files. This is a common requirement for transferring expert manual segmentations from radiotherapy treatment planning software into external image analysis environments (such as ITK). We present an implementation of RT structure set importing and present some example results from prostate cancer radiotherapy planning volumes.

Contents

1	Introduction	2
2	DICOM-RT Structure Files	2
3	Software Requirements	3
4	Usage	3
5	Method	4
6	Results	5
7	Test Data	8
8	Future work	9
9	Acknowledgements	9

1 Introduction

Earlier this year Subrahmanyam Gorthi et al. published a paper [1] in the Insight Journal on the conversion of binary masks to DICOM RT Structure Set Storage files (also known as uid: 1.2.840.10008.5.1.4.1.1.481.3). This service-object pair (SOP) class is defined in the DICOM standard PS 3.3 – 2008 (http://medical.nema.org/medical/dicom/2008/08_03pu.pdf). This work presents a method to do the reverse: extract all structures from a DICOM RT structure file and then save these structures as individual binary images for further analysis or visualization.

This work was motivated by the need of our group to

- i) Develop average shape atlases and shape models from pre-existing expert segmentations which are routinely made in the course of radiotherapy treatment planning.
- ii) To enable quantitative comparisons between automatic and expert manual segmentations (eg. Dice similarity coefficients or Hausdorff distance).
- iii) Enable the visualization of the manual segmentation in open source tools (eg. ITK-Snap).

There are some other applications which attempt to provide some of this functionality: The best of these is the Computational Environment for Radiotherapy Research (CERR) (available for free download from <http://radium.wustl.edu/CERR/about.php>). However CERR requires Matlab and does not export contours to binary files. Osirix (<http://www.osirix-viewer.com/>) is an open source ITK/VTK based application which can load RT structure files, however it cannot export them to binary image files, and it only runs on Apple Macs. RT_Image from Stanford University (<http://rtimage.sourceforge.net/>) requires the proprietary Interactive Data Language (IDL).

2 DICOM-RT Structure Files

An excellent overview of the RT structure set modules and tags has been previously reported in Gorthi et al [1]. In similar code to the GDCM 2.0 [2] example program *gdcmscene.cxx*, we have used the following tags via the GDCM 2.0 library to extract data from a single structure file:

```

Tag (3006, 0020) - Identify the start of structure set ROI sequence.
Tag (3006, 0039) - To identify start of ROI contour sequence
We then identify the number of structures found.
For each structure
    Tag (3006, 0026) Identify user-defined name of ROI.
    Tag (3006, 0040) Extract the contour sequence.
    For each item in the current structure:
```

```

                Tag (3006, 0050) Extract contour data.
    Loop
Loop

```

3 Software Requirements

The following software needs to be installed (in this order):

- **CMake 2.4** or higher (<http://www.cmake.org>)
- **GDCM 2.0** or higher (<http://gdcm.sf.net>)

For the purposes of this application, all of the initial CMake options can be set to OFF. Using make then make install will help ITK automatically locate the gdcm directory (see next step).

- **Insight Toolkit 3.14** or higher (<http://www.itk.org/>)

When building ITK, **ITK_USE_SYSTEM_GDCM=ON** must be set. ITK should then automatically set the GDCM_DIR.

4 Usage

Currently the parser requires the DICOM image data to be initially converted into a volume readable by ITK (eg. NIfTi) this is required to obtain spacing, orientation, and orientation information. Future versions of this code will read the DICOM file directory and generate this volume from the image files.

The program is currently a simple command line utility which is run with three parameters:
itkDICOMRT inputFile inputTemplateFile structPrefix

for example:

```

itkDICOMRT/bin/itkDICOMRT /home/dow170/Dev/DICOM-
RT/rbwh_20090727/RS1.3.6.1.4.1.2452.6.3249085479.1251782939.1768926618.296234
.dcm /home/dow170/Dev/DICOM-RT/rbwh_20090727/G1CX3_CT.nii G1CX3_CT_
template image size: [512, 512, 131]
template image spacing: [0.993, 0.993, 3]
template origin: [-236.523, -267.969, 174]
calculated dcm origin: -219
Parsing: /home/dow170/Dev/DICOM-
RT/rbwh_20090727/RS1.3.6.1.4.1.2452.6.3249085479.1251782939.1768926618.296234
.dcm
media storage: 1.2.840.10008.5.1.4.1.1.481.3
Number of structures found:6
Structure 0. Number of regions: 17
0. Structure name: CTV
Inserting region with 74 points into slice: 75
Inserting region with 99 points into slice: 76

