

ADMIRE

Generated by Doxygen 1.9.2

1 User Documentation	2
2 Installation Details	2
2.1 System Requirements	2
2.2 GPU Requirements	2
2.3 Prerequisites	2
2.4 Program Files	2
2.5 Data Files	3
2.6 Uninstalling	3
2.7 Advanced	3
2.7.1 Current Working Directory	3
2.7.2 Unattended Installation	4
2.7.3 Non-Administrative Installs	4
2.7.4 License File	4
3 Changelog	4
3.1 Changes for each version	4
3.1.1 Most Recent Versions	5
3.1.2 Older Versions	5
3.1.3 Really Old Versions	17
4 GPU Frequently Asked Questions	26
4.1 What GPU Should I Buy?	26
4.2 Hardware	27
4.3 Installation	28
4.4 Remote Access	30
4.5 Performance	30
4.6 Deployment	32
4.7 General GPU Computing	32
5 Image Spacing and Orientation	33
5.1 Overview	33
5.2 Image Spacing	33
5.2.1 Outlier images	33
5.2.2 Contour Generation	33
5.3 "Slightly" Oriented Images	34
5.3.1 The Problem	34
5.3.2 A Solution: A Tale of Two Tolerances	35
5.4 Troubleshooting	36
6 Segmenter Detail	37

6.1 Inter-Patient Deformable Registration	37
6.2 Intra-Patient / Adaptive Deformable Registration	38
6.3 Deep Learning Autosegmentation	38
6.4 Glossary	38
7 Structure Mappings	39
7.1 Motivation	39
7.2 Mapping Process	39
7.3 Important Mappings	39
7.3.1 All case types with CT patient (except Demons)	39
7.3.2 Head/Neck cases	40
7.3.3 Prostate cases	40
7.3.4 Inpatient cases	40
7.3.5 Patch Fusion, RF Label Fusion	40
7.4 Mappings File Location	41
8 Deformation Vector Field File Guide	41
8.1 ITK Image and the MHD File Format	41
8.2 DICOM Format	43
8.3 What Is the ADMIRE Deformation Field?	44
8.4 Using the deformation field	45
8.5 Post-Processing	46
8.6 Dimensions	46
8.7 Inverse Fields	46
8.8 Summary	47
9 Deep Learning Autosegmentation	47
9.1 System Requirements	47
9.2 Overview	48
9.2.1 Limitations	48
9.2.2 GUI	49
9.2.3 Command-line	49
9.3 Current Models	50
9.3.1 Autosegmentation	50
9.3.2 Synthetic CT	50
9.4 Frequently Asked Questions	50
10 Dose Deformation	51
10.1 Typical Workflow	51
10.2 Implementation Details	52
10.3 DICOM	52

11 Artifacts, Naming Conventions	52
11.1 Overview	53
11.2 Filename Generation	53
11.3 Location	53
11.4 Renaming	54
11.5 Deformed Images	54
12 Tools	54
12.1 AtlasHammer	55
12.1.1 Overview	55
12.1.2 Basic usage	55
12.1.3 Advanced Usage	55
12.1.4 Warnings	55
12.2 CBCTtoSCT	55
12.2.1 Overview	55
12.2.2 Basic usage	56
12.2.3 Exit Codes	58
12.2.4 Backwards Compatibility	58
12.2.5 DICOM Notes	58
12.2.6 Limitations	59
12.2.7 Model Details	59
12.2.8 Future Goals	59
12.3 CompareContours	59
12.3.1 Overview	59
12.3.2 Basic usage	60
12.3.3 Interpreting Results	61
12.3.4 Recommendations	63
12.3.5 Implementation details	63
12.3.6 Reference	63
12.4 ConvertDCMtoITKVTk	64
12.4.1 Overview	64
12.4.2 Basic usage	64
12.4.3 Supported Formats	65
12.4.4 Filename string formatting	65
12.4.5 Examples	65
12.4.6 ROI Subset Filtering	66
12.4.7 Putting It All Together	68
12.4.8 Automation	69
12.4.9 Misc	69

12.5 DoseWarpUtil	70
12.5.1 Overview	70
12.5.2 Dose Warping Algorithm	70
12.5.3 Plan Duplication	73
12.5.4 Target Grid Considerations	73
12.5.5 Current limitations	75
12.5.6 GUI Notes	75
12.6 ImageSimilarityUtil	76
12.6.1 Overview	76
12.6.2 Basic usage	76
12.6.3 Simple Use Case	76
12.6.4 Interpreting Results	78
12.6.5 Additional Options	79
12.6.6 Return Codes	79
12.6.7 Misc notes	79
12.7 ImageWarpUtil	79
12.7.1 Overview	79
12.7.2 Basic usage	80
12.7.3 Frame of Reference	80
12.7.4 Notes and Limitations	81
12.8 MAFusionUtil	81
12.8.1 Overview	81
12.8.2 Basic usage	81
12.8.3 Simple Use Case	82
12.8.4 ROI Binning	82
12.8.5 Multilabel STAPLE	82
12.8.6 Misc Notes	83
12.9 REGtoITK	84
12.9.1 Overview	84
12.9.2 Example Usage	84
12.10 StandardizeMR	85
12.10.1 Overview	85
12.10.2 Basic usage	85
12.10.3 Implementation	86
12.10.4 Known Limitations	87
13 Command Line Usage	87
13.1 Differences from GUI	87
13.2 Setup	87

13.3 Basic Usage Examples	88
13.4 Advanced Usage	88
13.4.1 Patient, Atlas paths	88
13.4.2 Results	88
13.4.3 Structures	88
13.5 STAPLE	89
13.6 Patch-Based Fusion	89
13.7 Random Forest Label Fusion	89
13.8 Intrasubject and RCCT	89
13.8.1 one-to-all	90
13.8.2 cascade	90
13.8.3 one-to-many	90
13.8.4 Single Case	90
13.8.5 Adaptive Lung	90
13.8.6 Constraints	91
13.9 Modalities	91
13.10 Combining Arguments	91
13.11 Output	92
13.12 Exit Codes	92
13.13 Cygwin	92
13.14 Other ADMIRE Binaries	92
13.15 Parameter Reference	93
14 DICOM Notes	96
14.1 Cone-Beam CT	96
14.2 ROI Interpreted Types	96
14.3 Spatial Registration	96
14.4 ROI Contour Sequence - Contour Image Sequence	96
15 Differences from Product Version	97
15.1 Overview	97
15.2 Algorithms	97
15.3 Workflows	97
15.4 DICOM	97
15.5 Misc	97
16 Windows 7 Notes	98
16.1 Displaying the ProgramData folder	98
16.1.1 Permanently Show	98
16.1.2 Temporary	101

16.2 Plug-in Location	102
17 Whitepapers & References	102
17.1 Elekta	102
17.2 Non-Elekta	103
17.3 Comparison and Performance Studies	104
18 Licenses and Acknowledgments	104
18.1 Tensorflow	104
18.2 ADV	107
18.3 Botan	107
18.4 DCMTK	108
18.5 VTK	109
18.6 ITK	109
18.7 Elastix	110
18.8 ITK	110
18.9 Boost	111
18.10 ZLib	111
18.11 libpng	112
18.12 Thrust	113
18.13 Segmentation Evaluation	114
19 Data Structure Index	114
19.1 Data Structures	114
20 File Index	114
20.1 File List	114
21 Data Structure Documentation	114
21.1 CmsAlgorithm::SurfaceComparisonStatistics Class Reference	114
21.1.1 Detailed Description	118
21.1.2 Member Function Documentation	118
21.1.3 Field Documentation	122
22 File Documentation	124
22.1 C:/wa/admire/BusinessLayer/SegmentationCore/CmsAlgorithm/SurfaceComparisonStatistics.h File Reference	124
22.1.1 Detailed Description	125
22.2 SurfaceComparisonStatistics.h	125
Index	129

1 User Documentation



This is the documentation for the research version of ADMIRE prototype. It is mostly a collection of technical notes and answers to frequently asked questions. It is not a comprehensive user's guide; this is a prototype, after all. The target audience is our research collaborators who are familiar with medical image segmentation and registration.

2 Installation Details

Lists installation location, files, etc.

2.1 System Requirements

- 64-bit CPU
- Windows 7 64-bit or higher
- .NET 4.5 runtime
- 4 GB RAM (8 GB preferred)
- multiple cores or CPUs preferred

2.2 GPU Requirements

While a GPU is not strictly required (almost all functionality has CPU implementations), refer to the [GPU Frequently Asked Questions](#) for detailed information regarding GPU selection.

2.3 Prerequisites

- If the .NET 4.5 runtime is not found, a message will be displayed asking the user or administrator to install it. This is usually not a problem if installing onto Windows 7.

2.4 Program Files

By default, ADMIRE installs its executables, DLLs, and model files into the 64-bit program files location (%ProgramFiles%). On Windows 7, this typically results in C:\Program Files\ADMIRE. This can be changed by the user during installation.

2.5 Data Files

By default, ADMIRE stores logs, configuration, etc into ADMIRE within the Windows Common Application Data location (%ALLUSERSPROFILE%), which is commonly C:\Users\All Users (an alias for C:\ProgramData) on Windows 7. Demo atlases and patient data are also installed into this location (if selected during installation), but user-provided atlases and patient data may be located anywhere. This location can be changed by the user during installation, but is usually unnecessary.

2.6 Uninstalling

Remove ADMIRE via the standard *Add/Remove Programs* section of the Control Panel. Once this process completes, the following files may be left behind because they are usually modified by the user after installation.

- %ALLUSERSPROFILE%\ADMIRE\admire.cfg
- %ALLUSERSPROFILE%\ADMIRE\mappings.txt
- %CommonProgramFiles%\ADMIRE\license.dat
- Any atlases, results, or other user-created data in ALLUSERSPROFILE%\ADMIRE, or %ProgramFiles%\ADMIRE which were not installed by the original installation.

ADMIRE does not use the registry. The uninstaller will not remove any files which have been modified by the user, or any folders which contain files which were not part of the original installation.

The Visual Studio 2017 Visual C++ Runtime redistributable packages will not be uninstalled, but this is rarely an issue, as many applications use these runtimes. However, if desired, they too may be uninstalled via *Add/Remove Programs*. However, doing so will very likely affect other installed applications.

2.7 Advanced

Under some circumstances, it may be desirable to have multiple installations of ADMIRE or to move executables or data files to support different deployment schemes. This may be particularly relevant when using ADMIRE in command line mode in an automated, scripting environment (see [Command Line Usage](#) for more detailed information about command line mode).

2.7.1 Current Working Directory

ADMIRE relies on the current working directory to search for its configuration and log files; everything that is installed into %ALLUSERSPROFILE% (except the demo patient data) must be in the current working directory. The executable, DLLs (everything in %CommonProgramFiles%\ADMIRE) can be located anywhere; simply use the fully-qualified path to the executable. The executable follows normal Windows DLL searching rules when looking for its dependencies (executable directory, current working directory, system directory, etc).

Atlases and patient data can be installed anywhere. Use fully-qualified paths when working in command line mode, or add the paths to the `admire.cfg` (manually, or via the GUI).

2.7.2 Unattended Installation

The installer supports various commandline parameters for automated deployment. For example, to install into `D:\ADMIRE` and configuration data into `D:\ADMIRE_DATA`, keeping an installation log at `C:\temp\admire_install.log`:

```
setup.exe /silent /dir="D:\ADMIRE" /ConfigDataDir="D:\ADMIRE_DATA" /log="C:\temp\admire_install.log"
```

When using ADMIRE in an automated workflow, it's often easiest to have everything in a single folder. The following merges both data and executables, does not install any Start Menu items, and will not even show a progress window or an error dialog:

```
setup.exe /tasks="" /suppressmsgboxes /verysilent /dir="D:\ADMIRE_V12" /ConfigDataDir="D:\ADMIRE_V12"
```

Note that **ConfigDataDir** is case-sensitive and is an ADMIRE-specific parameter. The installer is built using **InnoSetup** which supports **many more options** for unattended installation. You can also run `setup.exe /help` to see all of them.

2.7.3 Non-Administrative Installs

By default, the installer requires admin rights and is installed for All Users. However, it also supports Current User Only installations which don't require admin rights. If ADMIRE has not been installed before, the installer will ask which installation type is requested (and trigger a UAC if necessary). You can also force the installation type with the `/currentuser` or `/allusers` setup parameter.

2.7.4 License File

The license file, `license.dat`, must be located in the same directory as `ADMIRE.exe`. In a default installation, this is `%CommonProgramFiles%\ADMIRE\license.dat` and will be installed as Users-Modify. It is the only file in this directory that cannot be read-only.

3 Changelog

Log of important or user-visible changes for each version.

3.1 Changes for each version

Legend:

- **DCM**: related to DICOM functionality
- **SEG**: related to segmentation algorithms or the deformable registration process
- **GUI**: related to the graphical user interface
- **MISC**: miscellaneous; none of the above
- **DL**: related to deep learning

3.1.1 Most Recent Versions

3.1.1.1 ADMIRE 3.43 (Apr 2022)

- (SEG) To aid dose accumulation workflows, [DoseWarpUtil](#) can now warp dose to a user-specified image grid by specifying `--target-grid` and `--target-spacing`. Please read [Target Grid Considerations](#) for detailed usage guidelines.
- (GUI) Multiple dose files warped to the same patient will now share the same image grid: the patient image at the spacing of the source dose being warped. This is to aid dose accumulation workflows.
- (SEG) Added quality checking for constraint structures during Adaptive cases. Since constraints help guide, but do not hard restrict the registration process, comparing the overlap of the two constraints afterwards can help identify poor constraint contouring or registration errors. This is controlled by the `ADMIRE.Core.CheckConstraintSurfaceDeformationQuality` parameter.
- (MISC) [ConvertDCMtoITKVTk](#) now accepts -1 to indicate the original image spacing for that dimension, E.g., `--spacing -1 -1 3` will resample to 3mm slice spacing while leaving the X and Y dimensions alone.
- (MISC) Veenendaal project support

3.1.1.2 ADMIRE 3.42 (Mar 2022)

- (DL) Support anisotropic diffusion and scaling output median intensity for synthetic image models

3.1.1.3 ADMIRE 3.41 (Mar 2022)

- (SEG) Use the centroid of the smallest constraint ROI (if provided) as the starting alignment for rigid registration phase. (Inpatient)
- (SEG) Increase rigid registration initial iterations to improve starting alignment (Inpatient)
- (SEG) Improve Random Forest and Patch Fusion label fusion methods to better handle small ROIs
- (DCM) Fix calculation of new Bits Allocated (0028,0100) and Bits Stored (0028,0101) values when a derived image's intensity range exceeds the capacity of the original values in the source image. Previously either an exception would occur or intensity values would overflow resulting in artifacts.
- (MISC) [ConvertDCMtoITKVTk](#) will now flag extra ROIs that map to the same name; this is almost never a desirable outcome as it can result in label maps that don't include the expected ROI.

3.1.2 Older Versions

3.1.2.1 ADMIRE 3.40.1 (Feb 2022)

- (DL) Fix: Airscan submodel path handling

3.1.2.2 ADMIRE 3.40 (Feb 2022)

- (DCM) Fix: when specific constraint ROI names are provided on the commandline (e.g., `--constrain file.dcm ROI1 ROI2`), a bug in path comparison would result in ALL ROIs being included rather than the user-specified subset. Only affects command line mode—not GUI mode.
- (DL) Support for Airscan MR → sCT workflow with air pocket correction

3.1.2.3 ADMIRE 3.39.1 (Jan 2022)

- (DCM) Updated CBCT identification heuristics to better detect XVI CBCTs

3.1.2.4 ADMIRE 3.39 (Jan 2022)

- (DL) Support for applying N4 bias correction preprocessing to MR ImageSynth models
- (DL) Changed default Tensorflow memory allocation policy to AllowGrowth; improves initial model load startup time on GPU with a large amount of memory (e.g., >4GB)

3.1.2.5 ADMIRE 3.38 (Jan 2022)

- (DL) Support for pre-clamping intensity values used during ZScoreNormalization percentile calculation

3.1.2.6 ADMIRE 3.37 (Dec 2021)

- (SEG) Direct Dose Mapping (DDM) and Electron Mass Transfer (EMT) dose warping algorithms now available. Please read [DoseWarpUtil](#) for usage details. Previous dose warping algorithm is now deprecated.
- (SEG) Fix: Inpatient CT-CT algorithm selection incorrectly used inverse consistent DIR algorithm, regardless of ADMIREParameter.json settings. Regression was introduced in [ADMIRE 3.35 \(Sep 2021\)](#)

3.1.2.7 ADMIRE 3.36 (Dec 2021)

- (SEG) Fix: Largest connected component policy improperly applied to RF Label Fusion and Patch Fusion resulting in incomplete multi-component structures (e.g., RIBS, LUNGS)
- (SEG) Reduce peak memory usage of inverse DVF (inverse trilinear interpolation) method. This addresses OOM crashes/failures with certain GPU + image size combinations
- (DL) Fix: resample couch removal masks to match input image size if the input image is larger than the network model size. Only affects synthetic image models which are fed images that must be center-cropped to fit the model resolution

3.1.2.8 ADMIRE 3.35 (Sep 2021)

- (DCM) Update DCMTK to 3.6.6; addresses image load failures due to SIEMENS private tag encoding
- (DCM) Fix: DICOM loading from folder path lengths that exceed the Windows path limit of 260 characters. While images/RTSSs would write out correctly, they would fail to be subsequently loaded by ADMIRE.

3.1.2.9 ADMIRE 3.34 (Aug 2021)

- (DL) Fix: Patient masks created by some synthetic image models now reference the synthetic image in the exported RTSS, not the original patient source image
- (SEG) Fix: [MAFusionUtil](#) `--staple` no longer incorrectly uses the multilabel STAPLE algorithm (i.e., the one provided by `--mstaple`)

3.1.2.10 ADMIRE 3.33 (Aug 2021)

- (DL) Autosegmentation models operating at original DICOM resolution now downsample images that would exceed the network size after cropping

3.1.2.11 ADMIRE 3.32 (July 2021)

- (SEG) Fix: Crash during inpatient lung algorithm
- (DL) Support for autosegmentation models operating at original DICOM resolution
- (DL) Speed improvements and memory reduction for autosegmentation models with large number of classes
- (DL) Minor speed improvements and memory reduction for synthetic image models

3.1.2.12 ADMIRE 3.31.1 (July 2021)

- (DL) Fix: Crash during autosegmentation preprocessing in some cases

3.1.2.13 ADMIRE 3.31 (June 2021)

- (DL) Support for synthetic imaging DL models to use an autosegmentation model to generate patient masks
- (DL) Fix: For some synthetic imaging DL models, if the image FOV was too large for the model's FOV, a blank image or crash could occur
- (MISC) PercentageSame column added to CSV output for [ImageSimilarityUtil](#)

3.1.2.14 ADMIRE 3.30 (May 2021)

- (DCM) Warped dose (and associated warped plans) ProKnow compatibility; added Frame of Reference Module, non-empty Instance Number
- (DCM) SeriesDate and SeriesTime added to exported RTSS.
- (MISC) Removed error message newlines in [CompareContours](#) that could cause CSV parsing errors
- (MISC) Log files will now rotate out once cumulative size exceeds 100MB or 10 days worth.
- (MISC) GPU selection criteria improved to prefer better GPUs which might not be in TCC mode

3.1.2.15 ADMIRE 3.29 (Mar 2021)

- (DL) pseudo-MR models now supported. GUI labels changed from "sCT" to "synImage". Commandline `--type dl_sct` now deprecated in favor of `--type dl_synimage`.
- (MISC) Fix: crash in [StandardizeMR](#)

3.1.2.16 ADMIRE 3.28 (Feb 2021)

- (GUI) Fix: crash in Dice Calculation. Updated available metrics and column names to be consistent with [CompareContours](#). Added "Distance Percentile" input.
- (DL) Improvements for custom sCT models

3.1.2.17 ADMIRE 3.27.1 (Feb 2021)

- (MISC) [CompareContours](#) : Average Hausdorff distance supports `--distance_percentile`; added Balanced Average Hausdorff metric; updated column headers to be consistent among tabular, csv and json.
- (MISC) Removed unused crash reporting functionality; updated to Boost 1.73 and VTK 8.2.

3.1.2.18 ADMIRE 3.26 (Jan 2021)

- (DL) Improved preprocessing for MR autoseg models

3.1.2.19 ADMIRE 3.25 (Nov 2020)

- (SEG) Added `--skip-linear-reg` commandline option to bypass the initial linear (or rigid) registration phases in any inter/intrapatient deformable registration cases. Use it to avoid introducing error when atlas and patient images are already implicitly registered. If SROs are exported, they will have identity transformations. Has no effect on deep learning autoseg cases.
- (MISC) Removed Elekta-provided deep learning models to separate installer due to their size and infrequency of change.

3.1.2.20 ADMIRE 3.24 (Nov 2020)

- (DL) Added SurfaceSmoothingIterations parameter for DCNNSegmenter model descriptors
- (DL) [CBCTtoSCT](#) Fix: `--model` command line parameter handling

3.1.2.21 ADMIRE 3.23 (Oct 2020)

- (DCM) Fix: Export of deformed images failed with "std::logic_error Unexpected transformation type".