```

```

Inserting region with 85 points into slice: 77
Inserting region with 117 points into slice: 78
Inserting region with 114 points into slice: 79
Inserting region with 120 points into slice: 80
Inserting region with 142 points into slice: 81
Inserting region with 116 points into slice: 82
Inserting region with 72 points into slice: 83
Inserting region with 26 points into slice: 84
Inserting region with 49 points into slice: 85
Inserting region with 55 points into slice: 86
Inserting region with 57 points into slice: 87
Inserting region with 49 points into slice: 88
Inserting region with 40 points into slice: 89
Inserting region with 55 points into slice: 90
Inserting region with 38 points into slice: 91
File written: G1CX3_CT_CTV.nii.gz
Structure 1. Number of regions: 23
1. Structure name: PTV
Inserting region with 117 points into slice: 72
...

```

Example output files:

```

G1CX3_CT_sc.nii.gz
G1CX3_CT_rtlung.nii.gz
G1CX3_CT_ltlung.nii.gz
G1CX3_CT_EXTERNAL.nii.gz
G1CX3_CT_PTV.nii.gz
G1CX3_CT_CTV.nii.gz

```

5 Method

The code accepts an input DICOM RT structure file (usually with filename RS*.dcm) and a template volume readable by ITK (eg. NIfTi). In addition an output prefix is required and is added to the start of the filename for each structure.

The output is a single binary volume for each structure in the input DICOM RT structure file.

As each structure is represented as a set of points for a single slice, these points are extracted in sequence from the RT structure file as in section 2; transformed from physical to image space; and stored in pointlists which then used to generate itk::PolygonSpatialObjects. These polygons are inserted into a temporary 2D slice using the itk::SpatialObjectToImageFilterType. Each slice is then inserted into a 3D volume and the next set of points considered.

6 Results

The code has been tested on around 30 MRI, 30 CT and 15 Cone Beam CT volumes, which represents around 500 separate output binary volumes. No attempt has been made to optimize performance. The length of time required to parse a structure depends on the number of points and image size: we found the bladder, prostate and rectum were converted in under a minute, however CT pelvic bone, depending on the number of regions within a slice, could take up to ten minutes to convert.

Figs. 2 to 4 give some examples of the results. An image from the Varian Eclipse (http://www.varian.com/us/oncology/radiation_oncology/eclipse/) treatment planning software at Calvary Mater Newcastle Hospital is shown in Fig 1. These contours were then exported to a RT structure file and parsed using this code. Fig. 2 includes all of the exported structures in the same axial slice as Fig. 1 (the binary masks have been converted to edges by using the `itk::BinaryDilateImageFilter` and `itk::SubtractImageFilter`). Four binary images (representing the prostate, bladder, rectum and bone) have been combined from the DICOM RT structure file for the same patient and rendered in Fig. 3. The resulting binary masks for the Primary Target Volume (PTV) is shown overlaid on the template image in Fig. 4.

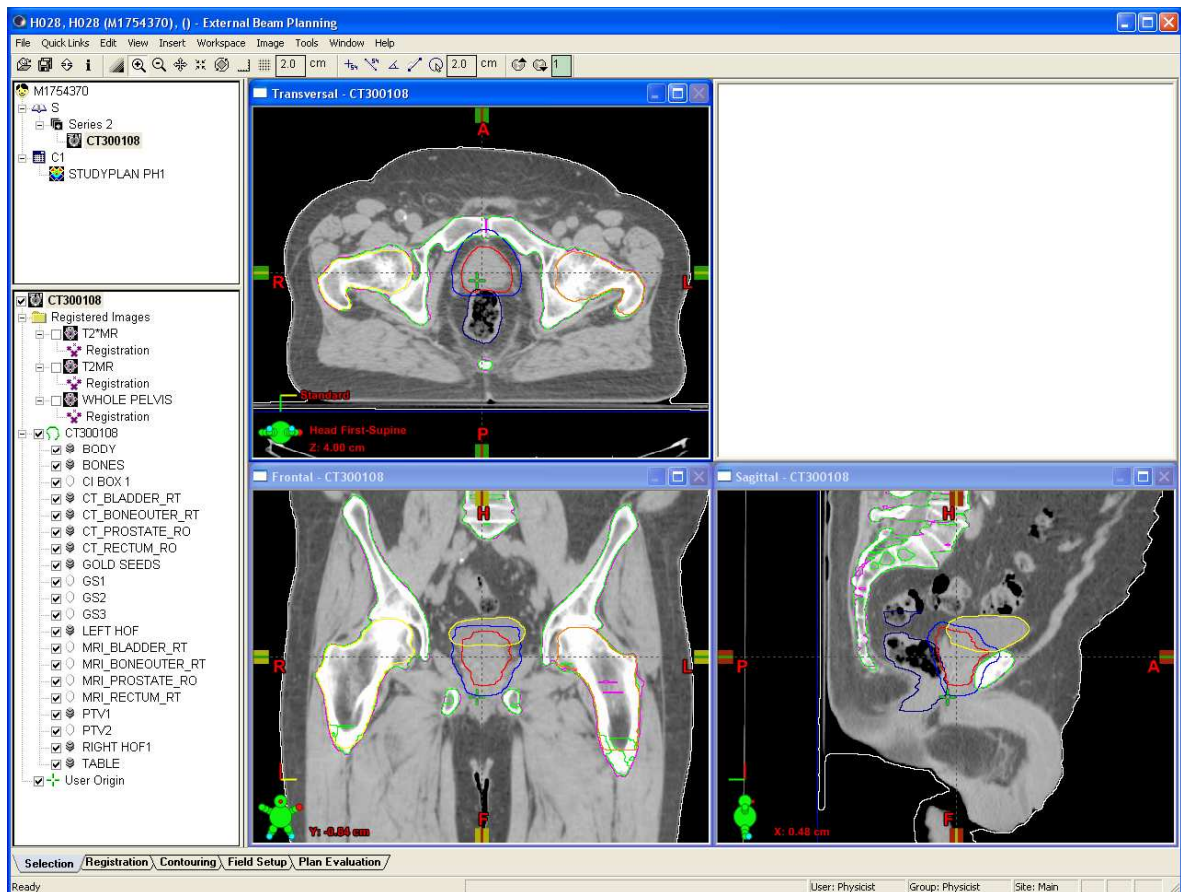


Fig 1. Screen capture of treatment planning software displaying manual contours for external beam radiotherapy treatment planning for a prostate cancer patient at Newcastle Mater Hospital.

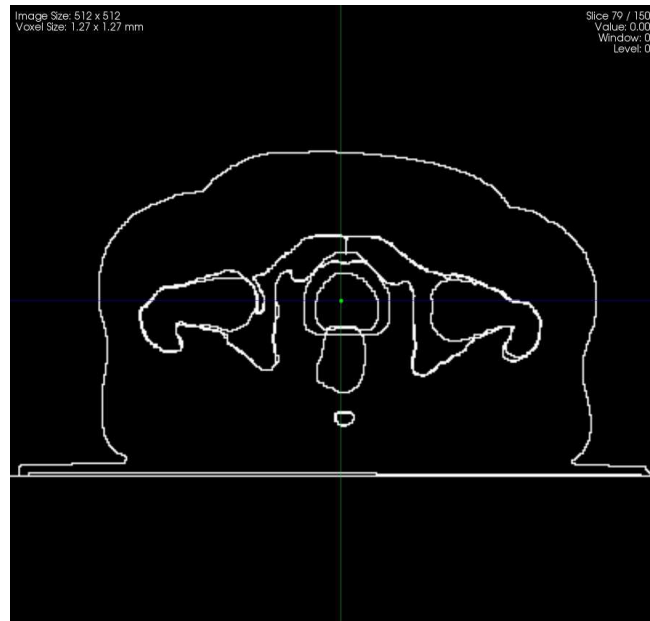


Fig 2. Parsed DICOM RT structures which have been exported from Eclipse (this is the same patient as shown in Fig. 1).