3.1.2.22 ADMIRE 3.22.2 (Oct 2020)

- (MISC) Fix: Wrong GPU selected in some configurations if one device is in TCC mode

3.1.2.23 ADMIRE 3.21.2 (Sept 2020)

- (DCM) Fix: [ImageSimilarityUtil](#) did not consistently apply "[Slightly](#)" [Oriented Images](#) workaround, resulting in "[↔ Images do not occupy same space](#)" exception when `--truthRTSS` is used.

3.1.2.24 ADMIRE 3.21.1 (Sept 2020)

- (DCM) Images warped by `--transform-atlas-images` or `--transform-patient-images` will now have the exact same origin/size/spacing as the destination frame of ref. This makes it consistent with applying the entire DVF (via `--deform-atlas-images`, `--deform-patient-images`). Previously, the destination size/origin was calculated by transforming the extents of the source image and taking the bounding box.

3.1.2.25 ADMIRE 3.20 (Sept 2020)

- (DCM) Exported Spatial Registration Objects will now set [Frame of Reference Transformation Matrix Type \(0070,030C\)](#) to RIGID if the transformation is orthonormal. Previously, all SROs were exported as AFFINE (because inter-patient linear registration generates an affine transformation), which, while mathematically correct, prevented import by some TPSs which only support RIGID.
- (DL) Added version 2 of the "EKT MR-sCT MalePelvis Unity T2 2min" MR→sCT DL model

3.1.2.26 ADMIRE 3.19.6 (August 2020)

- Fix: [ImageSimilarityUtil](#) Crash on startup

3.1.2.27 ADMIRE 3.19.5 (August 2020)

- Implemented workaround to enable better compatibility for contours and warped images generated from "slightly" oriented MR images. Added `OrientationCheck.exe` utility. See ["Slightly" Oriented Images](#) for more information.
- Upgrade Tensorflow to 1.15.3; Tensorflow.dll built with support for all SM versions (hence the large binary size). Avoids multi-minute first-time JIT on some GPU SM versions.
- Fix: Dice Calculation plugin crashes if multiple RTSSs are compared to the reference

3.1.2.28 ADMIRE 3.18 (Apr 2020)

- Added brachy breast DL model

3.1.2.29 ADMIRE 3.17 (Apr 2020)

- [CompareContours](#) : incorrect sample spacing used when inputs are binary masks and have larger spacing than the requested sample spacing
- SEG: Added experimental option: if `ADMIRE.Case.EXP_UseICAlgorithmForAdaptiveCTtoCT` is set to true in `ADMIREParameters.json`, all adaptive CT-CT cases will use the inverse-consistent DR method normally used for adaptive CT-MR cases. **USE WITH EXTREME CAUTION.**

3.1.2.30 ADMIRE 3.16 (Apr 2020)

- [CompareContours](#) : comparing surface masks with the same spacing, but different sizes cause crash in Hausdorff distance metric calculation
- (MISC) Doc updates

3.1.2.31 ADMIRE 3.15 (Mar 2020)

- [ConvertDCMtoITKVTk](#) : Enhancements
 - `--outputSurface` supports .ply, .stl mesh formats; Windows 10 has a built-in viewer
 - `--spacing` defaults to the input image resolution if unspecified. Be aware if bulk processing many `--src` inputs, this may result in each output having a different resolution.
 - added `{subdir}` and `{patientname}` format specifiers for output filenames
- [CompareContours](#) : `--spacing` defaults to resolution of `--truthImage` or `--truth` surface masks if omitted. Defaults to 2x2x2mm if only RTSSs or meshes (i.e., with no corresponding image).
- Crash report generation will only copy the two most recent log files instead of all of them. This prevents excessive delays on crash when an installation has a large amount of log files.

3.1.2.32 ADMIRE 3.14 (Mar 2020)

- [CompareContours](#) : Fix for statistics calculations when only binary mask surface representations are provided
- [ConvertDCMtoITK](#) : Fix. if `--outputImage` was not used and the output directory did not exist, the directory would not be created automatically
- (SEG) Fix Inter-patient GeneralSegmenterGPU exception "images do not occupy same physical space"
- (MISC) Added `ADMIREc.exe` and `ADMIREcw.exe` binaries that provide commandline-mode only executables with zero GUI and .NET dependencies. Useful for scripting environments. See [Other ADMIRE Binaries](#) for details.
- (MISC) Better error handling for parameter files being passed in via `--param`. Previously, invalid or missing files were silently ignored.

3.1.2.33 ADMIRE 3.13 (Mar 2020)

- [CompareContours](#) : Binary masks representations of surfaces are now supported—not just VTK meshes and DICOM contours.
- [ConvertDCMtoITK](#) : RTSSs are no longer required—images alone may be converted; fix VTK error message dialog popup

3.1.2.34 ADMIRE 3.12 (Feb 2020)

- [MAFusionUtil](#) : Multilabel STAPLE support added; prevents overlap of resulting ROIs

3.1.2.35 ADMIRE 3.11.2 (Feb 2020)

- (DCM) [DoseWarpUtil](#) now supports "warping" RTPLANS. Beam Dose Specification Point (300a,0082) and IsoCenter Position (300a, 012c) tags will be transformed and a new RTPLAN object created which is then associated with the warped RTDOSE. This is primarily to allow warped dose files to be imported into planning systems which require a separate plan for each dose.

3.1.2.36 ADMIRE 3.11 (Jan 2020)

- (SEG) Fix: Improved structure constraint performance
- (SEG) Fix: structure constraint naming on command line now uses case-insensitive matching
- (SEG) DVF inversion used during structure constraints in landmark lung and intrapatient (CT-CT,MR-MR) now also uses the inverse trilinear interpolation method introduced in 3.7.4. If you wish to revert to previous iterative algorithm, set `ADMIRE.Case.InvertDVUsingITI: false` in `ADMIREParameters.json`.
- (SEG) Fix: GPU out-of-memory when Unity or other very large MR images (e.g., larger than 512x512) used. Changed multiresolution pyramid resampling factor heuristics. Affects Intrapatient (CT-CT,MR-MR).

3.1.2.37 ADMIRE 3.10.1 (Dec 2019)

- (DCM) Fix: [CBCTtoSCT](#) was not applying `--patient-mask-threshold` optional override. (Regression in 3.7.4)

3.1.2.38 ADMIRE 3.10 (Dec 2019)

- (DCM) Fix: Incorrect values for dose grid dimensions (rows, cols, frame count, grid offset vector). (Regression in 3.9)
- (DCM) Fix: Position Reference Indicator (0020,1040) is Type 2 and should always be written, even if empty.

3.1.2.39 ADMIRE 3.9 (Nov 2019)

- (GUI) Added "Linear transform patient/atlas into atlas/patient frame and export new images" to Configuration → Options dialog. Similar to applying the DVF, but applies ONLY the linear registration. This functionality is also available on the command line via the `--transform-patient-images` and `--transform-atlas-images` commands.
- (DCM) [ImageWarpUtil](#) : `--transform` argument now supports spatial registration objects (not just DVFs). In addition, ITK formats are supported—not just DICOM.

3.1.2.40 ADMIRE 3.8 (Nov 2019)

- (DL) Updated to Tensorflow 1.15; deformable registration cases run after deep learning ones will now have all GPU memory available. Previously, there was no way to free GPU memory once it had been allocated by Tensorflow.
- (DL) [ConvertDCMtoITKVTk](#) : Added frequent item set mining and other statistical tools to help in preprocessing deep learning training data.
- (DCM) RTSTRUCTs now support Frame of Reference Module [added in DICOM2014c](#) instead of only the Referenced Frame of Reference sequence

3.1.2.41 ADMIRE 3.7.7 (Oct 2019)

- (DL) New MR-sCT model

3.1.2.42 ADMIRE 3.7.5 (Sep 2019)

- (DL) New Male Pelvis DL Autoseg model for MR; trained on T2 sequence images. Structures include External, Bladder, Rectum, Pubic Symphysis, Prostate, Femoral Joints, Seminal Vesicles, and Penile Bulb.

3.1.2.43 ADMIRE 3.7.4 (Sep 2019)

- (DL) **DL Male Pelvis**, **DL Head/Neck**, and **DL Thorax** GUI tabs merged into single tab, **DL Autoseg**
 - Each model now has an associated metadata .JSON file that allows it to function as a pseudo-atlas. This file determine which anatomical region the model is for, supported modalities, ROI Names, etc.
 - Models moved from the .EXE installation folder (default is C:\Program Files\ADMIRE) to the configuration data folder (default is C:\ProgramData\ADMIRE\DemoAtlases\DL Autoseg Models)
 - Commandline mode: `--type dl_(hn|malepelvis|thorax)` deprecated; use `--type dl_↵ autoseg` and an explicit model as the atlas to determine region.
- (DL) CBCT-to-sCT DL model now available in ADMIRE GUI via **DL sCT** tab.
 - Each model now has an associated metadata .JSON file that allows it to function as a pseudo-atlas.
 - Models moved from the .EXE installation folder (default is C:\Program Files\ADMIRE) to the configuration data folder (default is C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models).
 - Commandline mode: new case type added to ADMIRE.exe commandline (`--type dl_sct`).
- (DL) **CBCTtoSCT** `--region X` arguments deprecated in favor of explicitly providing model metadata file, e.g., `--model c:\path\to\model.pb.json`. See **CBCTtoSCT** for backwards compatibility information.
- (SEG) ADMIRE.exe: DVF inversion now performed by inverse trilinear interpolation method. If you wish to revert to previous algorithm, set `ADMIRE.Case.InvertDVFFUsingITI: false` in `ADMIREParameters.↵ json`. For implementation details, please read [Deformation field validation and inversion applied to adaptive radiation therapy](#). Tom Vercauteren et al 2013 Phys. Med. Biol. 58 5269.
- (DCM) Fix: ROI Physical Properties from newer DICOM standard versions were causing RTSS load failure
- (MISC) Replace NLog logging system with Boost logging.

3.1.2.44 ADMIRE 3.6

- (DL) **CBCTtoSCT** : New model for head/neck region; to use it, specify `--region headneck`
- (MISC) **CompareContours** : Fix crash when two truth surfaces map to same semantic name

3.1.2.45 ADMIRE 3.5

- (DL) Added `--patient-mask-threshold` command line argument to **CBCTtoSCT**
- (MISC) Upgraded to Innosetup installer 6.0.1; Installer now supports both All Users and Current User Only installation modes.
- (MISC) Upgraded to NLog 4.6.2.

3.1.2.46 ADMIRE 3.41 (Mar 2022)

- (DL) Fix: corrupted model files in installer package
- (MISC) Upgraded CrashRpt

3.1.2.47 ADMIRE 3.4 (Apr 2019)

- (DL) [CBCTtoSCT](#) : New model for thorax region; to use it, specify `--region thorax`
- (DL) [CBCTtoSCT](#) : Pelvis region model (`--region pelvis`) updated with new training data;
- (DL) Autoseg now suggests reasons for final result errors (e.g., low confidence, tiny ROI, ...)
- (MISC) Temporarily disabled async logging while investigating crash; upgraded to NLog 4.6.1; `${cwd}` layout replaced by `${currentdir}` in `NLog.config`

3.1.2.48 ADMIRE 3.3.1 (Mar 2019)

- (DL) Reverted change in 3.2: Dummy atlases once again affect what ROIs appear in DL results.

3.1.2.49 ADMIRE 3.3 (Mar 2019)

- (DL) "Add to structure set" now works for deep learning cases same as non-DL cases
- (DL) Plugged memory leak
- (SEG) Better RFLableFusion and PatchFusion support of ROIs with more than 1 connected component (e.g., RIBS, LUNGS)

3.1.2.50 ADMIRE 3.2 (Mar 2019)

- (MISC) Fix: [CompareContours](#) and Dice Calculation tool crash
- (DL) Dummy atlases no longer have any effect on what ROIs appear in DL results

3.1.2.51 ADMIRE 3.1 (Mar 2019)

- (DL) Trained new [CBCTtoSCT](#) model. Couch is removed from image. Added caching to speed up refinement of multiple image sets. Read [CBCTtoSCT](#) for more.
- (DL) Upgraded to Tensorflow 1.12
- (DL) Fix: If an autoseg result for an ROI comes back empty (not in region, probability too low, etc), it is skipped instead of throwing an exception.
- (MISC) [ImageSimilarityUtil](#) no longer requires an RTSS and can operate purely on two images; additional stats.

3.1.2.52 ADMIRE 3.0 (Nov 2018)

- (MISC) Deep learning CBCT to synthetic CT creation. Read [CBCTtoSCT](#) for more.
- (MISC) Upgraded CUDA to 10.0
- (SEG) Fix: Landmark Lung segmentation crash on synthetic CT input
- (DCM) Fix: RT ROI Interpreted Type value of NONE causes SS load error
- (MISC) [CompareContours](#) can export statistics in JSON format; supports ITK/VTK file formats directly

3.1.2.53 ADMIRE 2.5 (Aug 2018)

- (SEG) Deep Learning algorithms. ADMIRE now requires a GPU with compute level of 3.0 or higher. Use the "DL (region)" tabs (or "dl_" region types if using commandline). Read [Deep Learning Autosegmentation](#) for more.
- (MISC) Now built with Visual Studio 2017; upgraded following third party dependencies
 - Upgraded ITK to 4.13.1
 - Upgraded VTK to 8.1.0
 - Upgraded DCMTK to 3.6.3
 - Upgraded Boost to 1.68
 - Upgraded CUDA to 9.2
- Fix: Invalid contours/artifacts created during surface generation on results from all label fusion methods

3.1.2.54 ADMIRE 2.03 (June 2018)

- (DCM) Fix: Previous version introduced bug where Contour Data -> Contour Image Sequence was incorrect, resulting in contours being drawn incorrectly in some viewers (e.g., Monaco).

3.1.2.55 ADMIRE 2.02 (Feb 2018)

- (SEG) Label fusion cases (STAPLE, Patch, RF) fix: contour artifacts at image border; if all ROIs are identical (e.g., CT external SKIN), fusion is skipped to avoid discretization error
- (MISC) image preprocessing, contour generation performance improvements
- (MISC) Dice Calculation GUI tool: volumes were not scaled to cc; output now consistent with CompareContours.exe output (3 decimal places);
- (MISC) CompareContours.exe: fix: 2D ROIs were not handled correctly
- (MISC) StandardizeMR.exe: added `--reorientToDicomRCS`, `--skipBiasFieldCorrection`, `--skipDenoising`, `--skipMedianScaling` commandline options
- (MISC) Upgraded to NLog 4.4.12
- (MISC) HTTP crash report submission is functional again

3.1.2.56 ADMIRE 2.01 (Aug 2017)

- (SEG) [StandardizeMR](#) tool included

3.1.2.57 ADMIRE 2.0 (July 2017)

- (SEG) New inpatient feature-based lung segmentation for CT and CBCT. This is an expansion and evolution of the approach which won the [MICCAI 2010 EMPIRE10 challenge](#). Available in GUI (Adaptive Lung) and command line (`--ip one-to-all --type lung`). See [Whitepapers & References](#) for technical information.
- (SEG) New random forest label fusion method with auto-training for multi-atlas fusion cases. It is a general-purpose method like STAPLE and patch fusion. Available in GUI and commandline (`--rflabel-fusion`). See [Whitepapers & References](#) for technical information.
- (SEG) Improved surface triangle mesh creation performance
- (SET) Fix: inpatient mixed modality GPU memory usage estimation was inaccurate
- (SEG) Upgraded to Insight Toolkit 4.11.
- (SEG) Fix: GPU out-of-memory errors for CT-MR/MR-CT inpatient segmentation
- (SEG) Added [ImageWarpUtil](#) tool for standalone warping of images
- (MISC) Updated to current Elekta branding guidelines
- (DCM) Structure sets and image sets are cached between intermediate cases

3.1.2.58 ADMIRE 1.14.1.1 (Apr 2017)

- (MISC) Rebuild of 1.14.1 with support for all GPU virtual architectures supported by CUDA 6.5 SDK (`compute_13, compute_20, ..., compute_52`). Pre-Fermi GPU (`compute_13`, e.g., Tesla C1050/70) support is re-enabled, but deprecated, as NVIDIA has removed support after CUDA 6.5 SDK. Future planned ADMIRE features will require CUDA SDK 8.0, and as such, will necessitate dropping support for pre-Fermi GPUs.

3.1.2.59 ADMIRE 1.14.1 (Mar 2017)

- (SEG) STAPLE statistics now reports likelihood metric
- (SEG) STAPLE statistics can now be saved in JSON format (MAFusionUtil only)
- (MISC) Reduce logging overhead by using buffered Targets

3.1.2.60 ADMIRE 1.14 (Nov 2016)

- (SEG) Fix: Registering identical images (inpatient) generates non-zero DVF
- (SEG) Re-enabling support for pre-Fermi GPUs
- (DCM) Fix: Non-transverse warped images are now sliced correctly
- (DCM) Fix: Metadata cache constantly invalidates if an RTPLAN is cached
- (DCM) Tolerance for comparing direction cosines for equality (minor floating point deviations within an oriented MR)
- (MISC) Each application (ADMIRE.exe, compareContours.exe, etc) now has it's own log file (e.g., `admire.lastExec.log`, `compareContours.lastExec.log`)

3.1.3 Really Old Versions

3.1.3.1 ADMIRE 1.13.5 (May 2016)

- (MISC) Rebuild of 1.13.4 with support for pre-Fermi GPUs (compute_13)

3.1.3.2 ADMIRE 1.13.4 (May 2016)

- (SEG) Fix: crash when slicing ROIs with zero contours

3.1.3.3 ADMIRE 1.13.3 (Apr 2016)

- (SEG) Standalone STAPLE commandline utility (MAFusionUtil)
- (SEG) Better support for oblique contour surface reconstruction
- (SEG) Fix: Rigid registration SROs were not orthonormal if images had anisotropic spacing
- (MISC) Added "non-clinical use" branding

3.1.3.4 ADMIRE 1.13.2 (Mar 2016)

- (DCM) Fix: ROIs with no contours or inconsistent Reference Frame of Reference values abort case
- (MISC) User may choose a different configuration data directory during installation

3.1.3.5 ADMIRE 1.13.1 (Feb 2016)

- (SEG) Improvements and fixes to intrapatient segmentation
- (SEG) Fix: poor quality DVF results in intrapatient pelvic area cases using constraints
- (SEG) Fix: over-aggressive surface mesh simplification
- (DCM) Fix: Exported Spatial Registration Objects missing translation component
- (DCM) Config option to associate DREG and SREG with either patient or atlas
- (DCM) Warped dose dummy plans have iso centers warped
- (DCM) Exported deformed images series date/time, description improvements
- (MISC) User-configurable zero-point ROI propagation
- (MISC) New analysis tool: [ImageSimilarityUtil](#)
- (MISC) Fix: UNC paths handling regression

3.1.3.6 ADMIRE 1.12 (July 2015)

- (GUI) Fix: For some inpatient data organizations, incorrect image set would be used during segmentation
- (SEG) Reduced STAPLE execution time
- (SEG) Fix: `--save-intermediate-results` command line argument stuck on for composite case types (multi-series + multi-atlas)

3.1.3.7 ADMIRE 1.11.1 (Jun 2015)

- (GUI) Additional statistics added to Dice Calculation tool (Cohen's Kappa, etc)
- (MISC) MS VC++ Runtime installation improvements, fixes
- (SEG) Minor DVF inversion improvements

3.1.3.8 ADMIRE 1.11 (May 2015)

- (GUI) Better icons in Adaptive Tool patient info tree
- (GUI) DICE calculation tool is simplified; new statistics added
- (DCM) More plan and dose metadata is cached
- (DCM) Fix: Spatial Registration objects had frames of reference reversed
- (DCM) Deformable Spatial Registration object fixes
- (DCM) Fix: duplicate items in Contour Image Sequence causes Structure Set Loading Error
- (SEG) Dose deformation added to Adaptive Tool ([info](#))
- (SEG) Improved surface reconstruction (from 2D contours, see [Whitepapers & References](#))
- (SEG) Structure set constraints for MR-CT, CT-MR inpatient cases are now supported
- (MISC) Fix: .NET 4.5 detection in installer

3.1.3.9 ADMIRE 1.10.02 (Dec 2014)

- (GUI) Fix: Merge RTSS in Adaptive Tool was not working
- (DCM) Patient Position (0018,5100) is now copied from the source DICOM for any derived images.
- (DCM) Patient Orientation (0020,0020) is derived from the Image Orientation (Patient).
- (DCM) Fix: DICOM cache gets out of sync (and case subsequently fails verification) if an image is removed or renamed.
- (SEG) Reduced GPU memory usage for Adaptive (CT-MR, MR-CT) cases.
- (SEG) Adaptive (CT-MR, MR-CT) cases now use inverse-consistent rigid registration.

3.1.3.10 ADMIRE 1.10.01 (Sep 2014)

- (GUI) Atlases are no longer restricted to a single RTSS in a folder. Atlas controls show more information.
- (GUI) DICE Calculation now loads files correctly.
- (GUI) Additional image modality underlays
- (SEG) Improvements and fixes to Adaptive algorithms.
- (DCM) Changed output file naming policy. Timestamp and case configuration are now included.

3.1.3.11 ADMIRE 1.09.01 (May 2014)

- (GUI) Main patient data control can now show all 4D image sets that have been binned into subfolders, but share the same series instance UID; Adaptive tool will still only show the first one, however.
- (GUI) Fixed superfluous patient and atlas selection controls refreshes.
- (DCM) Improved DICOM caching; cache file will rebuild upon startup.

3.1.3.12 ADMIRE 1.08 (May 2014)

- (GUI) Added "Experimental" option group to expose temporary or new functionality
- (SEG) Adaptive cases now support structure constraints. Can be enabled via an experimental option.

3.1.3.13 ADMIRE 1.07 (Apr 2014)

- (GUI) Error reporting was not including *.log files. Improved video recording.
- (DCM) DICOM Deformable Registration Grid sequence was not written correctly when exporting Deformable Spatial Registration objects.
- (DCM) Relaxed RTSS merge constraints to only require same Frame of Reference as patient image set.

3.1.3.14 ADMIRE 1.06 (Apr 2014)

- (GUI) One-to-all is now the default multi-patient method.
- (GUI) RCCT Processor plugin shows additional information in the table to aid binning. No longer limited to operating on CT images—it's more of general binning tool now.
- (GUI) Patient, Atlas listings automatically refresh upon case completion.
- (MISC) DREG,SREG,DVF outputs for multi-patient cases no longer always write to the first patient folder when saving results to patient folder.
- (DCM) All deformed CT atlas/image sets are no longer erroneously identified as CBCT in the GUI.

- (DCM) Deformed CT image sets output from an "Inpatient (CBCT)" case will now have an DICOM Image Type of CONEBEAM added. For other case types, the standard CBCT detection heuristics are used.
- (DCM) Output filenames no longer truncate the patient name and atlas name. Filenames should now be more readable.
- (DCM) DICOM metadata is now cached and saved between application launches. Greatly improves GUI startup and interactivity. The cache file is written to the application data directory.
- (GUI) Additional DICOM fields are displayed in the Detailed Image Information pane
- (GUI) Added "Open configuration data folder" File menu item
- (GUI) Added "Clear DICOM cache" File menu item
- (MISC) Added documentation for file naming conventions and result locations.

3.1.3.15 ADMIRE 1.05 (2014 Jan)

- (GUI) Additional fixes related to file paths with lengths longer than 260 characters

3.1.3.16 ADMIRE 1.04 (2014 Jan)

- (MISC) File paths with lengths longer than 260 characters are now supported. Note that files written this way does not guarantee other applications will be able to handle them correctly. For example, the Windows Explorer will allow copying of these files to a new location. However, renaming them in the new location would fail because Windows Explorer will not allow a user to create paths that are greater than 260 characters.
- (MISC) Improved speed of showing DICOM data in the GUI.
- (DCM) Fix: Exporting deformed image sets whose original referenced DICOM image was an unsigned pixel type could fail due to interpolation forcing values negative.
- (DCM) Updated Manufacturer, ManufacturerModel, and SoftwareVersion values used when exporting any DICOM files. Exported deformed images now have these tags as well.

3.1.3.17 ADMIRE 1.03 (2014 Jan)

- (SEG) Fixed regression that prevented creation of Adaptive cases with a CBCT subject and Brain cases.
- (DCM) Fix: DICOM Image Sets aren't as strict requiring Pixel Spacing and Bit values consistency.
- (DCM) Fixed regression in 1.02 that caused some CT image sets to not load correctly
- (GUI) Improved startup error reporting
- (MISC) Documentation regarding CBCT image detection behavior [added](#).

3.1.3.18 ADMIRE 1.02 (2014 Jan)

- (SEG) DVF memory usage is reduced for certain cases. Exported DVFs may now be smaller (i.e., coarser sampling) for some cases.
- (SEG) DVF inversion (Patch Fusion, export) will now downsample the input DVF if there is not enough GPU memory.
- (MISC) EditAdmireLicense.exe re-added to installer package.

3.1.3.19 ADMIRE 1.01 (2013 Nov)

- (SEG) Enabled MR-MR for inpatient. Uses the inverse-consistent algorithm introduced in the previous version.
- (DCM) Non-orthogonal (oblique) image rotations are now supported for subject and atlas images.
- (GUI) Fix: Adaptive Tool crashed trying show image sets with a DICOM time that contained fractional seconds, e.g., 12:00:01.334.
- (GUI) DiceCalculation tool now saves results in proper CSV format.
- (MISC) Updated [commandline usage documentation](#)

3.1.3.20 ADMIRE 1.00.00 (2013 Oct)

- (MISC) Project name changed to ADMIRE to reflect the research direction of the application as opposed to ABAS, the commercially-released product.
- (SEG) Inpatient CT-MR segmentation now supported. The new algorithm requires the GPU and is inverse-consistent. Algorithm is automatically chosen if the atlas is MR and the patient is CT (via the Adaptive tab).
- (SEG) Speed improvements for applying DVFs to structures, prostate refinement, CT preprocessing, surface-to-mesh conversion, various image processing filters.
- (DCM) Speed improvement to RTSS export with many ROIs.
- (SEG) Fix: Improved linear registration for non-CT modalities.
- (MISC) Speed improvements in DiceCalculation plug-in.
- (MISC) Addition of crash/error reporting in Help menu (Diagnostics.dll)
- (SEG) Segmentation code refactored into Segmentation.dll; overall application size is reduced
- (SEG) Upgraded to CUDA 5.5; minimum GPU requirement is now hardware version 1.3.
- (MISC) New logging library (CmsLogging.dll, CmsLoggingCLR.dll, NLog.dll)
- (MISC) New installer packages can migrate previous ABASProto installation. Demo Data moved to separate package.

3.1.3.21 ABASProto8 0.64 (2013 Sept)

- (SEG) Fix: Patch fusion and STAPLE could throw exception or crash if zero or only one input structure pre-processed correctly. This can happen when there is a large number of extremely small structures. This was introduced in 0.62.

3.1.3.22 ABASProto8 0.63 (2013 Aug)

- (SEG) Fix: Regression with raw DVF export file naming. MHD base filename was incremented as if a naming conflict occurred, when in fact, none did.

3.1.3.23 ABASProto8 0.62 (2013 Aug)

- (SEG) Fix: Patch Fusion no longer aborts case if a small structure cannot be discretized during binary mask creation. Instead, the structure is excluded from the patch fusion inputs.
- (SEG) Fix: STAPLE for a structure excludes bad inputs rather than failing the entire structure.
- (SEG) Fix: Commandline inpatient cases can now be created with single patient + atlas, matching GUI behavior.
- (SEG) Fix: Commandline `--patch-fusion` and `--staple-stats` ignored `--results` option, and instead always used the `results_dir` specified in the configuration file.
- (GUI) Fix: Main/Adaptive dialog would not allow creation of both a STAPLE and patch fusion case with identical inputs within the same batch.
- (GUI) Fix: Result Date/Time column in Status dialog would not update properly if segmentation recoverable errors occurred.
- (SEG) Structure sets are now created and saved immediately after segmentation, rather than after all post-segmentation actions (DVF export, etc). This allows structure sets to be created, even if the other actions fail.

3.1.3.24 ABASProto8 0.61 (2013 July)

- (GUI) New configuration option "Save intermediate results" enables skipping over any structure sets, DVFs, etc generated by intermediate cases created during Patch Fusion and STAPLE cases. Also available via command line option.
- (GUI) New configuration options to individually toggle output of the following: DICOM DVF, inverse DICOM DVF, spatial registration, inverse spatial registration, raw (ITK) DVF. Note that these options respect the "Save intermediate results" option. Also available via command line options.
- (GUI) Case number column added to Status dialog.
- (SEG) Fix: Structure mapping was not case-insensitive under rare circumstances in Dice Calculation, STAPLE, and Patch Fusion.
- (SEG) Inpatient segmentation now uses same SKIN contour generation as other segmenters. This only applies if the patient image is CT and the atlas contains an ROI that maps to SKIN.
- (SEG) Any failure during STAPLE statistic handling no longer fails case. Improved error handling.
- (DCM) ROIs in Patch fusion results are now sorted in same order they appear in reference atlas. This matches STAPLE.

3.1.3.25 ABASProto8 0.60 (2013 Jun)

- (GUI) Many dialogs no longer remain always on top
- (GUI) Added "Explore to..." context menu item to many controls which have atlases or patients
- (GUI) "Select Structures" dialog is now lexicographically sorted
- (SEG) STAPLE and Patch Fusion now skip over ROI's that caused a failure with a warning. Previously, the case was aborted.

3.1.3.26 ABASProto8 0.59 (2013 May)

- (SEG) Improved patch fusion result quality
- (SEG) Fix: Memory leak in patch fusion
- (GPU) Fix: GPU memory leak in DVF inversion if exception occurs

3.1.3.27 ABASProto8 0.58 (2013 May)

- (GUI) Adaptive Tool shows more study and series information in the data tree
- (GUI) Fix: Creating multi-series (one-to-all,etc) cases no longer creates individual cases as if the "Single" case type were chosen.

3.1.3.28 ABASProto8 0.57 (2013 May)

- (SEG) New: ROIs with a single point (markers, points, etc) are now supported.
- (SEG) Improved handling of small ROIs so they are no longer discarded as part of ROI contour cleanup.

3.1.3.29 ABASProto8 0.56 (2013 May)

- (SEG) New: Inverse spatial registration and deformable spatial registration can now be exported.

3.1.3.30 ABASProto8 0.55 (2012 Oct)

- (SEG) Fix: Patch fusion no longer executes only on the largest surface if input structure has multiple connected components (e.g., ribs, where the vertebrae are segmented separately)
- (SEG) New: Added RIBS mapping

3.1.3.31 ABASProto8 0.54 (2012 Aug)

- (SEG) New: Patch fusion now available for all case types (rcct, intra, etc) via commandline
- (SEG) Improved `-phases` commandline argument parsing
- (SEG) Speed improvements while applying deformation field to structures for inpatient and demons segmenters.
- (MISC) GPU detection no longer chooses devices which are unavailable due to being in exclusive mode.
- (MISC) Added `-gpu-info` commandline argument for displaying detailed information on all detected GPUs.
- (SEG) Fixed rare non-convergence with STAPLE

3.1.3.32 ABASProto8 0.53 (2012 Jun)

- (SEG) Fix: SKIN mapping wasn't properly recognized in some segmenters under some circumstances.
- (SEG) Improved affine linear registration for certain cases

3.1.3.33 ABASProto8 0.52 (2012 May)

- (GUI) "Adaptive" and "Adaptive (Other)" tabs changed to "Adaptive" and "Adaptive (CBCT)". Use "Adaptive" for CT and MR images. Use "Adaptive (CBCT)" to force CT images to be treated as CBCT.
- (GUI) New: Patch fusion now available in Adaptive Tool plugin/dialog.
- (SEG) New: MR support for inpatient cases. Use the "Adaptive" tab.
- (GUI) Fix: Adaptive cases with only 1 atlas and 1 subject are now explicitly supported in both main GUI and Adaptive Tool.
- (GUI) Adaptive Tool and main GUI will generate multiple cases if multiple atlases/subjects are selected and STAPLE/Patch fusion is NOT chosen.
- (MISC) Collected miscellaneous documentation, whitepapers, etc into a Windows Help File which is now installed into the CMS folder on the start menu and available in the application via the Help menu.

3.1.3.34 ABASProto8 0.51 (2012 May)

- (SEG) New: Patch fusion support for RCCT and inpatient case configurations. Again, currently only available in GUI.
- (GUI) Fix: Patch fusion and STAPLE case status lines now appear in the proper order in the case status dialog instead of always appearing at the end of the list.

3.1.3.35 ABASProto8 0.50 (2012 May) (internal only)

3.1.3.36 ABASProto8 0.49 (2012 Apr)

- (DCM) Fixed crashing/case failure bug in handling MR image data with slope or intercept. Slope and intercept are not valid MR modality DICOM, but are used by some scanners regardless.
- (SEG) Beta implementation of patch fusion for standard cases. Currently only available in the GUI and for standard case types (i.e., not RCCT or IP cases).

3.1.3.37 ABASProto8 0.48 (2012 Mar)

- (SEG) Major fix for both GPU and CPU Demons segmentation. Results should be much improved.
- (SEG) Linear registration improved for cases where subject image is significantly smaller than atlas.

3.1.3.38 ABASProto8 0.47 (2012 Mar)

- (DCM) New: Atlas image sets can be deformed onto the patient image and exported as a new DICOM image set. Commandline and configuration options added. Requires GPU for deformation vector field inversion.
- (GUI) Fix: Plugins use application icon. User-facing file paths displayed in OS native format.

3.1.3.39 ABASProto8 0.46 (2011 Oct)

- (GUI) Fix: Cases created with Adaptive Tool cause crash if dialog is dismissed before executing case.
- (DCM) New: Patient image sets can now be deformed onto the atlas and exported as a new DICOM image set. Commandline and configuration options added.
- (DCM) Fix: Under some circumstances, decimal strings (DS VR) could be created which were longer than allowed by the standard. Only affected Spatial Registration objects.
- (GPU) Fix: Minor updates in preparation for upgrading to CUDA 4.0.
- (MISC) New: Commandline support for Visual Studio style response files via -@

3.1.3.40 ABASProto8 0.45 (2011 Sept)

- (GUI) Paths display using native OS format in the atlas/patient select dropdowns and configuration edit dialog.
- (GUI) Configuration errors are displayed on startup.
- (MISC) More robust handling of invalid ADMIRE.cfg data.
- (DCM) Spatial registration export path handling now works correctly when writing results to the patient data folder.
- (MISC) Deformable field export path handling now works correctly in commandline mode when writing results to the patient data folder.

3.1.3.41 ABASProto8 0.44 (2011 Aug)

- (SEG) Allowed MR cases to be used with GPU intrapatient segmenter (aka "Adaptive Other")
- (SEG) Single-point and otherwise invalid contours in atlases no longer cause entire structure to be discarded. Instead, just the contour is ignored.
- (DCM) Initial support of exporting Spatial Registration objects
- (DCM) Initial support of exporting Deformable Spatial Registration objects
- (DCM) DICOM functionality now distributed as CmsDicom.dll.
- (DCM) Fixed error logging that caused excessive CPU usage when browsing/loading a large number of DICOM files whose transfer syntax did not match actual encoding.
- (GUI) Ellipses added in menu items where appropriate.

3.1.3.42 ABASProto8 0.43 (internal only)

4 GPU Frequently Asked Questions

Frequently asked questions and answers regarding ADMIRE's GPU support.

4.1 What GPU Should I Buy?

The perennial question. There are [too many GPUs](#) to list them all. However, at minimum:

- NVIDIA GPUs supported by CUDA 10.0
- Compute capability 3.0 - 7.5 ([Kepler](#), [Maxwell](#), [Pascal](#), [Volta](#), [Turing](#)).
- 4+ GB of memory is recommended for deep learning algorithms.
- 3+ GB of memory for deformable registration algorithms.

Choose the most powerful you can, subject to your budget, chassis, and availability. A [spreadsheet of our suggestions](#) is available.

4.2 Hardware

- **What brand GPU should I buy?**

NVIDIA. All of ADMIRE's GPU functionality is written using NVIDIA's Compute Unified Device Architecture (CUDA) toolkit. GPUs from other manufacturers (ATI, Intel) will not work.

- **Can I use video cards from PNY, XFX, ASUS, MSI, etc?**

Yes, as long as it has an NVIDIA GPU. NVIDIA does not actually build the video cards themselves. Instead, the company sells the GPUs to third parties that actually design and assemble the rest of the video card.

- **Should I use the GeForce, Quadro, or Tesla line?**

Any. As we are still in the research phase, we have found the GeForce line to be cost-effective and as performant as the Quadro/Tesla product line. Currently, ADMIRE does not require one or the other.

That being said, according to NVIDIA, Quadro-series cards are wholly assembled and stress-tested by NVIDIA. In video gaming uses (the target market for the GeForce), a transient memory error would most likely be inconsequential, since an incorrect pixel color probably won't be noticed, especially since new frames are constantly being rendered. But for GPU computing, a transient memory error becomes much more relevant (though extremely rare).

- **Can I put a GPU in a rack-mount server?**

Depends. Early GPUs, especially the GeForce line, were double-height cards that required a large amount of space in the case. Many newer models are single-height and much shorter. In some rack-mount servers, riser cards are available. However, verify the power-requirements of the GPU can be met. Early models could require upwards of 400W and needed supplementary power cables (either 4, 6, 8 pin). Check with the manufacturer of your server to see what GPUs are certified.

- **What Compute Capability does ADMIRE require?**

Compute Capability 3.0 (Kepler) or higher. Previously, we only required 1.3, but NVIDIA has removed support for older hardware.

- **What does "Compute Capability" mean?**

Compute Capability identifies an NVIDIA GPU's hardware capabilities. Each version corresponds roughly to a generation of GPUs. The version determines which [features are available to use via CUDA](http://en.wikipedia.org/wiki/CUDA#Version_features_and_specifications).

- **Can I use my laptop's GPU?**

Yes; if it is NVIDIA and meets the above requirements.

- **My laptop video memory is "shared". What does that mean?**

For space, power, and cost savings, many laptop GPUs will use the main system memory, rather than having separate memory (also referred to as "discrete"). For a majority of users, this presents no problems. However, due to the high memory requirements for image processing, this is not a recommended configuration for ADMIRE.

- **I have multiple video cards with NVIDIA GPUs in my workstation. Which one will ADMIRE use?**

On startup, ADMIRE chooses the "best" device by comparing the theoretical maximum throughput (GPU speed * GPU Cores) of each, giving preference to the one with more memory.

- **Can ADMIRE utilize multiple GPUs?**

No. At this time, ADMIRE will use only a single GPU. Even if multiple GPUs are bonded together in SLI mode, the CUDA driver enumerates the individual GPUs; SLI mode is only relevant for OpenGL and DirectX.

- **Can I have the GPU installed as well as my existing video card?**

Yes. This is, in fact, the recommended configuration.

4.3 Installation

- **Which operating systems are supported?**

- Windows 7 64-bit or newer

- **What about Windows Vista?**

Unsupported. It probably works, but it is not tested.

- **Is any additional software required (other than ADMIRE)?**

Yes. A NVIDIA device driver with CUDA support must be installed by a user with administrator access-it cannot be installed by the ADMIRE installer. Normally, this is already present if the drivers are relatively up to date on the machine.

- **How do I determine if I need to install the driver?**

Depending on how new the computer is, the original system builder may have already installed drivers that support CUDA. To verify, perform the following steps:

1. Start menu -> Control Panel -> NVIDIA Control Panel -> Help -> System Information.
2. Choose the Components tab.
3. Beneath the 3D Settings heading, verify that the NVCUDA.DLL is present and has a product name of "↔ NVIDIA CUDA 10.0 driver" or higher version.

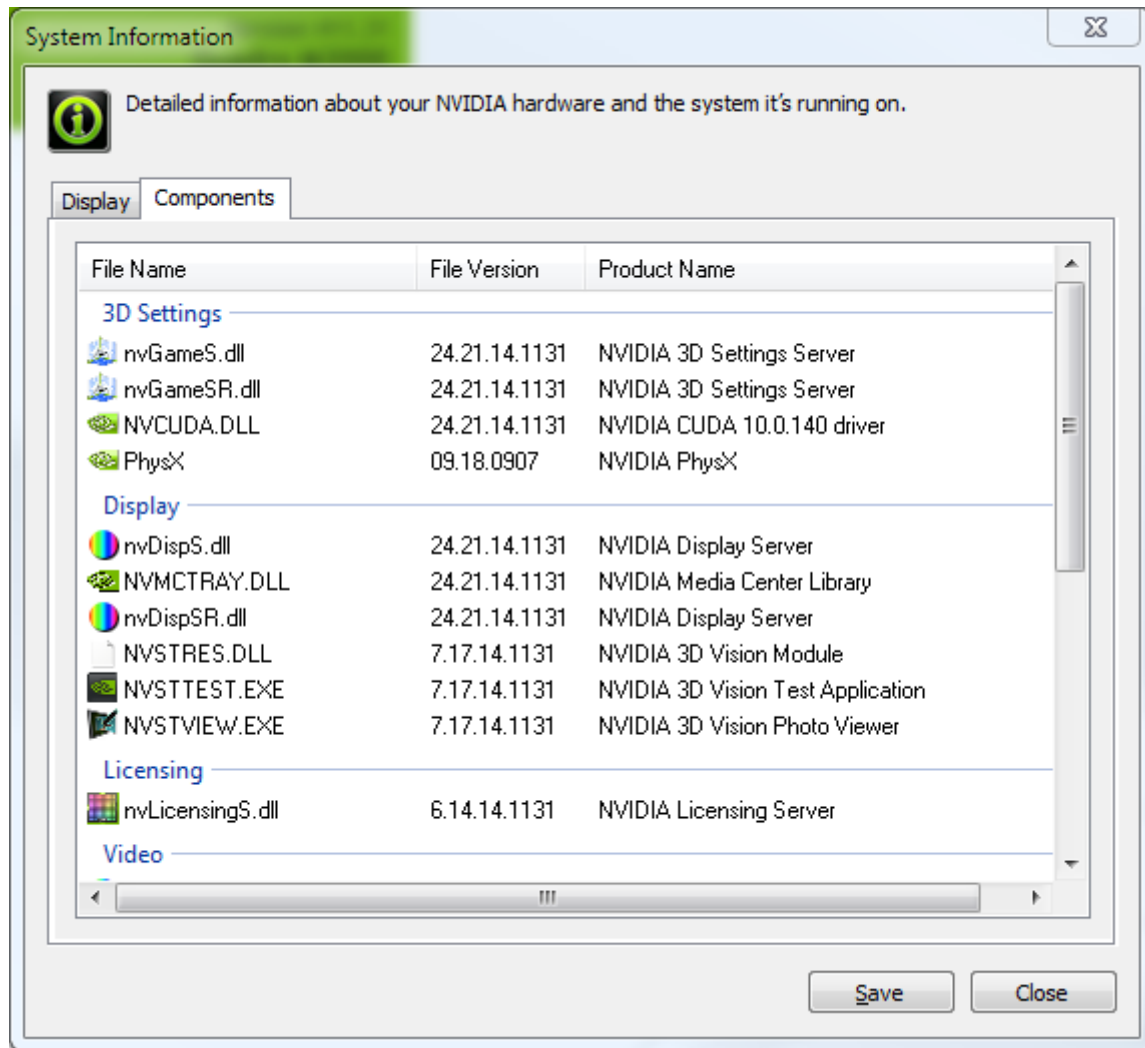


Figure 1 Verifying CUDA driver support



- **How do I install the display driver?**

1. Verify you have Administrator privileges, otherwise you will be unable to install the driver.
2. Visit the [Driver Download section on NVIDIA's website](#)
3. Choose the product and operating system that matches the computer.
4. After downloading, start the driver installation and follow the instructions.
5. Reboot the computer after installation finishes.