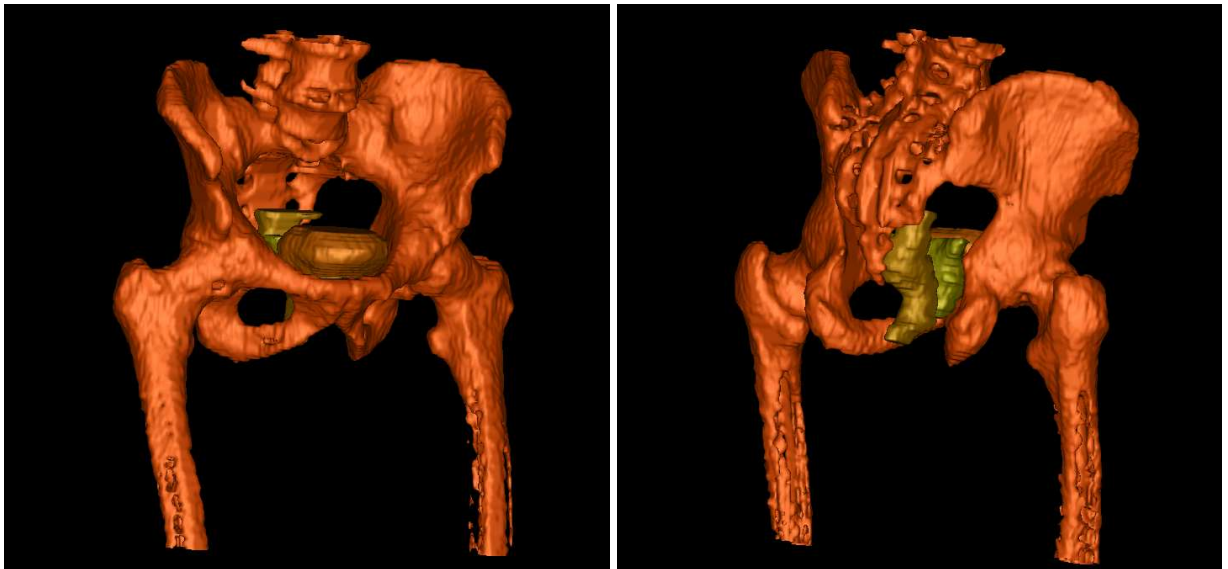


Fig 3. Front and rear rendered surfaces from the parsed manual contours for the bladder, rectum, prostate, and bone. ITK-Snap was used to generate these images (<http://www.itksnap.org>) [3].