- **Is VMWare supported?**

Sort of. Virtualization products (VMWare, Parallels, VirtualBox, Windows Server with Hyper-V), by their nature, must assume direct control of physical hardware in order to share access among virtual machine instances. Some of these products support "direct mapping" of a PCI device into a single VM instance. This typically requires certain CPU or motherboard chipset capabilities. Read more on VT-d extensions and VMWare's VMDirectPath.

4.4 Remote Access

- **Can I use GPU functionality when running ADMIRE via Terminal Services, Remote Desktop, or Citrix?**

In order to provide high-performance, Microsoft Terminal Service-based technologies operate by intercepting drawing commands and sending them to a remote location. Typically, when these services are active, applications executing on the remote machine will not have direct access to the drivers needed to support GPU computing. In this situation, ADMIRE will not detect the CUDA driver is unavailable and GPU features will be disabled. However, the Tesla GPUs **have a TCC driver**, that allows GPU access via Remote Desktop on some versions of Windows. If this driver is installed, the GPU is not used for the primary display; instead, the generic unaccelerated VGA driver is used. So, when sitting at the machine locally, window dragging will stutter, scrolling won't be smooth, etc.

XenApp 6.5 and XenDesktop allow remote users to access any GPU device, not just the Tesla models.

These technologies are still relatively new and we are not yet officially supporting them. However, they do seem to work.

- **I need something simple, right now. Is there a workaround?**

Yes. Use a VNC-based product. The VNC protocol operates at a higher level and does not interfere with the display drivers. It captures the final screen data which has changed, and transfers it from the remote machine. As such, it requires more bandwidth to maintain a smooth, interactive session. It is cross-platform and does not interfere with the GPU device drivers.

VNC is an open standard and there are many different clients and servers products. RealVNC, **has a free version**.

- **Are concurrent users supported?**

Not well. While the GPU driver will isolate multiple applications from each other, the GPU resources are still shared. For Tesla GPUs, the TCC driver offers the ability to put the GPU into an exclusive mode. This will prevent other applications from using the GPU. Also, refer to question [Can I run multiple instances of ADMIRE?](#)

4.5 Performance

- **Does the GPU functionality speed up all of ADMIRE?**

No. Algorithms designed for GPU execution are fundamentally different from their CPU counterparts. At this time, the core deformable registration algorithms have been accelerated. When GPU functionality is enabled, the following cases will show speed improvements:

- General
- Demons
- Head/Neck
- Prostate
- Brain (only for MR)
- Adaptive (GPU only)

- **If I run the exact same case with and without GPU support, will the results be the same?**

No. As mentioned above, the core algorithms have fundamentally different implementations on the CPU versus GPU. Therefore, results will not be identical.

- **I just replaced my GPU with a newer generation of GPU; will ADMIRE results be identical to those created by the earlier one?**

Not necessarily. Due to changes in hardware, parallelism, driver optimizations, results will not be identical. However, they should be within an acceptable margin. The reason for this discrepancy is analogous to how floating point calculations between different hardware platforms (e.g., PowerPC vs x86) and even compilers (Visual Studio 2003 vs 2008) can vary.

- **Why is my Tesla GPU reporting less than the full amount of memory?**

When ECC is enabled, a portion of the memory is reserved to store the ECC bits. The ECC bits are used to detect errors in the remaining memory. If this is problematic, ECC can be disabled.

- **Can I run multiple instances of ADMIRE?**

Yes, but only if one instance is using GPU functionality. In GPU computing, memory is currently the scarce resource. While multiple applications can use the GPU at the same time, if a memory request would exceed the remaining free GPU memory, that access will fail. The GPU does not swap out processes' memory as the computer's operating system does. Since portions of ADMIRE's GPU algorithms will use nearly all the available GPU memory, a second instance of ADMIRE at the same point of execution would get a memory allocation error. While it's certainly possible that two instances could execute in such a way that memory would not be exhausted, it's not something to rely upon.

- **Why did ADMIRE fail with a memory allocation error?**

If the GPU is the only video card in the system, part of the GPU memory is required for the primary display. Depending on the resolution being used, this amount may be large. For example, running at 1680x1050x24bit color will require at least $1680 \times 1050 \times 3$ bytes = 13MB for its frame buffer; there will usually be multiple frame buffers for the primary display. If multiple monitors are connected to the same GPU, the required memory is even more.

- **When ADMIRE is running, the Windows user interface isn't as responsive.**

If the GPU is also driving the primary display, this is normal behavior. When ADMIRE is using the GPU, CUDA temporarily gives the GPU calculations full control. During this time, screen updates are slowed. This is currently a limitation of CUDA and the display driver. If the GPU isn't driving the primary display (there is another video card in the computer), this doesn't occur.

- **Windows gives "Display driver stopped responding and has recovered" error and aborts the currently executing case.**

In order to recover from a driver crash or hardware error, Windows XP and higher employ a "watchdog timer" which monitors the display driver and restarts the driver service if it does not respond within a certain time limit. As stated in the previous question, the GPU is unresponsive while servicing a calculation. If a calculation exceeds the time limit, Windows aborts the call and restarts the driver. While ADMIRE does its best to manage its calculations so that any individual call does not exceed this limit, the wide variety of GPUs and patient data sizes almost guarantee this limit will be exceeded at some point.

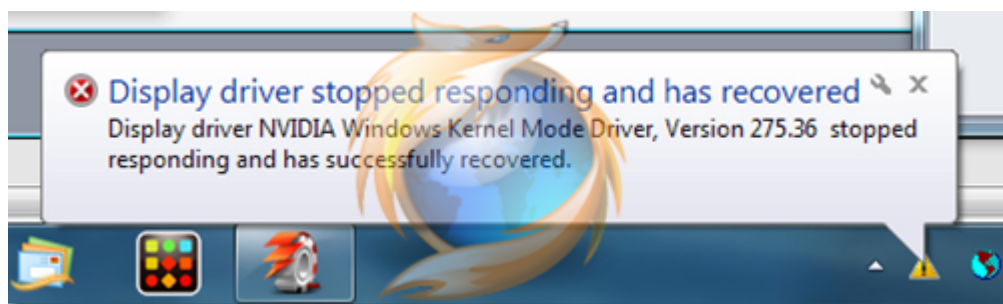


Figure 2 Watchdog timer error



It is possible to disable this timer. If the GPU is a Tesla model, the drivers provided by NVIDIA handle this automatically. On Windows XP, the timer doesn't apply if the GPU has no display attached. Unfortunately on Vista/7, the Quadro and GeForce drivers never disable the timer, even if the GPU has no display attached. As a workaround, there are registry settings which can be tweaked to increase/disable the watchdog timer. [Instructions for Vista/Windows7](#). [Instructions for Windows XP](#), [Alternate](#)

- **I upgraded from Windows XP to Windows 7, and now certain cases abort and Windows gives "Display stopped responding and has recovered" error. These cases ran successfully before the upgrade.**

On Windows XP, the watchdog timer is 5 seconds. On Windows Vista/7, the timer decreased to 2 seconds. See the previous question for more information.

- **I upgraded from Windows XP to Windows 7, and now certain cases abort with an "out of memory" error in the Console window.**

Windows Vista introduced the Aero GPU-accelerated desktop interface to draw and composite the desktop interface. However, depending on display size(s), this causes less memory to be available to CUDA applications. For example, driving two 1680x1050 displays from a single GPU reduces the available GPU memory by about 90MB.

To solve this problem, disable Aero: Control Panel -> System -> Advanced system settings -> Performance (Settings) -> Uncheck "Enable Aero peek" and "Enable desktop composition". Alternatively, install another GPU/video card to drive the display.

4.6 Deployment

- **Can I explicitly set the GPU to be used?**

Yes. If `CUDA_VISIBLE_DEVICES` environment variable is set, the CUDA runtime will ignore all devices not in this list. For example, if you have devices 0, 1, and 2, and you only want 1 and 2 to be available, set `CUDA_VISIBLE_DEVICES=1,2`. This is useful if you want to run two instances of ADMIRE and force each to use a separate GPU. Other aspects can also be [controlled via environment variables](#).

- **Can I limit the amount of GPU memory used?**

Although NVIDIA does not provide this functionality via `CUDA_*` environment variables, the ADMIRE application can approximate this functionality by pre-allocating memory. Set `CMSCUDA_MEMORY_GLOBAL_MAX` to the max number of bytes the application will use. This is usually only for debugging purposes.

- **I have a Quadro/Tesla GPU. How do I enable TCC mode so I can use Remote Desktop?**

First, verify you have the proper Quadro/Tesla driver installed. Then, from the commandline, find the `nvidia-smi.exe` program (typically, this installed into `C:\Program Files\NVIDIA Corporation\NVSMI`). From there, you can execute `nvidia-smi.exe -L` to see a list of devices. Then, `nvidia-smi.exe --id=[deviceid] --force-driver-model=1` to force a specific GPU into TCC mode. Rebooting will be required, and sometimes it takes a few tries to get it stick. You will need to be Administrator.

4.7 General GPU Computing

- **What is OpenCL and how does it relate to CUDA?**

GPU computing is a relatively new technology. As such, vendors support it in their own way. At this time, CUDA is most mature toolkit available. However, it is specific to NVIDIA products.

In December 2008, the first OpenCL (Open Computing Language) standard was published. It is a vendor-neutral API for GPU programming. It was created under the auspices of the Khronos Group, the same group responsible for OpenGL. As such, it shares a similar design philosophy. All major vendors (ATI, Intel, NVIDIA) were involved in its creation and plan to support it. At this time, the CUDA toolkit is the most mature GPU computing toolkit available. We are developing with a mind towards OpenCL, but it may be a number of years before the standard is supported well enough.

- **What about Intel's "Phi"?**

Phi is Intel's hybrid approach to parallel computing. Currently, programming GPUs requires very specific algorithm structure and low-level memory access pattern management to achieve peak performance. Instead, Phi uses "mostly x86" CPUs to provide a more traditional multithreading API that allows threads to be more complicated. We do not have any current plans to support Phi.

5 Image Spacing and Orientation

ADMIRE can handle images in arbitrary orientations with uniform or non-uniform image spacing.

5.1 Overview

ADMIRE can handle images in arbitrary orientations. Images are resampled into the standard DICOM patient-centric coordinate system if necessary. All calculations are done in physical space—each voxel has a physical size and the image volume has a physical orientation in space.

5.2 Image Spacing

Uniformly and non-uniformly spaced images are both supported. If the DICOM image is non-uniform, it will first be resampled to a uniform spacing value that matches the minimum distance between slices found. This is the distance between Image Position (Patient) (0020,0032) values (as opposed to Slice Thickness (0018,0050)). Trilinear interpolation is used.

5.2.1 Outlier images

If there are images on either end of the volume with a spacing greater than 10x the slice spacing, they will be discarded as outliers because interpolating such a great region of the image isn't useful.

5.2.2 Contour Generation

Upon loading, stacks of 2D contours are converted into a 3D surface mesh. In deformable registration, the transformation that is calculated between the two image volumes being registered is then applied to mesh. Finally, the object is sliced back into 2D contours before being written back out as DICOM RT Structure Sets.

Because the transformed surface is a mesh representation, the slicing locations can be arbitrary. A contour is only generated where there is intersection between the slicing plane and the surface mesh. **By convention, ADMIRE slices the mesh at the original image locations, respecting the orientation of the image plane.**

5.3 "Slightly" Oriented Images

TPSs typically verify contours match the image they are drawn upon using a variety of methods and assumptions. A typical HFS CT scan has Image Orientation (Patient) (0020, 0037) ("IOP") row and column direction cosine vectors of (1 0 0 0 1 0). Multiplying the row and column direction cosines gives a slice normal vector of (0 0 1). Combined into a rotation matrix, we can see it's identity.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This means the image slices are exactly transverse w.r.t. to the patient's inferior→superior axis. Looking at these IPP values, they share the same X and Y coordinate as expected...only the Z coordinate varies. Consider the following transverse, axially-aligned image with 1mm slice spacing:

Slice	X	Y	Z
1	-199.5829	-199.5833	-17.504
2	-199.5829	-199.5833	-16.504
...	
33	-199.5829	-199.5833	16.504
34	-199.5829	-199.5833	17.504

In this case, a system can verify contour/image plarity by simply comparing the Z coordinate of each contour point against the Z coordinate of the Image Position (Patient) with an appropriate floating point tolerance.

5.3.1 The Problem

Some MR scanners (i.e., early Elekta Unity MR Linac exam cards) can sometimes generate MRs that are *slightly* rotated, even though they appear (and likely were meant to be) to be standard transverse images. Inspecting the IOP tag, these images typically have a value that are very close to a standard orientation.

Suppose the direction cosines were instead (1, 0, 2.359e-05, 0, 1, 0). It results in a rotation (roughly 0.0014 degrees) matrix of:

$$\begin{bmatrix} 1 & 0 & 2.35934e-05 \\ 0 & 1 & 0 \\ -2.35934e-05 & 0 & 1 \end{bmatrix}$$

Note the slight drift in the X and Z coordinates:

Slice	X	Y	Z
1	-199.586855	-199.58332	-17.495291
2	-199.586832	-199.58332	-16.495291
3	-199.586808	-199.58332	-15.495291
...	
33	-199.582962	-199.58332	15.504708
34	-199.582939	-199.58332	16.504708
34	-199.582915	-199.58332	17.504708

This can cause compatibility issues later on because some systems will treat the above as "close enough" to (1 0 0 0 1 0) and treat the image as axial instead of slightly oriented. Furthermore, because contour points are encoded using floating point numbers, a tolerance value must be used when comparing contour point/image planarity. 0.0001mm is a common value—if the contour point is within this distance of the image plane, it's considered coplanar.

Because ADMIRE used the original orientation when slicing, the contours will have a slight orientation (i.e., the Z coordinate will vary slightly over the contour). However, the consuming TPS *thinks* the image is exactly axial, and is doing a straight floating point comparison of the contour and image Z coordinates. Depending on the tolerance values used and the distance of the contour from the image volume origin, the contour may, or *may not* fail coplanarity checks.

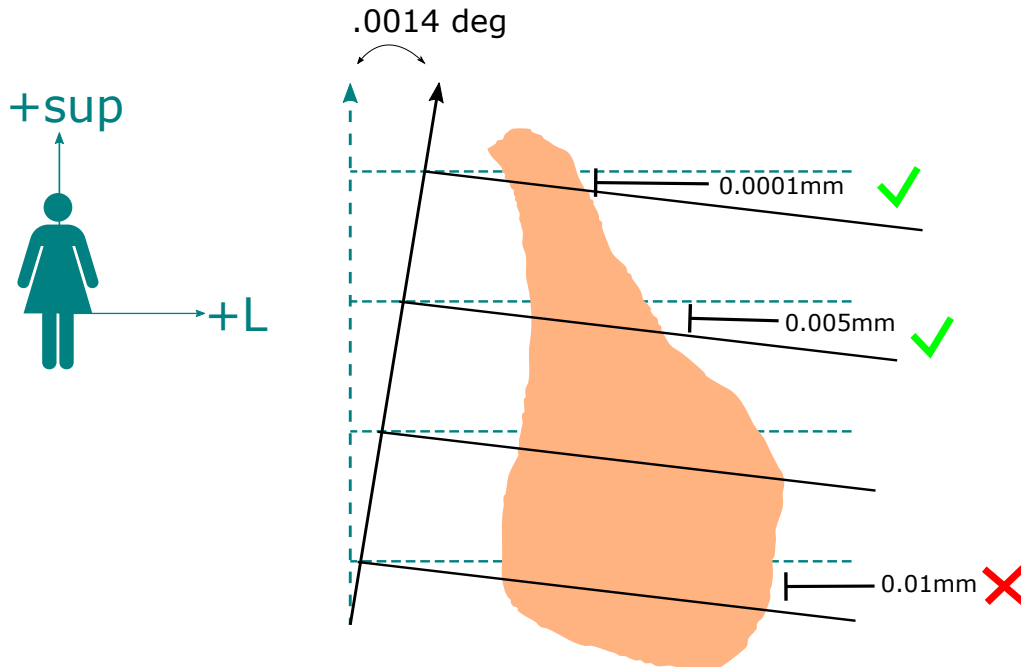


Figure 3 Only bottom contour fails coplanarity check with 0.01mm tolerance

5.3.2 A Solution: A Tale of Two Tolerances

ADMIRE's workaround (implemented in version 3.19) is to pre-emptively "snap" images to standard orientations. Upon reading, DICOM images' direction cosine values are rounded to -1, 0, or 1 if they fall within a certain tolerance. Then, the Image Position (Patient) values are rotated by this adjustment so the image volume remains orthogonal (i.e., a rectangular prism)—otherwise it would become skew, which almost no product supports. Since the values involved are well within the volume of a single voxel, there is no resampling required.

This tolerance is set via `ADMIRE.Core.UnitaryDirectionCosineTolerance` in `ADMIREParameters.json`. This affects all images loaded from DICOM. To disable this tolerance, set it to 0. Otherwise, the default value 0.0001, which corresponds to ~ 0.005 degrees, seems to work well with systems we've tested (ProKnow DS, Monaco).

The "correct" value of this tolerance is determined by the systems consuming ADMIRE's outputs. After all, the mathematically correct value of this parameter is 0—slicing contours exactly on the slightly oriented images. But this would require all TPSs to use the image's orientation as part of the contour planarity check and effectively mandates arbitrary orientation support. This is unlikely to happen when so many operations, algorithms, and optimizations assume transverse images.

5.4 Troubleshooting

A small command line utility, `OrientationCheck.exe` can be found in the ADMIRE installation folder (usually this is `C:\Program Files\ADMIRE`). When run against a directory of images (and optional structure sets) it will provide diagnostic information and consistency checks, such as:

- Do the direction cosines define an orthonormal coordinate system (i.e., all axes are perpendicular)?
 - if not, the image volume is skew... not rectangular
- Is the vector formed by the image positions consistent with the image plane normal?
- Would the direction cosines be corrected (as described above) per the `ADMIRE.Core.UnitaryDirectionCosineTolerance` value?
 - If so, what is the maximum contour point-to-image plane distance error that would occur within the image volume if the direction cosines were snapped, but the contour points left as-is.
- Max per-contour ΔX , ΔY , ΔZ for each ROI. E.g., most CTs, being transverse and axial, will have a $\Delta Z=0$, assuming direction cosines of (1 0 0 1 0)

```
PS C:\ProgramData\ADMIRE> C:\Program Files\ADMIRE\OrientationCheck.exe --src C:\mrskew
--contourPlanarityTolerance=0.0001
2020-06-17 12:03:03.302141: I
tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
cudart64_100.dll
Initializing dmctk logging system
Installing DCMtkLoggingAdapter; prefiltering DCMtk to DEBUG
Installing ITK logging adapter and enabling GlobalWarningDisplay
Installing VTK logging adapter and enabling GlobalWarningDisplay
OrientationCheck 3.19 Copyright (c) 2008 - 2020 Elekta, Inc. For Investigational Use Only
Contact Daniel.Thill@elekta.com for help
-----
Loading parameters file: ADMIREParameters.json
Parsing "ADMIRE.Core.UnitaryDirectionCosineTolerance=0.0" as parameter string
Searching for atlases in (C:\mrskew)
Found image series ref by series uid (1.2.826.0.1.63521.111890309932130250155909667039895413195)
Found image series ref by frame of ref uid (1.2.826.0.1.63521.86285679508537193957816764238997272690)
-----
Image Orientation (Patient): (1,0,-2.35934e-05,0,1,0)
1 0 2.35934e-05
0 1 -0
-2.35934e-05 0 1
Rotation angles w.r.t. X (patient L->R): [0, 90, 90.0014]
Rotation angles w.r.t. Y (ant->pos) : [90, 0, 90]
Rotation angles w.r.t. Z (inf->sup) : [89.9986, 90, 0]
-----
IOPx : 1 0 -2.35934e-05
IOPy : 0 1 0
IOPz : 2.35934e-05 -0 1
|IOPx x IOPz| (should be 1): 1
|IOPy x IOPz| (should be 1): 1
^ If both above are 1, then IOP forms a valid, orthonormal direction cosine rotation matrix
-----
First slice: -199.587 -199.583 -149.495
Last slice: -199.58 -199.583 149.505
IPP vector (last slice - first slice): 2.35934e-05 0 1
Are IPPs consistent with IOP direction cosines?
IPP Normal X IOPz (should be close to 0 0 0): 0 -6.62746e-11 -0
|IPP Normal X IOPz| (should be 0): 6.62746e-11
-----
Image direction cosines (1,0,-2.35934e-05,0,1,0) fell within default tolerance parameter ADMIRE
.Core.UnitaryDirectionCosineTolerance=0.0001 (~0.00572958 degrees) and would be rounded to
unitary values (1,0,0,0,1,0). This aids compatibility with TPSs handling Elekta Unity MR
data that is "almost" axial because they often compare the contour Z value directly against
the slice Image Position (Patient) Z coord. However, this rounding may cause problems in other
TPSs that handle oriented images completely and correctly. Within this image volume, using
these rounded cosines would result in a max contour point-to-image plane
difference of -2.00004e-06 -2e-06 -1.49505e-06 (mm) compared to slicing at
the original direction cosines exactly.
-----
Checking all contours for out-of-plane points within tolerance 0.0001 mm....
```

```

Returning default mapper
Loaded RTSS: C:/mrskew/PRSTSBRT02_StrctrSets.dcm
ROI 0 (CTV), contour 0, pt 32 is out of plane ( |pt . IOPz| = -0.000110889): 10.5 29.6 -20.4953
ROI 0 (CTV), contour 0, pt 33 is out of plane ( |pt . IOPz| = -0.000117967): 10.2 29.6 -20.4953
ROI 0 (CTV), contour 0, pt 34 is out of plane ( |pt . IOPz| = -0.000127404): 9.8 29.6 -20.4953
ROI 0 (CTV), contour 0, pt 35 is out of plane ( |pt . IOPz| = -0.000160435): 8.4 29.6 -20.4953
ROI 0 (CTV), contour 0, pt 36 is out of plane ( |pt . IOPz| = -0.000186388): 7.3 29.6 -20.4953
... plus 12647 additional points
ROI 0 (CTV) max contour delta_XYZ: (68.9, 61.8, 0)mm
ROI 1 (PTV), contour 0, pt 1 is out of plane ( |pt . IOPz| = 0.000115607): 12.1 9.2 -25.4953
ROI 1 (PTV), contour 0, pt 2 is out of plane ( |pt . IOPz| = 0.000162794): 14.1 11.8 -25.4953
ROI 1 (PTV), contour 0, pt 3 is out of plane ( |pt . IOPz| = 0.000209981): 16.1 14.8 -25.4953
ROI 1 (PTV), contour 0, pt 4 is out of plane ( |pt . IOPz| = 0.000200544): 15.7 26.7 -25.4953
ROI 1 (PTV), contour 0, pt 5 is out of plane ( |pt . IOPz| = 0.000153357): 13.7 29.7 -25.4953
... plus 2493 additional points
ROI 1 (PTV) max contour delta_XYZ: (68.5, 74.5, 0)mm
ROI 2 (PTV_OVL_retto), contour 0, pt 2 is out of plane ( |pt . IOPz| = -0.000221778): 4.7 31.2 -24.4953
ROI 2 (PTV_OVL_retto), contour 0, pt 3 is out of plane ( |pt . IOPz| = -0.00024773): 3.6 31.5 -24.4953
ROI 2 (PTV_OVL_retto), contour 0, pt 4 is out of plane ( |pt . IOPz| = -0.000245371): 3.7 31.7 -24.4953
ROI 2 (PTV_OVL_retto), contour 1, pt 3 is out of plane ( |pt . IOPz| = -0.000280761): 3.7 31.2 -23.4953
ROI 2 (PTV_OVL_retto), contour 1, pt 4 is out of plane ( |pt . IOPz| = -0.000337385): 1.3 31.8 -23.4953
... plus 209 additional points
ROI 2 (PTV_OVL_retto) max contour delta_XYZ: (23.7, 7.5, 0)mm

```

6 Segmenter Detail

The following table describes the how different modalities and regions require different approaches and algorithms. The first column denotes phases/features of the process and is ordered roughly chronologically.

6.1 Inter-Patient Deformable Registration

	General>			Head/↔ Neck>		Prostate>	
Modality (Atlas↔ Patient)	CT-CT	CT-CT (GPU)	MR-MR (GPU)	CT-CT	CT-CT (GPU)	CT-CT	CT-CT (GPU)
Linear registration metric (12-dof)	SSD	SSD	MI	SSD	SSD	SSD	SSD
Image correction				Mandible	Head pose (Mandible,Brain Stem)	Rectal Gas, Bladder filling	Bladder filling
Coarse DR	MI	MI	MI	MI	MI	MI	MI
Dense DR	Demons LCC	Hybrid MI-NMSD	Hybrid MI-NMSD	Demons LCC	Hybrid MI-NMSD	Demons LCC	Hybrid MI-NMSD
ROI refine- ments				Parotid+↔ SMG, Mandible	Parotid+↔ SMG, Mandible	Pelvis+↔ Femur, Bladder, Prostate, Rectum	Pelvis+↔ Femur, Bladder, Prostate, Rectum
Skin refinement	*	*		*	*	*	*

6.2 Intra-Patient / Adaptive Deformable Registration

Feature							
Modalities (Atlas-Patient)	CT-CT	CT-CBCT	CT-MR	MR-CT	MR-MR	MR-MR (mixed sequence)	Lung
		CBCT-CT					CT-CT
		CBCT-CBCT					CBCT-CBCT
Rectum gas removal	*						
Rigid registration metric (6-dof)	LCC	LCC	LCC	LCC	LCC	LCC	LCC
Coarse DR	NMSD(SAD)	NMSD			NMSD		NMSD(SAD)
Dense DR	LCC	LCC	NMI	NMI	LCC	NMI	NMSD(SAD)
ROI constrained DR support	*	*	*	*	*	*	*
Inverse-consistent			*	*		*	
Pelvic area refinement	*						
Skin refinement	*	If CT patient		*			*

6.3 Deep Learning Autosegmentation

See [Deep Learning Autosegmentation](#) for detailed information.

Note

GPU is required for all intra-patient functionality

Currently ROI constrained DR support is mutually exclusive with pelvic area refinement

Mixed-sequence refers to registering anatomical (e.g., T1 and T2) and non-anatomical sequence types

6.4 Glossary

- **BM**: Block-matching
- **DR**: Deformable Registration
- **LCC**: Local Cross-Correlation
- **MI**: Mutual Information
- **NMI**: Normalized Mutual Information
- **NMSD**: Normalized Mean Squared Difference
- **SAD**: Sum of Absolute Differences
- **SSD**: Sum of Squared Differences

7 Structure Mappings

Explains the purpose and behavior of mapping structure names.

7.1 Motivation

Due to differences in structure naming conventions among institutions, ADMIRE uses a mappings file to convert the structure name as it appears in DICOM to a *semantic* name. This allows ADMIRE to consistently perform structure-specific processing. The mappings are stored in a simple comma-separated value (CSV) file. Each line represents one structure in the format of *semantic name, name 1, name 2, name 3, [etc]*. The following is an excerpt from a mapping file:

```
BLADDER
BRAIN
BRAIN_STEM
CHIASMA
ESOPHAGUS, Oesophagus
EYE_LT, eye_l, eye_lt, eye l.
EYE_RT, eye_r, eye_rt, eye r.
FEMUR_LT, lt femur
FEMUR_RT, rt femur
SEM_VES, seminal_vesicles
SKIN, patient, exterior, external, body, ext
SPINAL_CORD, Cord
SUBMAND_LT
```

Values are case-insensitive and leading and trailing whitespaces are stripped from each value. If the structure name already matches the semantic name, there's no need to include a mapping for it. For example, if institution conventions dictate "bladder", it will automatically map to the BLADDER semantic name.

7.2 Mapping Process

The mappings are applied as the first step of any segmentation/registration process. During calculations, logging messages in the console will usually refer to the semantic name. When a new structure set is being generated, the semantic name is mapped back to the original name used in the atlas. For multi-atlas segmentations (i.e., STAPLE, patch fusion), the semantic names are used to group the intermediate segmentations from each atlas.

7.3 Important Mappings

While any mapping can be added to the mappings file to aid matching structures for multi-atlas processing, ADMIRE will take advantage of the following specific structures to improve segmentation results.

7.3.1 All case types with CT patient (except Demons)

- **SKIN**

If present, the SKIN is generated using image processing techniques (thresholding, etc) rather than deformable registration; this almost always gives smoother, better results. SKIN contours are also used as a bounding box to improve segmentation speed by cropping the atlas image volumes.

7.3.2 Head/Neck cases

- **MANDIBLE**

If present, physical model-based refinement is performed during the coarse and final deformable registration phases

- **PAROTID_LT, PAROTID_RT, SUBMAND_LT, SUBMAND_RT**

If all are present, refinement is performed after final deformable registration.

- **SKIN, MANDIBLE, SPINAL_CORD (optionally PAROTID_LT, PAROTID_RT, SUBMAND_LT, SUBMAND_RT, EYE_LT, EYE_RT, LENS_LT, LENS_RT, HYOID, THYROID, LARYNX, MASSETER_LT, MASSETER_RT, PTERY_LAT_LT, PTERY_LAT_RT, PTERY_MED_LT, PTERY_MED_RT)**

If first three are present, all available are used to constrain the coarse block-matching deformable registration phases

- **SKIN, MANDIBLE, SPINAL_CORD, BRAIN_STEM**

If all are present, they are used to constrain the dense deformable registration phases

- **MANDIBLE, BRAIN_STEM**

If both present, head-pose correction is performed on atlas and subject image data during pre-processing

7.3.3 Prostate cases

- **BLADDER, PROSTATE, RECTUM**

If all three are present, these structures are given physical model-based refinement after deformable registration.

- **FEMUR (or FEMUR_LT and FEMUR_RT), PELVIS (or PELVIS_RT and PELVIS_LT)**

If all are present, these structures are given extra refinement after deformable registration.

7.3.4 Inpatient cases

- **RECTUM**

If present, preprocessing is performed on atlas and subject images to correct for rectal gas

- **BLADDER, PROSTATE (or UTERUS)**

If both are present, model-based bladder refinement is performed as part of the dense DR phase.

- **LUNGS (or LUNG_LT and LUNG_RT)**

Feature-constrained lung segmentation can use lung structures to define bounding boxes; if both LUNG_LT and LUNG_RT are provided, they are merged. If none are provided, autodetection is attempted.

- **SKIN** If present (and image is CT, CBCT), the SKIN is generated using image processing techniques (thresholding, etc) rather than deformable registration; this almost always gives smoother, better results.

7.3.5 Patch Fusion, RF Label Fusion

- **BRACHIALPLEXS, BREASTS, KIDNEYS, LUNGS, NVB, PELVIS, RIBS, RETINAS, TESTIS**

Multilabel fusion algorithms typically operates on the largest connected component to reduce artifacts and noise. However, in some cases, a single ROI is segmented into two separate connected components. For example, if the vertebrae aren't part of the RIBS ROI and all RIBS components are operated upon to avoid discarding half the ribcage.

7.4 Mappings File Location

The mappings file, `mappings.txt`, is located alongside other program data (config file, log file, etc) in the current working directory. In a normal installation, this is the common application data folder. On Windows 7 systems, this is typically `C:\ProgramData`. On Windows XP, it would be `C:\Documents and Settings\All Users`. You can also check the `ALLUSERSPROFILE` environment variable for the path.

8 Deformation Vector Field File Guide

Explain DVF format and usage.

The output deformation field from ADMIRE can be stored as a vector-valued 3D image in the ITK MetaImage format (.mhd). Detailed explanation of this image file format can be found at and <http://www.itk.org/Wiki/MetaIO>. It can also be stored as a Deformable Spatial Registration Object as defined in the [DICOM 2009 standard](#).

We start with the definition of an ITK image. We briefly introduce this file format in the following and explain how to interpret and properly use the saved deformation field in practice.

8.1 ITK Image and the MHD File Format

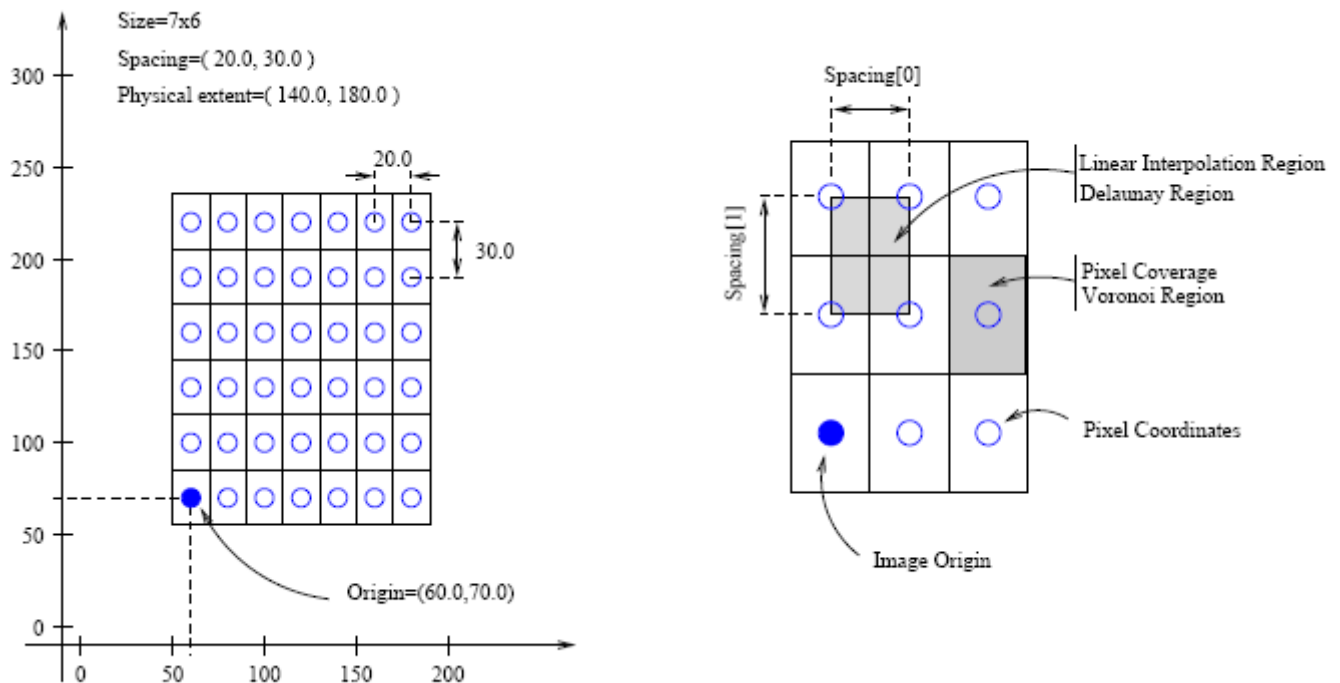


Figure 4 Geometric concepts associated with an ITK image

Fig. 1 is taken from the [ITK software guide](#), which illustrates all the information needed to uniquely define an image (assuming the image is not oriented). In particular, the information associated with the physical spacing between image pixels and the position of the image with respect to some world coordinate system are extremely important.

In Fig. 1, circles are used to represent the center of pixels (voxels for a 3D image). The value of a pixel is assumed to exist as a Dirac Delta function located at the pixel center. Pixel spacing is measured between the pixel centers and can be different along each dimension. The image origin is associated with the coordinates of the first pixel in the image.

The image pixel value can be a single number (e.g., a gray-scale image) or a vector of numbers (e.g., RGB value of a color image, or displacement vector of a deformation field image). The former type can be referred to as a scalar image, and the latter a vector image.

Several different image file formats are supported by ITK, as listed at [http://www.itk.org/Wiki/ITK_↵File_Formats](http://www.itk.org/Wiki/ITK_File_Formats). The MetaImage format is relatively simple but sufficient to store the output deformation field.

A deformation field stored in the MetaImage format actually consists of two separate files, one with a ".mhd" extension and one with a ".raw" extension. The ".mhd" file is the header file and the ".raw" file stores the actual deformation field data. The ".mhd" file is also a simple ASCII file and its content can be easily visualized using any text editor. The following list shows the typical contents of an ADMIRE deformation field .mhd file (the corresponding ".raw" file is assumed to be "deffield.raw"):

```
ObjectType = Image
NDims = 3
BinaryData = True
BinaryDataByteOrderMSB = False
CompressedData = False
TransformMatrix = 1 0 0 0 1 0 0 0 1
Offset = -120 -94.5 -119.063
CenterOfRotation = 0 0 0
AnatomicalOrientation = RAI
ElementSpacing = 0.9375 1.5 0.9375
DimSize = 256 128 256
ElementNumberOfChannels = 3
ElementType = MET_FLOAT
ElementDataFile = deffield.raw
```

Explanation of the tags used can be found at <http://www.itk.org/Wiki/MetaIO/Documentation>. We explain some important ones in the following: "BinaryDataByteOrderMSB = False" indicates that the data is stored in little-endian bit-ordering. If this value is "True", then the image data is stored in big-endian bit-ordering. "↵TransformationMatrix" stores the directional-cosines or the orientation matrix of the image. Here the transformation matrix is the identity matrix, which indicates that the image is non-oriented or non-tilted. "Offset" specifies the origin of the image in the underlying physical coordinate system as illustrated in Fig. 1. "ElementSpacing" specifies the image spacing. "DimSize (= XN YN ZN)" indicates the image size in each of the coordinate directions. "ElementNumberOf↵Channels = 3" indicates that the deformation field image is a vector image and each voxel value is a three-dimensional vector. "ElementType = MET_FLOAT" indicates that each component of the image data is stored as a 32-bit floating number. "ElementDataFile = deffield.raw" indicates that the image (deformation field) data is stored in the "deffield.raw" file.

ITK is not required to read the MetaImage format (though it is the easiest). After parsing the extents from the .mhd, the raw floating point data can then be read using standard C file I/O. The binary data is in the same order as the itk::Image image data: data[slice][row][col], where each voxel is 3-elements. For example, if a DVF has "DimSize = 128 256 64", it would map into a C array defined as `float dvf[64][256][128][3]`.

8.2 DICOM Format

DICOM PS 3.3-2009 defines a Deformable Spatial Registration IOD for representing a spatial relationship between two frames of reference using deformation fields (the standard uses the term "grid") and affine transformation matrices. The transformation maps data from a *Source* frame of reference into the *Registered* frame of reference. As with the MetaImage format, the transformation is purely spatial—it is independent of the data being transformed. In general, this IOD contains the following items:

- The deformation field values stored as a binary blob of floating point values
- Metadata about the dimensions, origin, spacing, etc
- References for the images in the Source and Registered image sets
- Additional pre- and post- affine transformations
- Other patient and equipment information

The specifics of reading DICOM data files is beyond the scope of this document. There are many DICOM toolkits and applications available; ADMIRE DICOM functionality is built upon the [DCMTK toolkit](#), which has free libraries and commandline tools. All of the data that is present in the ITK MetaImage format will be found in the DICOM file represented in a similar fashion. In fact, the binary floating point arrays are identical. The following table documents the most relevant tags and any notes specific to ADMIRE's usage of them.