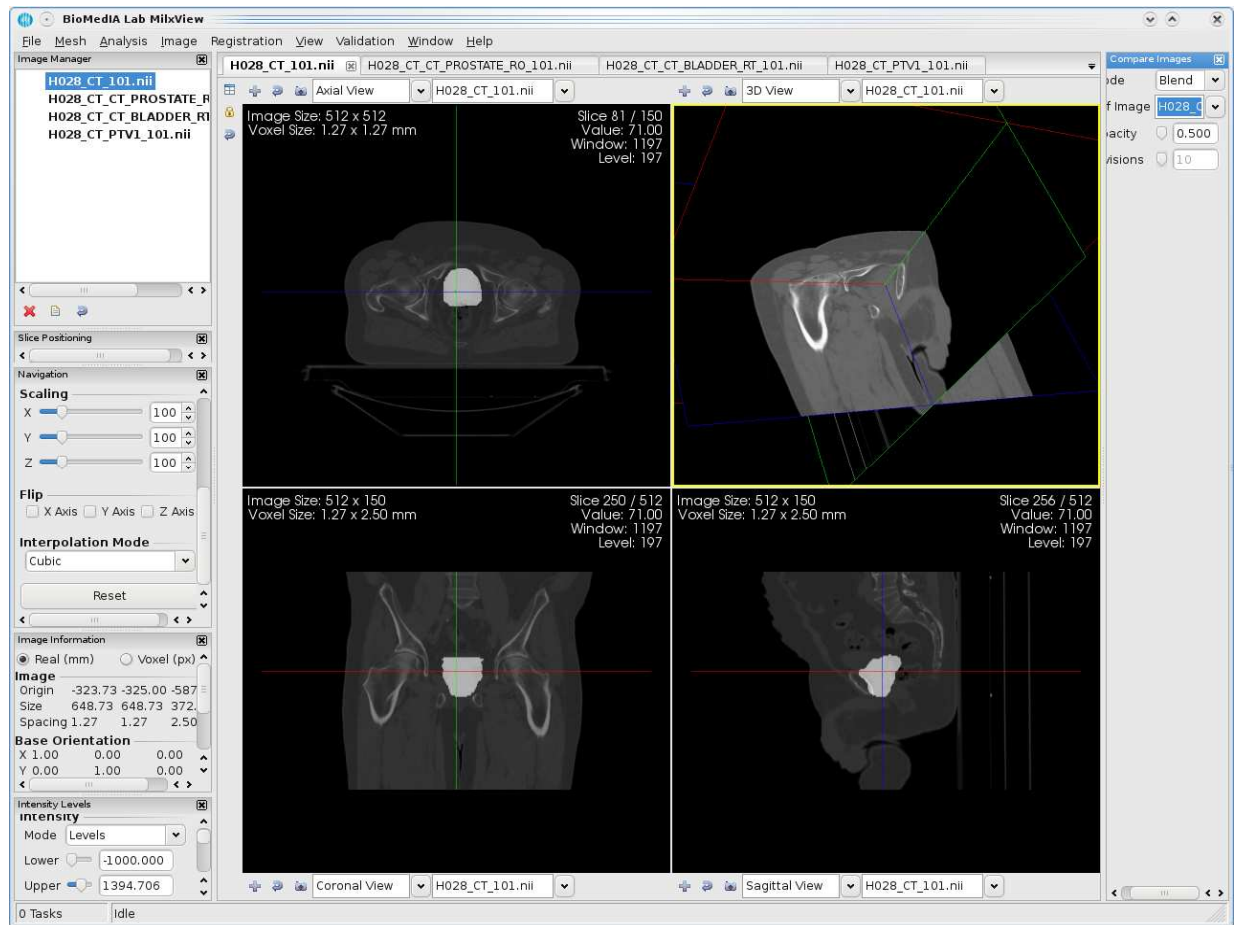


Fig 4. Parsed binary masks for the prostate (same patient as Fig. 1 – exported to a DICOM RT structure file and parsed. The binary mask for the PTV is shown in overlay over the template volume). The display software used is MILXView (http://aehtc.com/biomedical_imaging/milx.html).

7 Test Data

The following two files are provided for testing:

1. A sample DICOM RT structure file:
RS.1.2.246.352.71.4.886768594.5257.20090622110825.dcm
2. A template volume for this structure file: **b032.nii**

The DICOM RT structure file contains the following four structures:

1. MRI_BLADDER_RT

2. MRI_BONEOUTER_RT
3. MRI_PROSTATE_RO
4. MRI_RECTUM_RT

There are also three empty structures (this is quite common - these have either not been exported from the treatment planning software, or have been created and are empty).

8 Future work

This program has been tested on DICOM RT structure files exported from Varian Eclipse and Oncentra Masterplan (<http://www.nucletron.com/ProductsAndSolutions/Pages/Oncentra%C2%AEMasterPlan.aspx>). Further modifications may be required for structure files from other vendors.

The need for a template volume to obtain the origin, orientation, etc. should be replaced with by a DICOM image reader. In this case a user would only need to specify a directory for the structures to be extracted (as in CERR).

The performance of the parser could be improved. Karthik Krishnan has suggested that rasterizing the meshed output from TestvtkGDCMPolyDataReader using an vtkOBTree (as in the vtkPolyDataToImageStencil class) may be a faster method to generate the image output (an example is contained in ./Hybrid/Testing/Tcl/TestImageStencilWithPolydata.tcl). This is an area which we should investigate in the future.

9 Acknowledgements

The authors thank Jackie Patterson, Joel Parker, James Denham, Sanjiv Gupta, Colin Tang and Jonathon Lambert from the Calvary Mater Newcastle Hospital for assistance with manual contouring and data exporting. Thanks also to Karthik Krishnan (Kitware) and Nick Dowson (CSIRO) for testing and providing helpful comments on the code.

This work was partially funded by the Cancer Council NSW, Project Grant RG 07-06.

This work benefited from the use of Grassroots DICOM (GDCM), an open source software developed as an initiative of the CREATIS laboratory and available at <http://gdcml.sourceforge.net> . The multi-platform configuration tool CMake was used for configuring GDCM, ITK and facilitating its use from our project. CMake was partially funded by the U.S. National Library of Medicine as part of the Insight Toolkit project. CMake is an open source system and it is freely available at www.cmake.org .

References

- [1] S. Gorthi, M. Bach Cuadra, and J.-P. Thiran, Exporting Contours to DICOM-RT Structure Sets, *Insight Journal*, 2009. available online at <http://www.insight-journal.org/browse/publication/316>

[2] Mathieu Malaterre et al. *GDCM Reference Manual*, 2008 available online at <http://gdcm.sourceforge.net/gdcm.pdf>

[3] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *Neuroimage* 2006 Jul 1;31(3):1116-28.