Attribute Name	Tag	Type	Description
Deformable Registration Sequence	(0064,0002)	1	A sequence containing 1 registration
> Source Frame of Reference UID	(0064,0003)	1	The source FOR (typically the atlas)
> Referenced Image Sequence	(0008,1140)	1C	Identifies the set of images in the Source FOR. There will be item for each image, each consisting of a Class UID and Instance UID.
> Registration Type Code Sequence	(0070,030D)	2	Describes the method used for the registration process. It will contain a single ICD-9 code (Value: 125024, Scheme Designator: DCM, Meaning: "Image Content-based Alignment")
> Pre Deformation Matrix Registration Sequence	(0064,000F)	1C	A sequence that specifies one spatial registration to be applied prior to the deformation. ADMIRE does not use this tag as the linear registration is composed into a total deformation field
> Post Deformation Matrix Registration Sequence	(0064,0010)	1C	A sequence that specifies one spatial registration to be applied after the deformation. ADMIRE does not use this tag.
> Deformable Registration Grid Sequence	(0064,0005)	1C	Describes the deformation grid used to sample into the Source RCS.
>> Image Orientation (Patient)	(0020,0037)	1	The direction of cosines of the first row and first column of the Vector Grid Data (0064,0009) with respect to the patient. E.g. "1\0\0\0\1\0"

Attribute Name	Tag	Type	Description
>> Image Position (Patient)	(0020,0032)	1	The x, y, and z coordinates of the upper left hand voxel (center of the first voxel transmitted) of the grid, in mm in the Registered Frame of Reference. This is identical to the Metalmage. A string triple, e.g.: "-120\94.5\119.063".
>> Grid Dimensions	(0064,0007)	1	The dimensions of the grid, in voxels. A string triple representing the number of voxels along the X, Y, and Z axes. I.e. "256\128\256".
>> Grid Resolution	(0064,0008)	1	The resolution of the grid voxels. A triple representing the size of a deformation voxel in along the X, Y, and Z dimension, in mm. I.e., "0.9375\1.5\0.9375"
>> Vector Grid Data	(0064,0009)	1	A data stream of vectors. The value is encoded as a string of 32-bit IEEE 754:1985 floating point words. The vectors are ordered in exactly the same order as in a itk::Metalmage (i.e., <code>data[slice][row][col][3]</code>). Endianness is defined as part of the DICOM file header and is typically handled by the toolkit. The total size in bytes will be <code>slices*rows*cols*3*4</code> .

8.3 What Is the ADMIRE Deformation Field?

The output deformation field stores the result of deformable image registration. As illustrated in Fig. 2 (also taken from the ITK software guide), image registration is the process of determining the spatial transform that maps points from one image (called the fixed, atlas, or target image) to the corresponding points in the second image (called the moving, patient, or floating image). In the case of a rigid or an affine image registration, every image point of the fixed image follows the same linear transformation, and hence the whole image transform can be represented using a single linear transformation matrix. However, when "deformable" image registration is applied, each fixed image point has its own transformation. To represent the result of such a deformable registration, it is conventional to store at each fixed image location its displacement vector (e.g., vector $\vec{pq} = (\vec{p} - \vec{q})$) for point p in Fig. 2). Consequently, the spatial transform for the whole fixed or atlas image is represented by an array of displacement vectors, which is usually referred to as the displacement field or the deformation field. This deformation field is then stored in a pair of .mhd and .raw files as mentioned above.

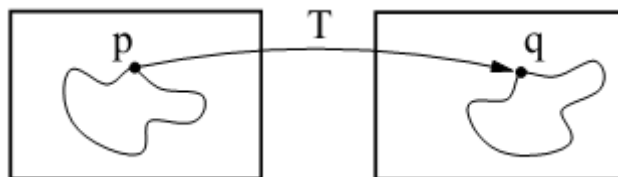


Figure 5 Image registration is the task of finding a spatial transform mapping points from one image into another



"Fig2: Image registration is the task of finding a spatial transform mapping points from one image into another"

ADMIRE performs the image registration in all three dimensions, hence each voxel value of the saved deformation field image is a three-dimensional vector. The first component of the vector is the displacement in the x-direction, the second is the displacement in the y-direction, and the third is the z-direction. It should be emphasized that ADMIRE performs the deformable registration in the physical space specified by the DICOM images. As a result, the displacement vectors all have a physical unit associated with them, which is the same as the unit (typically mm) used for specifying the origin and spacing information in the DICOM images.

It should also be noted that ADMIRE performs a linear registration (this is usually a 4x4 affine transformation, but varies depending on segmentation type) first before the deformable registration, and the saved deformation field actually represents the total composed transform. For example, if the atlas image and the subject image have a large offset in the x-coordinate direction, then the deformation field image will have a very large value for the x-component of the displacement vector at each voxel location.

8.4 Using the deformation field

The deformation field establishes a correspondence between points of two images (in the physical space). Thus, it can be used to map structures or data associated with one image into the space of the other image. For example, suppose that the atlas image has some manually drawn structure contours. Once the deformation field is computed, the displacement vector at each atlas contour point can be computed (through interpolation if necessary) and added to the point's physical coordinates. The points with the new coordinates then define a corresponding contour in the patient image for the same structure. This process is illustrated in Fig. 2 above.

There are, however, a few issues one must keep in mind in order to get the right deformation field and to apply it properly. First of all, the deformation field is *unidirectional*. In other words, the deformation field only stores the displacements for points in the fixed image (or the atlas image for ADMIRE), but not for the moving image (the patient image for ADMIRE). Thus, it cannot be used to map contours from the moving image back to the fixed image. To map contours from the moving image to the fixed image, the *inverse* deformation field will be needed. Unfortunately, unlike a linear transform, the inverse deformation field is in general not easy to compute. ADMIRE can compute this inverse deformation field iteratively but it currently requires GPU acceleration. As a result, one may need to re-run ADMIRE a second time to compute the deformation field in the opposite direction. In practice, one should choose carefully which image to use as the "atlas" and which one as the "patient" based on the direction of the deformation field that is needed.

Second, one should note that the desired direction of the deformation field may be different for the purpose of mapping structured data (like a dose map) than for the mapping of contours as mentioned above. This difference can be explained with the help of the following figure.

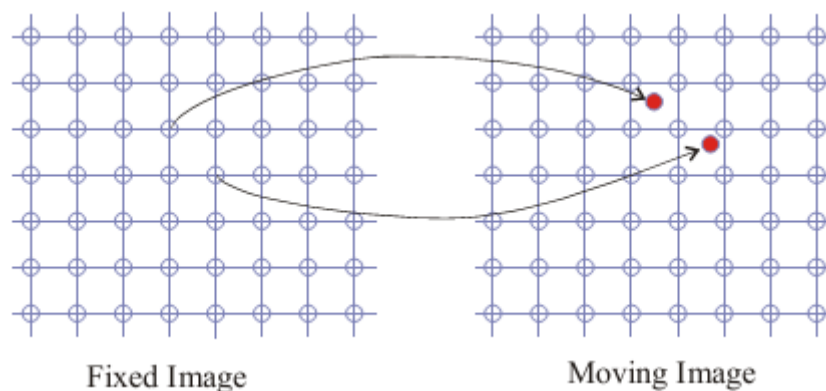


Figure 6 grid positions of the fixed image are mapped to non-grid positions of the moving image



"Fig2: Image registration is the task of finding a spatial transform mapping points from one image into another"

Suppose that some structured data (such as a dose map) is defined in the physical space of the moving image. To map the data to the physical space of the fixed image and sample it on a regular fixed image grid (for the purpose of storage using an image array), one can simply go through every pixel of the fixed image grid, compute its physical coordinates, add the displacement vector to map the point to the moving image space, and then find the data value at this mapped point (through interpolation if needed). On the other hand, if the original data is defined on a regular grid in the fixed image space, the data mapped to the moving image space will in general fall on a set of irregularly spaced points as illustrated by the red dots in Fig. 3. Resampling this set of irregularly-spaced data-samples to a regular moving image grid is then a much harder problem.

Third, it should be emphasized that image registration is performed in the physical space (or the world coordinate system), which is consistently with how image data are used in treatment planning. Thus, the output deformation field may *not* match the atlas image in a voxel-by-voxel sense, especially if the original atlas DICOM image has a non-uniform slice spacing or is tilted (oriented). As one example, suppose that the atlas DICOM image has a non-uniform spacing of 3 mm in middle slices and 5 mm in both ends, then the saved deformation field will have more slices than the original DICOM data. This is because ADMIRE internally re-samples the original DICOM image into a uniform spacing of 3 mm before performing image registration. Thus, accessing the displacement value at an atlas image point should be done through the point's physical location (i.e., its physical coordinates), and by finding the deformation field value at the same physical location in the deformation field image. If this physical location does not exactly match any voxel center of the deformation field image, interpolation will be needed to get the value of the displacement at this non-voxel location. Tri-linear interpolation is typically sufficient for this purpose.

8.5 Post-Processing

Unless the intra-patient segmentation is performed, the **DVF does not represent the changes made during region-specific refinement** (e.g., Head/Neck or Prostate). The refinement phase operates on the contours directly—it does not use the atlas image data. We are researching ways to update the DVF.

8.6 Dimensions

The DVF will represent the volume of the atlas after it has been cropped (couch removal, empty space, etc) during preprocessing. All the various segmentation types use a multi-resolution approach and different heuristics to determine the resampling factors for each resolution level. It's common for the final level of the registration to result in a DVF that is sampled at one-half resolution in one or more dimensions. This is because at some point, operating at the original voxel spacing doesn't give good enough improvement for the much higher computational cost. Therefore, the physical volume extents will almost always be smaller than the original atlas image volume and sampled at a coarser resolution. It is up to the user to pad the DVF to a larger size if necessary.

8.7 Inverse Fields

As mentioned above, there are instances where the inverse DVF is useful. ADMIRE can calculate this field using an iterative technique implemented on the GPU. The inverse DVF is required for ADMIRE's [Patch Fusion](#) implementation. After inversion, the resulting field is sampled at the original patient voxel locations—empty space and all. Unlike the original DVF, it will have the exact same resolution and size of the original image volume it is based upon. Therefore, it will usually be larger (either bytes-wise or dimensions-wise) than the non-inverse DVF. In the future, we look to reduce the exported inverted DVF to just the relevant voxel data to avoid storing all the zero-vectors of the air space.

8.8 Summary

- DVFs are defined as a regular grid on the atlas (fixed) image
- DVFs will usually be smaller and coarser resolution than the original image volume
- Inverse DVFs are defined as a regular grid on the patient (moving) image
- Pay special attention to which DVF is required
- DVFs do not represent structure refinements that occur post-registration (except for intrapatient segmentation)
- Exported DICOM DVFs are large, on the order of 100MB to 300MB depending on volume size.
- DVFs represent the entire transformation; they are composed with the linear registration.

9 Deep Learning Autosegmentation

ADMIRE 2.5 was Elekta's first release that contains deep learning functionality. It is currently implemented using [Tensorflow](#). As such, CUDA software and hardware requirements are much higher than previous versions of ADMIRE. Unfortunately, this requires us to drop support for older GPUs, as NVIDIA has stopped supporting them.

9.1 System Requirements

- 64-bit CPU with [AVX instruction support](#). The installation process will check, and if not found, a message will appear. Installation can be completed, but the application will likely crash during execution with an 0xc0000142 or 0xc000001d exception.

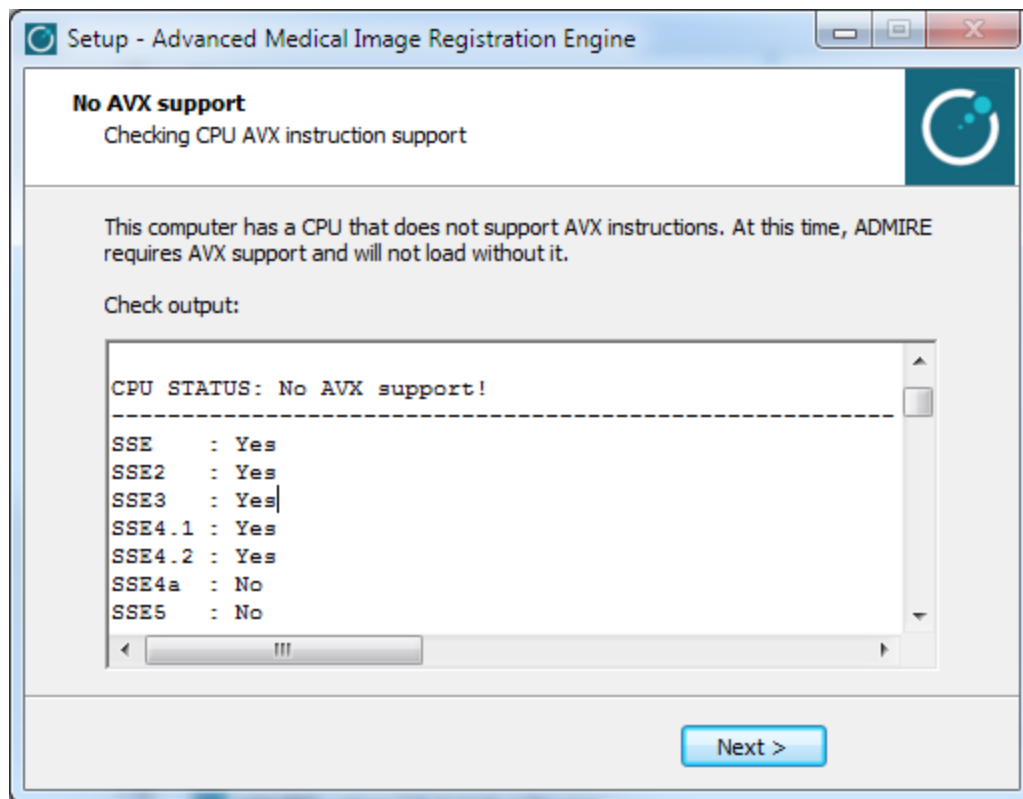


Figure 7 Missing AVX



- Windows 7 or later
- .NET 4.5 runtime
- 8 GB RAM
- NVIDIA CUDA hardware version 3.0 (Kepler) or greater (e.g., Tesla K20c) with ≥ 4 GB memory
- NVIDIA display driver which supports CUDA v10.0
- A license for the specific deep learning model(s)

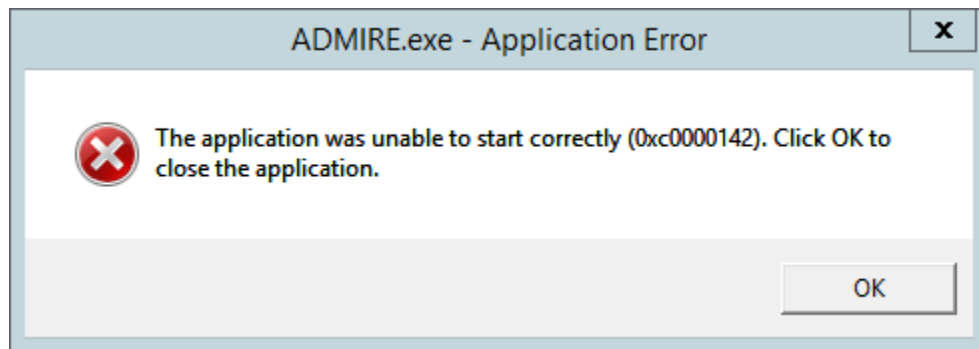


Figure 8 ADMIRE startup error with no AVX



9.2 Overview

Unlike deformable registration, deep learning segments images directly without the use of a pre-existing segmentation (i.e., atlas). The result of a DL algorithm is a labeled image where each voxel is assigned to a structure. There does not exist any transformation, linear, or otherwise.

9.2.1 Limitations

Because DL autosegmentation is not a form of registration, any functionality that uses a transformation will be silently skipped if it is enabled in the Configuration section:

- Export of any DVF, spatial, or deformable spatial registration objects
- Multi-atlas fusion (STAPLE, patch fusion, random forest)
- Warped atlas/subject images or doses

9.2.2 GUI

ADMIRE's user interface and underlying workflow engine has always been designed around registering pairs of image sets: an *atlas* and *subject*. Therefore, for DL functionality, pseudo-atlases are created to represent the specific DL models. The Tensorflow weights file *modelname.pb* contains the graph structures and weights; *modelname.pb.json* is an ADMIRE-specific file that describes the model, its inputs, and label-specific parameters. The GUI will read this JSON file and synthesize a fake DICOM RTSS and image set on the fly. This allows the autoseg model to substitute as a DICOM atlas throughout the GUI and workflow.

Unwanted structures can be deselected like normal atlases; they will not appear in the final results.

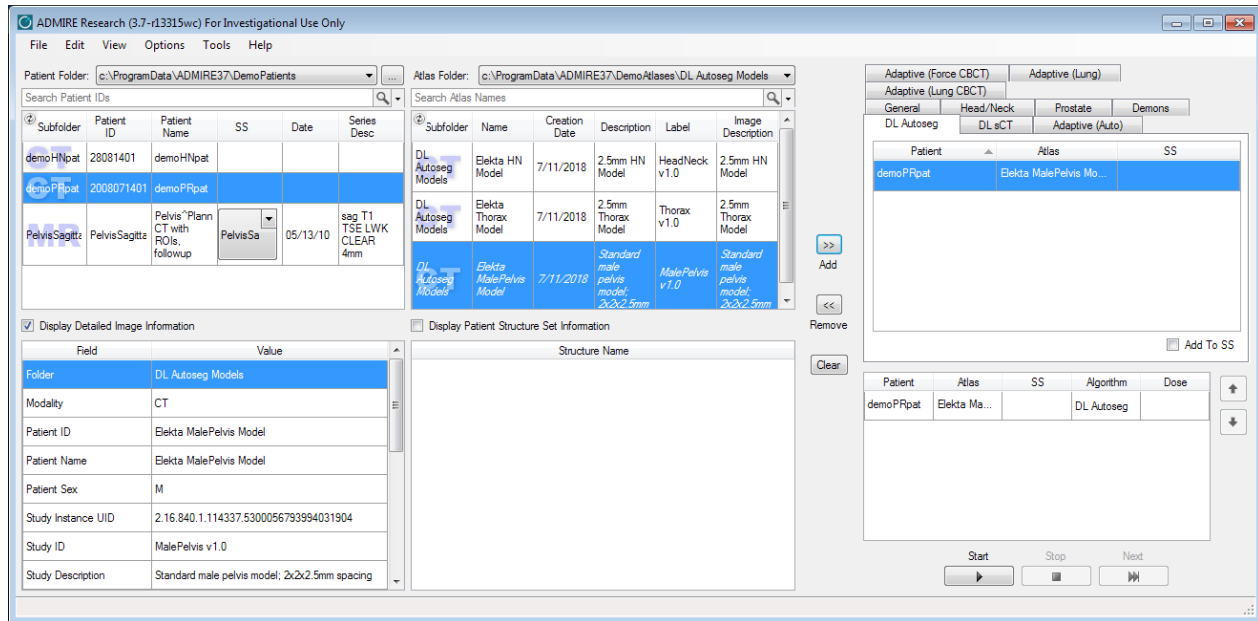


Figure 9 Male Pelvis DL



9.2.3 Command-line

As with the GUI, the command-line usage requires specification of one of the DL psueudo-atlases. The DL algorithm is selected by choosing the model:

```
--type dl_autoseg --atlas "c:\ProgramData\ADMIRE\DemoAtlases\DL Autoseg Models\DCNN_ProstateModel_final.pb.json"
```

The Powershell equivalent to the above GUI case would be:

```
C:\> cd c:\programdata\admire
C:\ProgramData\ADMIRE> &"c:\program files\ADMIRE\admire.exe" --type dl_autoseg `
--atlas "c:\programdata\ADMIRE\Demo Atlases\DL Autoseg Models\DCNN_ProstateModel_final.pb.json" `
--patient "c:\programdata\admire\DemoPatients\DemoPRpat" `
--results "c:\results" --gpu
```

Note

Will exit with error code 2 if not licensed for the specific model

9.3 Current Models

Note

All MR models were trained on Elekta Unity T2 (2min acquisition) images. They do not work well on other MR sequence types.

9.3.1 Autosegmentation

↓ Structure / Model →	HeadNeck CT	Thorax CT	MalePelvis CT	MalePelvis MR	Brachy Breast CT
Bladder			*	*	
Bones					*
Brain_Stem	*				
Esophagus		*			
External	*	*	*	*	*
Femur_Joint_L,R				*	
Femur_L,R			*		
Heart		*			
Lung_L,R		*			
Lungs					*
Mandible	*				
Parotid_L,R	*				
Pelvis L,R			*		
Penilebulb				*	
Prostate			*	*	
Pubic_Symphysis				*	
Rectum			*	*	
Seminalves				*	
Spinal_Cord	*	*			
Submand_L,R	*				

9.3.2 Synthetic CT

Creation of synthetic CTs images from a different modality. Different models are trained for different anatomical regions. Models exist for each configuration below.

↓ Input Modality / Region →	MalePelvis	Thorax	HeadNeck
CBCT	*	*	*
MR	*		

9.4 Frequently Asked Questions

- **What network structure is used?**

3D ResUNet. Please refer to [Automatic Image Segmentation using Deep Convolutional Neural Networks](#) in the Whitepapers section for technical details.

- **Why don't all models support more ROIs?**

Training a model requires 50-200 consistently contoured data sets, all with the same ROIs, from a variety of scanners, institutions, etc.

- **How do I set which specific GPU to use?**

At this time, you must use the `CUDA_VISIBLE_DEVICES` environment variables. See [Deployment](#).

- **Must I have a GPU?**

In theory, no. In practice, yes. While Tensorflow can run on CPUs, the performance is much, much slower. For example, during development, a Tesla K20c segmented a particular case in 30s, while that same case took 4m30s on a machine with Xeon E5-2695 48-cores @ 2.4GHz.

10 Dose Deformation

Dose deformation and RTDOSE exporting

Dose deformation (or warping) does not intrinsically use deformable registration, rather, it uses the product of deformable registration, the [deformation vector field](#).

10.1 Typical Workflow

The typical use case (and the only one currently supported by the UI) is to warp a dose *forward* in time. To do this, perform the following actions:

1. Load the patient into the Adaptive Tool
2. Use Ctrl-click to select both the Dose and SS and add as an Atlas. This is typically from the original plan.
3. Select the newer image set to register to and add as a Case Data. This is typically a daily scan.
4. Add the case and execute as normal.
5. The warped dose is written to the same location as the structure set, the [filename](#) will end with `_DOSE.dcm`.

Current limitations that will be addressed in future versions:

- If a Plan is not available, the dose will not appear in the Patient Data tree
- Select a pre-existing deformable spatial registration object and perform the warp without running a complete deformable registration
- Deform a dose *backward* in time, i.e., select a planning CT and a daily CT (with dose) with the goal of warping the dose *back* to the planning CT. Typically used for dose accumulation studies.

10.2 Implementation Details

When warping dose forward in time, the DVF that results from an atlas-based autosegmentation must be inverted. After inversion, the DVF is downsampled to match the resolution of the source dose grid. Since the vector field is uniformly-spaced in the Z-direction and RTDOSE grids can have non-uniform slice spacing, the mode of the spacings is used as the new Z-spacing. Trilinear interpolation is used when sampling into the source dose grid. All calculations are done in physical space (center of voxel).

10.3 DICOM

By default, ADMIRE will export the warped dose as an RTDOSE IOD series in the same study as the source dose, referencing the same plan. Since the standard does not provide any tags for RTDOSE IODs to represent derivation, ADMIRE will add an `Image Type (0008,0008)` tag with "DERIVED\SECONDARY" value as well as a `Derivation Description (0008,2111)`.

If your planning system does not support plans with multiple associated doses, enable the *Duplicate associated plan for warped doses* option in the Dose section of the Configuration settings. This will create a copy of the source plan with the following characteristics:

- new Series and Instance UIDs
- references the RTSS created during the deformable registration process for this case (instead of the RTSS used to develop the original plan)
- `Approval Status (300e,0002)` will be UNAPPROVED
- Description fields will clearly identify the plan as a "COPY_FOR_WARP".
- The warped RTDOSE will instead reference this duplicate plan.
- All `Isocenter Position (300a,012c)` and `Beam Dose Specification Point (300a,0082)` values will be warped using the same DVF as the dose grid. No other coordinates are modified.

No other changes are made to the plan—beams will not reflect the warped dose!

11 Artifacts, Naming Conventions

Output files, naming conventions.

Suffix	Format	Description
--------	--------	-------------

11.1 Overview

While a DICOM structure set (RTSS) is the primary output from ADMIRE, there are other files that may be written during the segmentation process.

Suffix	Format	Description
*_RTS.dcm	DICOM RT Structure Set	The new structures
STPLE_*_RTS.dcm	DICOM RT Structure Set	Final structures from a STAPLE case
PF_*_RTS.dcm	DICOM RT Structure Set	Final structures from a patch fusion case
*_DVF.mhd, .raw	ITK MetaImageIO format	Complete deformation vector field (composed with linear registration). .mhd is an ASCII file containing metadata (grid dimensions, etc). .raw is the grid of vectors, stored as binary floating point.
*_DVF_INV.mhd, .raw	ITK MetaImageIO format	Same as above, but with the inverse transformation.
*_DREG.dcm	DICOM Deformable Spatial Registration	Complete deformation vector field (composed with linear registration) stored as DICOM Deformable Spatial Registration IOD
*_DREG_INV.dcm	DICOM Deformable Spatial Registration	Same as above, but with the inverse transformation.
*_SREG.dcm	DICOM Spatial Registration	The linear registration stored as an affine transformation.
*_SREG_INV.dcm	DICOM Spatial Registration	Inverse of the above.
*_DOSE.dcm	DICOM RT Dose	Dose volume warped by a deformation vector field
*_PLAN.dcm	DICOM RT Plan	A duplicated plan resulting from dose warping; eases importing warped dose into a TPS.
STPLE_*_STATS.txt	ASCII	Specificity and sensitivity of each atlas used in a STAPLE case. Stored in tabular format.

11.2 Filename Generation

At this time, filenames are generated using the atlas name and patient name with one of the above suffixes, e.g., demoPRAtlas_demoPRpatient_RTS.dcm. If a name isn't available, "Unknown" is used for the name. For multi-atlas case types (STAPLE, patch fusion), intermediate atlases that are generated during the case use "Intermediate" as the atlas name. This is only relevant if the "Output intermediate results" configuration option is enabled.

11.3 Location

Files can be written to either a specific folder, or to the folder of the patient involved in the creation of the artifact.

11.4 Renaming

Once a filename has been created, if a file with the same name already exists at the destination folder, an increment will be added (e.g., `_RTS_1.dcm`, `_RTS_2.dcm`, etc) until it can be written. This holds for all output file types. Keep in mind this is only a string manipulation, and has no relation to the underlying case, (e.g., inpatient cases where there are typically fraction 1, fraction 2, etc).

Existing files will never be overwritten.

11.5 Deformed Images

If the configuration option is enabled, the DVF may be used to warp either the atlas onto the patient image (or vice-versa). This deformed image is useful for checking the performance of the registration. The image set is saved in DICOM format, and written to the folder `$atlasName_dfrmOnto_$patient` or `$patientName_dfrmOnto_$atlasName` depending on whether the forward or inverse DVF was used. The folder will be written to the same location as the other artifacts.

12 Tools

Standalone utilities

In the course of developing ADMIRE, various commandline utilities have been written. Some contain subsets of functionality, additional features that would be difficult to integrate into the main application, or prototypes. While they are often not as advanced as the main ADMIRE commandline interface and have limitations, we hope that advanced users may find them useful in the course of their research.

- [AtlasHammer](#) Forcefully associate a DICOM RT Structure Set with a set of images by re-writing UIDs
- [CBCTtoSCT](#) Create synthetic CT images directly from conebeam CT images
- [CompareContours](#) Calculate overlap statistics for contours
- [ConvertDCMtoITK](#) Convert DICOM to ITK/VTK images, surfaces, label maps
- `CudaBurn.exe` GPU burn-in/stress testing tool
- `CudaCheck.exe` Basic NVIDIA CUDA diagnostics
- [DoseWarpUtil](#) Apply DICOM deformation fields to DICOM dose and export a warped dose
- [ImageSimilarityUtil](#) Calculate statistics for images, ROIs, etc
- [ImageWarpUtil](#) Apply DICOM spatial registration and deformation fields to DICOM images and export a new image set
- [MAFusionUtil](#) Run STAPLE or Patch Fusion directly
- [REGtoITK](#) Convert DICOM Deformable Spatial Registrations objects to ITK DVFs
- [StandardizeMR](#) Normalize MR images
- `VTKtoRTSS.exe` Create a DICOM RT Structure Set from a set of .vtk surface files

12.1 AtlasHammer

Repairs an orphaned RT Structure Set by forcefully associating it with a new DICOM image set; changes Frame of Reference UIDs

12.1.1 Overview

A simple utility for forcefully re-associating a DICOM RTSS with a new image set. Sometimes due to errors in anonymization, UIDs aren't consistent. This simple utility will regenerate all the UIDs in an input structure set so that they "belong" to the specified image set. It will also copy the General Study and Patient Demographic attributes over.

12.1.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\ProgramData\ADMIRE>C:\Program Files\ADMIRE\AtlasHammer.exe --help
Repairs an orphaned RT Structure Set by forcefully associating it with a new DICOM image set;
changes Frame of Reference UIDs, Contour Image Sequence values, General Study and Patient
module attributes.
Use with caution.
```

```
-----
AtlasHammer:
--ssin arg      Filename of the original (broken) structure set
--ssout arg     Filename to store the fixed structure set
--images arg    Directory containing image set to match
-? [ --help ]  Print all program options
--version      Print program version information and exit.
```

12.1.3 Advanced Usage

Nothing requires the input structure set to *actually* have originated from the destination `--images` set.

12.1.4 Warnings

Nothing in this program can prevent you from associating the wrong RTSS with an image.

12.2 CBCTtoSCT

Note

This utility deprecated. Unless you need the ability to input/output ITK images directly, please use the [conventional ADMIRE command line interface](#) with the `--type dl_synimage` argument.

12.2.1 Overview

Generates a synthetic CT (referred to as *sCT* hereafter) from CBCT using deep learning. These refined images remove and correct artifacts present in many CBCTs. Images can be DICOM CT or CBCT image series, or any ITK-supported format.

12.2.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
CBCTtoSCT:
--model <model.pb.json>          Region-specific model to use
--src <file|dir>                 CBCT ITK file(s) or DICOM dir(s)
--output <file>                  ITK filename(s) to export the refined
                                image to; (e.g., refined.mha)
--output-dcm <dir>               Directory export the refined DICOM
                                images; if it doesn't exist, it will be
                                created.
--output-dcm-study <dir>         A DICOM directory or file containing
                                DICOM study metadata to be used for the
                                result
--series-description <string> (=DL sCT from CBCT)
                                (optional) Series description to use
                                for DICOM output
--series-number <string>         (optional) Series Number to use for
                                DICOM output
--patient-mask-threshold <intensity> If provided, override default patient
                                mask intensity threshold
--param <file|param setting> ... Path to a parameters file which
                                overrides default parameters. Can also
                                override values directly using
                                "name=val" syntax. If specified
                                multiple times, overrides are applied
                                in order.
-? [ --help ]                   Print all program options
--version                       Print program version information and
                                exit.
-@ [ --response-file ] <file>   Read in additional arguments from
```

Files specified by `--src` can be ITK images or DICOM folders –the program automatically determines which type. If `--output` is used, the specific ITK output type is determined by filename (e.g., `.mha`, `.nrrd`, `.nii`). If `--output-dcm` is used, a new derived CT DICOM image series is written.

12.2.2.1 Examples Both input and output are ITK images:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

Input is ITK, output is a different ITK format:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

Input is a DICOM image series, output is ITK:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

Input is a DICOM image series, output is a new, derived image series:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

Same as above, but setting a new Series Description and Number:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

Input is ITK, output is DICOM. To fill in all the DICOM metadata, a template image series must be provided. The resulting sCT image will be a new Series within the Study of the template:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_lr_0.000005_98000
```

12.2.2.2 More Examples Multiple images can be processed at once. This can help overall throughput as the deep learning libraries, models, etc, are only loaded once. `--src` can be repeated multiple times, have multiple values, or use wildcards. The following are all equivalent:

- `--src conebeam1.mha --src conebeam2.mha --src conebeam3.mha`
- `--src conebeam*.mha`
- `--src conebeam1.mha conebeam2.mha conebeam3.mha`

The following will refine 2 CBCTs in ITK format.

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL SCT Models\frozen_model_thorax_04022019.pb.json" --
```

If there are multiple input images and a single output directory, filenames will be generated automatically by appending `*_processed*` to the filename (if ITK) or using Instance UIDs (if DICOM). The following will result in `c:\sCT_results\conebeam1_processed.mha`, `c:\sCT_results\conebeam2_processed.mha`, `c:\sCT_results\conebeam3_processed.mha` being written.

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL SCT Models\frozen_model_thorax_04022019.pb.json" --
```

If the `--src` is a directory containing multiple CT or CBCT image series, they will all be processed:

```
> CBCTtoSCT.exe --model "C:\ProgramData\ADMIRE\DemoAtlases\DL SCT Models\frozen_model_thorax_04022019.pb.json" --
```

12.2.2.3 Patient Mask Threshold For pelvis region, masking the patient surface can be useful for removing artifacts (and couch) . A default intensity threshold of -600 is used, but for some scanners, it may be necessary to override it via `--patient-mask-threshold`. Voxels lower than this threshold will be removed. Common value is within [-850,-250]. Setting it to -9999 effectively disables this preprocessing step (because there are no voxels with intensity values below this, therefore no voxels are removed). This is the default for thorax region because masking tends to introduce artifacts in the lung air cavities. To summarize:

- If there are too many artifacts outside/on the patient surface:
 - Increase the value (e.g., -300) → *More* voxels removed → Shrinks patient mask area
- If the patient surface is clipped or truncated:
 - Decrease the value (e.g., -800) → *Fewer* voxels removed → Expands patient mask area

Do not set this parameter unless you are having problems with the defaults.

12.2.2.4 Response Files Commandline arguments can also be placed into a "response file" style text file. The contents of this file will be read in and processed exactly as if they were arguments on the original command line. This can be useful in some automation environments/workflows as response files can contain comments, extra whitespace, and can avoid shell-quoting problems.

c:\ex.rsp contents:

```
--src c:\conebeam.mha
--output-dcm c:\patient1\day2sCT
--model "C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_thorax_04022019.pb.json"

# This is a comment line and will be ignored

--output-dcm-study "c:\patient1\refCT"
```

Using the response file:

```
> CBCTtoSCT.exe -@ c:\ex.rsp
```

You can use both commandline and response files. Response file contents are always processed *after* actual commandline arguments (regardless of the order of arguments)

```
> CBCTtoSCT.exe --series-description "A new sCT" -@ c:\ex.rsp
```

12.2.3 Exit Codes

- 0: No error; all cases completed successfully
- 1: Exception; one or more cases did not complete successfully. See `CBCTtoSCT.lastExec.log` for details.
- 2: Not licensed for requested case type, feature, or model

12.2.4 Backwards Compatibility

Previous versions had hardcoded model files for each region, e.g., `--region pelvis`. Current versions prefer explicitly providing the model via a metadata file, e.g., via `--model C:\ProgramData\ADMIRE\DemoAtlases\DL sCT Models\frozen_model_HN_4000_G_Ir_0.000005_98000.pb.json``.

For compatibility, `ADMIREParameters.json` can set the metadata file to be used when the legacy `--region XXX` method is used. For example:

```
"DeepLearning": {
  "ImageSynthesizer": {
    "Pelvis": {
      // This is for backwards compatibility for CBCTtoSCT.exe --region pelvis
      "LegacyModel": "DL sCT Models/frozen_model_pelvis_7datasets_04022019.pb.json"
    }
  }
}
```

12.2.5 DICOM Notes

If the output is DICOM, the resulting image series will always have CT modality and Class UID. The Image Type will be set to "DERIVED\SECONDARY". The rest of the DICOM follows IHE-RO Resampled/Combined CT Series Storage [RO-11] rules ([RO TF-2: 3.11](#)) for deriving a new CT image series from an existing one.

12.2.6 Limitations

- While non-uniformly sliced, non-axial CBCT images are supported (the image volume is constructed and resliced to uniform, axial slices just as ADMIRE does), the resulting sCT will always have uniformly-spaced, axial slices in order to avoid any more interpolation error than necessary.

12.2.7 Model Details

The deep learning model is CycleGAN implemented using Tensorflow. The internal resolution is 256x256. Images are processed slice-wise.

12.2.8 Future Goals

- Increase model size to produce sharper images
- Allow sites to incrementally train the base model by providing additional training data sets
- Remove patient masking preprocessing step

12.3 CompareContours

Utility for evaluating accuracy of contours

12.3.1 Overview

To evaluate the performance of ADMIRE's deformable registration and segmentation algorithms, it is useful to compare the agreement of the new set of contours against a gold standard (e.g., from using a different algorithm, or created by a human expert). The most popular method is the DICE coefficient which measures how much the two structures overlap one another. DICE, however, is but one metric; for an introduction to several more, please read:

[Evaluation of whole - body MR to CT deformable image registration.](#) A.Akbarzadeh, D.Gutierrez, A.Baskin, M.R.Ay, A.Ahmadian, N.Riahi Alam, KO Lövsblad, H.Zaidi.

Also, technical details are available in [CmsAlgorithm::SurfaceComparisonStatistics](#)

12.3.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

The simplest comparison is to compare two DICOM RT Structure Sets. ROIs that have the same name will be compared. By default, the DICE coefficient, and maximum distance-to-face are output. The following example compares `test.dcm` against the gold standard `gold.dcm`. The results aren't particularly good.

```
> CompareContours.exe --truth c:\data\gold.dcm --test c:\data\test.dcm
```

```
ROI comparison tool.  Computes percent overlap, vertex distances, and other metrics
Loading truth RTSS: C:\data\pr.dcm
Loading test RTSS: C:\data\gen.dcm
Searching for truth image set for C:\data\pr.dcm
Searching for test image set for C:\data\gen.dcm
Timer: 1.201 s: compareContoursLoading
Calculating surface statistics
Converting 8 structures into surfaces, in-plane isospacing 1.000000
Converting 8 structures into surfaces, in-plane isospacing 1.000000
Calculating overlap statistics for 8 ROI pairs, sample spacing of 1.00 1.00 1.00
Comparing ROI 0 SKIN
Done comparing SKIN
Comparing ROI 1 FEMUR_LT
Done comparing FEMUR_LT
Comparing ROI 2 BLADDER
Done comparing BLADDER
Comparing ROI 3 PROSTATE
Done comparing PROSTATE
Comparing ROI 4 RECTUM
Done comparing RECTUM
Comparing ROI 5 PELVIS_LT
Done comparing PELVIS_LT
Comparing ROI 6 FEMUR_RT
Done comparing FEMUR_RT
Comparing ROI 7 PELVIS_RT
Done comparing PELVIS_RT
Timer: 12.735 s: compareContours
# CompareContours.exe v2.0
# Generated          : Thu Dec 17 15:56:52 2015
# Truth RTSS         : C:\data\gold.dcm
# Truth Image        :
# Testing RTSS        : C:\data\test.dcm
# Testing Image       :
# Sampling spacing    : 1.00 1.00 1.00
# Min acceptable DICE : 0.930000
# Max acceptable mean vertex-to-face distance (mm) : 1.500000
# Structure mappings file:
MappedName      TruthName      TestName      Dice      MeanDistance (mm)      StdDevDistance
"BLADDER"       "BLADDER"       "Bladder"     0.721      3.909                  3.484
"FEMUR_LT"      "FEMUR_LT"      "lt femur"    0.896      0.869                  1.109
"FEMUR_RT"      "FEMUR_RT"      "rt femur"    0.906      0.815                  1.060
"PELVIS_LT"     "PELVIS_LT"     "left pelvis" 0.874      0.908                  1.209
"PELVIS_RT"     "PELVIS_RT"     "right pelvis" 0.876      0.804                  1.021
"PROSTATE"      "PROSTATE"      "Prostate"    0.654      7.057                  4.628
"RECTUM"        "RECTUM"        "rectum"      0.752      3.420                  3.816
"SKIN"          "SKIN"          "external"    1.000      0.000                  0.000
```

12.3.2.1 Other Usage `--truth` and `--test` can both accept multiple files. Each file will be loaded in order, and the ROIs matched up by name, as per usual. Both DICOM and VTK formats are supported and can be mixed. For VTK formatted surfaces, the basename of the filename (path and extension is removed) is used as the ROI name. Since there is only one surface per file (e.g., `c:\gold\skin.vtk` will be assigned a name of "skin"). Because VTK surfaces will often have arbitrary filenames (e.g., `skin_orig.vtk`, `skin_deformed.vtk`), the names usually not match up. The `--match_by_position` is used to match up truth and test surfaces in the order they were provided on the commandline (or within the RTSS), regardless of filename.

```
> & 'C:\wa\trunk\x64\Release\CompareContours.exe' --match_by_postion `
--truth C:\gold\skin.vtk c:\gold\prostate.vtk c:\gold\femur_lt.vtk `
--test C:\test\external.vtk c:\test\prost.vtk c:\gold\lt_femur.vtk
```

An ADMIRE mappings file can be used via `--mappings`.

12.3.2.2 Binary Masks ITK binary mask representations of surfaces are also supported. Zero is considered background, and non-zero considered foreground. E.g.,

```
> & 'C:\wa\trunk\x64\Release\CompareContours.exe' --match_by_postion `
--truth C:\gold\skin.nii c:\gold\prostate.mha c:\gold\femur_lt.mha `
--test C:\test\external.nii c:\test\prost.mha c:\gold\lt_femur.nii
```

Note `--match_by_position` is required in the above scenario because the ROIs are named differently.

At this time, you cannot mix mesh (VTK, DCM) and mask representations in the same execution.

12.3.2.3 Spacing The `--spacing` parameter determines the resolution of the binary mask representation that are created from the DICOM contours or VTK surfaces. If only one value is provided, it's used for all 3 dimensions, but all three dimensions may have separate values, e.g., `--spacing 0.8 0.5 1.0`. Increasing the spacing will increase the runtime of the calculations—especially if an EXTERNAL surface is present (due to its large volume). If omitted, the spacing of the truth image (if found) is used as a default.

Because the binary masks represent discretization of the surface mesh, choosing a spacing value smaller than the original image resolution is desirable.

If the surfaces to be compared are already binary masks, the truth mask spacing is used as the default. If overridden, the mask will be resampled using nearest neighbor interpolation. If a finer spacing is chosen, it will not significantly affect the accuracy of most statistics because the upsampled mask will still be as coarse as the source.

12.3.3 Interpreting Results

12.3.3.1 Handling Outliers Some metrics, such as Hausdorff distance, are susceptible to outliers. To handle this, a `--distance_percentile` value between 0.0 and 1.0 may be provided. The default is 1.0, which means 100% of the distance values are included. This only affects the following stats:

- `meanDistance_mesh`
- `stdDevDistance_mesh`
- `hausdorff_avg`
- `hausdorff_mesh`

12.3.3.2 DICE Results In the preceding results, the SKIN structure has a DICE of 1.0 which denotes perfect overlap. The BLADDER is much lower at 0.7. In general, DICE coefficients 0.85 or higher are considered good agreement. DICE, however, is proportional to the volume of the structures being compared. For example, if the both the BLADDER and SKIN contours were all contracted by 1mm, the drop in DICE is much larger for the BLADDER than the SKIN. Because of this, DICE cannot be the only measure used.

The *MeanDistance* is the arithmetic mean of the distance from each vertex in the truth ROI to the surface of the test ROI. Continuing with our example, the BLADDER has an average difference of 4mm. It is apparent this test result does not handle BLADDER well. *MeanDistance* is useful to help interpret low DICE for tiny structures (i.e., EYE, PAROTIDS, etc). If the distance is small, then the lower DICE is most likely still acceptable.

The *StdDevDistance* is simply the standard deviation of the same distances.

12.3.3.3 Other Statistics Many more statistics are available. To see them, save the results to a tabular, csv, or json file.

```
CompareContours.exe --truth C:\data\gold.dcm --test C:\data\test.dcm `
--tabular results.txt --json results.json --csv results.csv
```

Opening `compare.txt` in a text editor reveals Hausdorff, Jaccard, and more (which have been elided to save space). [Detailed information is available here.](#)

```
# Generated          : Thu Dec 17 16:26:02 2015
# Truth RTSS        : C:\data\gold.dcm
# Truth Image       :
# Testing RTSS      : C:\data\test.dcm
# Testing Image     :
# Warped Testing Image :
# Sampling spacing   : 1.00 1.00 1.00
# Min acceptable DICE : 0.930000
# Max acceptable mean vertex-to-face distance (mm) : 1.500000
# Structure mappings file:
MappedName      TruthName      TestName      Dice      MeanDistance (mm)      StdDevDistance      Hausdorff      J
"BLADDER"       "BLADDER"       "BLADDER"     0.721      3.909                  3.484            24.172
"FEMUR_LT"      "FEMUR_LT"      "FEMUR_LT"     0.896      0.869                  1.109            45.091
"FEMUR_RT"      "FEMUR_RT"      "FEMUR_RT"     0.906      0.815                  1.060            39.243
"PELVIS_LT"     "PELVIS_LT"     "PELVIS_LT"     0.874      0.908                  1.209            41.437
"PELVIS_RT"     "PELVIS_RT"     "PELVIS_RT"     0.876      0.804                  1.021            42.063
"PROSTATE"      "PROSTATE"      "PROSTATE"     0.654      7.057                  4.628            22.579
"RECTUM"        "RECTUM"        "RECTUM"       0.752      3.420                  3.816            22.942
"SKIN"          "SKIN"          "SKIN"         1.000      0.000                  0.000            0.000
```

NOTE: Some programs will not ignore the comment lines in CSV or tabular formats (e.g., Matlab). If this is the case `--skip_output_headers` can be used to suppress them. Note that this will prevent error messages from being added to the files as well, such as missing ROIs.

12.3.3.4 Images While not normally used for comparing contours, some image intensity statistics can be useful for other purposes and they are easy to calculate once the hard part of discretizing the surface volumes has been done. The `--truthImage` and `--testImage` refer to the images volumes the surfaces were drawn upon. Usually, these are the images that are associated with each set of surfaces. When provided, some basic statistics are calculated over the *image voxels where the corresponding mask is nonzero*. These include:

- arithmetic mean
- standard deviation
- signal difference to noise ratio (SDNR)

Images can be provided as any format supported by ITK or can be a directory of DICOM images. The statistics will simply report NaN values if images aren't provided.

NOTE: For this step, the image is resampled to match the spacing of the surface binary mask using linear interpolation.

12.3.4 Recommendations

- Prefer RTSS or VTK surfaces as inputs and choose a `--spacing` value smaller than the original image resolution.
- If inputs are already binary masks, choose a `--spacing` that matches the finer of the truth/test structures. Smaller values only increase runtime—not accuracy.
- If converting surfaces to masks elsewhere (i.e., [ConvertDCMtoITKVTK](#)), convert using the same `--spacing` you intend to use with CompareContours

12.3.5 Implementation details

The process is as follows:

- Match ROIs by semantic name via the structure mappings (or in commandline order if `--match_by_position` is specified)
- For each ROI pair...
 - If mesh representations were provided:
 - * compute mesh statistics (distance-to-surface, surface area, etc)
 - * compute bounding box of surface meshes
 - * using bounding box, create binary mask of both ROIs using the `--spacing` parameter
 - If binary mask representations were provided (or using masks created above)
 - * Resample to `--spacing` if necessary
 - * compute overlap statistics using binary masks (DICE, Jaccard, sensitivity, specificity, etc)
 - * compute optional image intensity stats within volume defined by binary mask if `--truthImage` and `--testImage` were provided.
- For each output type (csv, tabular, json)...
 - save statistics

12.3.6 Reference

<code>--truth <file></code>	Truth file(s)
<code>--test <file></code>	Test file(s)
<code>--truthImage <file dir></code>	Optional image file/dir for truth file(s); enables intensity metrics for image data within surfaces
<code>--testImage <file dir></code>	Optional image file/dir for test file(s); enables intensity metrics for image data within surfaces
<code>--spacing x y z</code>	Voxel grid spacing (mm) to use when building surfaces from 2D contours. Smaller values increase runtime and accuracy. Default value is the spacing of the truth image
<code>--mappings <file></code>	Structure mappings file (optional). Maps structure names from both RTSSs before comparing them. If unset, structure names are used as-is.
<code>--min_dice n (=0.93000000000000005)</code>	Minimum acceptable dice coefficient (1.0 = perfect, default = 0.93); only affects the return code and the ExceedsDice convenience statistic
<code>--max_distance n (=1.5)</code>	Maximum acceptable mean vertex-to-surface distance difference (default = 1.5mm); only affects the return

<code>--distance_percentile n (=1)</code>	code and the ExceedsDistance convenience statistic Discard outliers from distance and distance_mesh metrics; (default is 1.0 which includes all distances)
<code>--match_by_position</code>	Match truth and test surfaces by commandline argument position (or definition order in RTSS) instead of ROI name. Usually used when individual VTK surfaces meshes or ITK surface masks are being compared.
<code>--skip_output_headers</code>	Remove error messages, informational headers from CSV and tabular formats
<code>--csv <file></code>	Filename to store comparison results in CSV format
<code>--tabular <file></code>	Filename to store comparison results in tabular format
<code>--json <file></code>	Filename to store comparison results in JSON format
<code>-@ [--response-file] <file></code>	Read in additional commandline arguments from a file
<code>--nostatusrc</code>	Sets return code to 0 regardless of min_dice or max_distance failures. Other error codes are returned normally.
<code>-? [--help]</code>	Print all program options
<code>--version</code>	Print program version information and exit.

12.4 ConvertDCMtoITKVTK

12.4.1 Overview

A general utility for converting a DICOM image set + structure set into ITK/VTK images/surfaces/label maps. The primary usage is for bulk preprocessing DICOM data into raw image data for use in training deep learning models. However, it is also a suitable tool for generalized conversion of DICOM images and structure sets into raw image and mesh formats (anything supported by ITK/VTK).

It also contains some statistical reporting abilities to help with filtering and organizing large datasets.

12.4.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory. The general execution is:

1. Read DICOM image and RTSS from `--src` directory
2. If `--spacing` is provided, resample to this spacing; otherwise the original image spacing is retained. Oriented images and non-uniformly sliced volumes will be re-sampled into uniformly spaced standard DICOM RCS (linear interpolation).
3. Save resampled image to `--outputImage`.
4. For each RTSS ROI, create a surface mesh
5. Map ROI name to canonical name according to the `--mappings` file
6. If image is CT, create an EXTERNAL surface if it is missing and `--addExternalROI` is specified
7. Sort ROIs by largest volume size first (if `--labelOrder` is omitted). Otherwise, sort according to `--labelOrder`
8. For each ROI surface mesh
 - Convert to binary mask. Mask image size and spacing will be same as the resampled image.

- Save mesh to `--output dir` (if `--outputSurface` is specified)
 - Save binary mask to `--output dir` (if `--outputSurfaceMask` is specified)
9. Combine all binary masks (or only those specified by `--labelOrder`) into a single label map image. Label values will start at 0 (background) and be assigned in ascending value according to the sorted order. Save to file specified by `--outputLabelMap`.

These steps are repeated for each `--src` image series.

12.4.3 Supported Formats

For inputs, DICOM images and RT Structure Sets are read. For outputs, the filename extension determines the format. In general these include anything supported by ITK and VTK. Most common ones are `.mha`, `.nii`, `.nii.gz` (for images and masks) and `.vtp`, `.vtk`, `.stl`, and `.ply` (for surface meshes).

12.4.4 Filename string formatting

All output filenames support some basic formatting flags (powered by the `fmt library`). For example:

Format	Output
<code>{patientid}_src.mha</code>	<code>patient701011_src.mha</code>
<code>{patientid}/{modality}_src.nii</code>	<code>patient701011/CT_src.mha</code>
<code>{label:02d}_{roi}_mask.nii</code>	<code>01_SKIN_mask.nii</code>
<code>{patientname}_src.mha</code>	<code>Richard^Jones_src.mha</code>

All filenames are relative to the `--output` directory. However, as seen above, subdirectories can be created automatically.

12.4.5 Examples

Assuming an input directory that looks like this, which represents a single patient.

```
E:\Data\1050877095
E:\Data\1050877095\CT\image1.dcm
E:\Data\1050877095\CT\image2.dcm
...
E:\Data\1050877095\CT\image256.dcm
E:\Data\1050877095\RTSS.dcm
```

and a mappings file containing:

```
EXTERNAL, skin, body
PROSTATE, prostaat
PELVIS_LT, pelvisL, L pelvis, left pelvis
PELVIS_RT, pelvisR, R pelvis, right pelvis
# ..etc
```

All ROIs, resampling images and label map to 2x2x2.5mm

```
C:\ProgramData\ADMIRE> %"C:\Program Files\ADMIRE\ConvertDCMtoITKVTK.exe"
--mappings .\TG263Mapping.txt
--src "E:\Data\1050877095"
--output outdir
--outputLabelMap="{patientid}_seg.nii"
```

```
--outputImage="{patientid}.nii"
--outputSurfaceMask="{label:02d}_{roi}_mask.mha"
--spacing 2 2 2.5
```

A subset of ROIs, combined label map and image only

```
C:\ProgramData\ADMIRE> &"C:\Program Files\ADMIRE\ConvertDCMtoITKVTk.exe"
--mappings .\TG263Mapping.txt
--src "E:\Data\1050877095"
--output outdir
--outputLabelMap="{patientid}_seg.nii"
--outputImage="{patientid}.nii"
--labelOrder PROSTATE PELVIS_LT PELVIS_RT
--spacing 2 2 2.5
```

A subset of ROIs in final combined label map, autogen EXTERNAL contour if necessary using a non-default thresholding value

```
C:\ProgramData\ADMIRE> &"C:\Program Files\ADMIRE\ConvertDCMtoITKVTk.exe"
--mappings .\TG263Mapping.txt
--src "E:\Data\1050877095"
--output outdir
--outputLabelMap="{patientid}_seg.nii"
--outputImage="{patientid}.nii"
--labelOrder EXTERNAL PROSTATE PELVIS_LT PELVIS_RT
--addExternalROI -600
--spacing 2 2 2.5
```

Just the images, at the original resolution, using the deepest subdirectory in the filename. Useful for when the DICOM patient demographic information is anonymized to oblivion. Consider a directory of patients:

```
> dir
E:\Data\TrainingPatient1\*.dcm
E:\Data\TrainingPatient2\*.dcm
E:\Data\TrainingPatient3\*.dcm
E:\Data\readme.txt
C:\ProgramData\ADMIRE> &"C:\Program Files\ADMIRE\ConvertDCMtoITKVTk.exe"
--src "E:\Data\Training*"
--output outdir
--outputImage="{subdir}.nii"
# Will result in:
outdir\TrainingPatient1.nii
outdir\TrainingPatient2.nii
outdir\TrainingPatient3.nii
```

12.4.6 ROI Subset Filtering

ADMIRE's autoseg algorithms require each of the training data to have all the labels being trained. Unfortunately, it is often difficult to obtain large datasets with consistent labeling. To help determine the correct subset of ROIs, the tool can generate a list of largest common subsets (adapted from the [FP-Growth algorithm](#) implementation from Bart Goethals). To do this, the `--statsOnly` parameter will export a .CSV table that that enumerates the largest, most frequently occurring sets of ROIs.

```
ConvertDCMtoITKVTk.exe --mappings c:\data\TG263Mapping.txt
--src E:\data\*
--statsOnly E:\tmp\report.csv
--minDataSetSupport 10
...
---- Calculating largest common ROI subset on 16 inputs
BRAIN
BRAINSTEM
ESOPHAGUS
EXTERNAL
EYE_L
EYE_R
INNEREAR_L
INNEREAR_R
OPTICNERVE_L
OPTICNERVE_R
PAROTID_R
SPINALCORD
---- Largest common subset has 12 ROIs which exist on ALL 16 sources
BRAIN, BRAINSTEM, ESOPHAGUS, EXTERNAL, EYE_L, EYE_R, INNEREAR_L, INNEREAR_R, OPTICNERVE_L, OPTICNERVE_R,
PAROTID_R, SPINALCORD
```



```

-----
E:\data\DLH01
E:\data\DLH02
E:\data\DLH03
E:\data\DLH04
E:\data\DLH19
E:\data\DLH20
E:\data\DLH21
E:\data\DLH22
E:\data\DLH23
E:\data\DLH24
E:\data\DLH25
E:\data\DLH26
E:\data\DLH27
E:\data\DLH28
E:\data\DLH29
E:\data\DLH30
-----

---- Performing frequent item mining to find the top 20 common ROI subsets that appear in AT LEAST 10 of 16
      inputs
Sorting 765951 possible subsets
Top 20 ROI subsets found in at least 10 of 16 inputs
Writing detailed report to c:\data\test_output\report.csv
18:10: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19]
18:10: [ 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19]
18:10: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 19]
18:10: [ 1 2 3 4 5 6 7 8 9 10 11 13 14 15 16 17 18 19]
17:11: [ 1 2 3 4 5 6 7 8 9 10 11 13 14 15 16 18 19]
17:11: [ 2 3 4 5 6 7 8 9 10 11 13 14 15 16 17 18 19]
17:11: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 15 16 18 19]
17:11: [ 1 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19]
17:11: [ 0 2 3 4 5 6 7 8 9 10 11 12 15 16 17 18 19]
17:11: [ 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 19]
17:11: [ 1 2 3 4 5 6 7 8 9 10 11 12 15 16 18 19 20]
17:10: [ 1 2 3 4 5 6 7 8 9 10 11 14 15 16 17 18 19]
17:10: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 15 17 18 19]
17:10: [ 1 2 3 4 5 6 7 8 10 11 13 14 15 16 17 18 19]
17:10: [ 2 3 4 5 6 7 8 9 10 12 13 14 15 16 17 18 19]
17:10: [ 2 3 4 5 7 8 9 10 11 12 13 14 15 16 17 18 19]
17:10: [ 1 2 3 4 5 6 7 8 9 10 11 12 13 15 16 18 19]
17:10: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 15 16 17 19]
17:10: [ 2 3 4 5 6 7 8 9 10 11 12 13 14 16 17 18 19]
17:10: [ 0 1 2 4 5 6 7 8 9 10 11 12 15 16 17 18 19]

```

The first common subset is that in which each ROI appears on *EVERY* input. Often, this is much smaller than desired but is very fast to compute. Then Frequent Item Mining is performed to find the largest alternate groups of ROI that appear IN *AT LEAST* some number of datasets. The `--minDataSetSupport 10` sets the minimum number of datasets. In the the console output above, `18:10: [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18]`, is showing that the largest group of ROIs is 18 and it appears in 10 of the 16 datasets. Only the ROI numbers are printed for brevity. The CSV report replicates the same data, but prints the mapped ROI name instead of the number. Load it into Excel to better visualize it.

12.4.6.1 All ROI Summary If a filename is provided, e.g., `--statsOnly report.csv`, a second report will be written: `report.allrois.csv`. This is a simple table of all the datasets and all the ROIs in each one. This serves as a cross reference to find which datasets are missing which ROIs. Rows are sorted by filename, while columns are sorted by the number of datasets the ROI appears in. This means ROIs that aren't consistently present will be further to the right.

Use Excel to visualize or Powershell (with a very wide console window):

```

> Import-Csv c:\data\test_output\report.allrois.csv | Format-Table * -AutoSize
FileName      ROICount  BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
-----
C:\data\DLH01 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH02 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH03 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R              INNEREAR_R OPTICNERVE_L
C:\data\DLH04 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH05 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R
C:\data\DLH06 18      BRAIN BRAINSTEM EXTERNAL EYE_L              INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH07 18      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH08 17      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L
C:\data\DLH09 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L

```

```

C:\data\DLH10 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH11 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
C:\data\DLH12 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R
C:\data\DLH13 19      BRAIN BRAINSTEM EXTERNAL EYE_L EYE_R INNEREAR_L INNEREAR_R OPTICNERVE_L
...etc

```

Usually, it takes several invocations, varying the `minDataSetSupport` number to find the most suitable grouping. It is up to the user to determine the tradeoff between ROI coverage and data set coverage. If the option value is less than 1, it's interpreted as a percentage, i.e., 0.85 = 85% of the input data sets.

12.4.6.2 Additional Filtering After looking over the initial results, it is often useful to exclude some ROIs from the sets because the user does not want it be part of the model. `--exclude` is used to pre-emptively remove these ROIs from consideration. This improves the chances of assembling larger common ROI subsets over a larger number of the datasets.

```

ConvertDCMtoITKVTX.exe --mappings c:\data\TG263Mapping.txt
--src E:\data\*
--statsOnly c:\tmp\report.csv
--minDataSetSupport 10
--excludeLabels BRAIN BRAINSTEM

```

Analogously, `--requireLabels` can be used to filter the list of possible subsets to those which contain all of the labels.

12.4.7 Putting It All Together

When processing a large number of datasets, the workflow is usually:

1. Use `--statsOnly report.csv` to generate initial combinations. If all data is on an SSD, the `--jobs` parameter can be set rather large as disk I/O is the primary bottleneck during `--statsOnly` phase.
2. Look at `report.csv` column headers and check for additional mappings that can be added. Look for misspelled ROI names and inconsistent naming.
3. Correct ROI names via mappings or modifying original DICOM data
4. Start selecting ROIs to use to explicitly require or exclude
5. Re-run `--statsOnly` with the additional restrictions of `--excludeLabel` and `--requireLabel`
6. Go to 1. Repeat as necessary until an acceptable balance of subset size to input data set coverage. Lower `--minDataSetSupport` as necessary.

Note: If `--statsOnly` is specified, no conversions, resampling, label map creation will be performed. It's meant to be a mode to help determine *which* ROIs to convert into training data.

1. Once an acceptable set of labels is found, remove the `--statsOnly`
2. Set `--outputImage` and `--outputLabelMap` accordingly
3. Set `--labelOrder` accordingly. Some of the console output from the prior `--statsOnly` executions will have a list of the labels in descending order of organ volume.
1. Add `--requireAllLabels` to skip conversion of any inputs that don't have *ALL* the labels in `--labelOrder`. This ensures the `--output` dir only contains the data that is suitable for the training.

12.4.8 Automation

The program will recursively scan the `--src` folder for patients, so if directories are used in the output filename strings, everything can be processed in one pass. For example, assuming the following directory layout:

```
E:\Data\Patient1
E:\Data\Patient1\CT\image1.dcm
E:\Data\Patient1\CT\image2.dcm
...
E:\Data\Patient1\CT\imageN.dcm
E:\Data\Patient1\RTSS.dcm
E:\Data\Patient2
E:\Data\Patient2\CT\image1.dcm
E:\Data\Patient2\CT\image2.dcm
...
E:\Data\Patient2\CT\imageN.dcm
E:\Data\Patient2\RTSS.dcm
```

Using the following commandline:

```
C:\ProgramData\ADMIRE> &"C:\Program Files\ADMIRE\ConvertDCMtoITKVRTK.exe"
--mappings .\TG263Mapping.txt
--src "E:\Data"
--output outdir
--outputLabelMap="{patientid}/image_seg.nii"
--outputImage="{patientid}/image.nii"
--labelOrder EXTERNAL PROSTATE PELVIS_LT PELVIS_RT
--outputSurfaceMask="{patientid}/{label:02d}_{roi}_mask.mha"
--outputSurface="{patientid}/{label:02d}_{roi}.vtp"
```

Will result in:

```
outdir\Patient1\image_seg.nii
outdir\Patient1\image.nii
outdir\Patient1\01_EXTERNAL_mask.mha
outdir\Patient1\02_PROSTATE_mask.mha
outdir\Patient1\03_PELVIS_LT_mask.mha
outdir\Patient1\04_PELVIS_RT_mask.mha
outdir\Patient1\01_EXTERNAL.vtp
outdir\Patient1\02_PROSTATE.vtp
outdir\Patient1\03_PELVIS_LT.vtp
outdir\Patient1\04_PELVIS_RT.vtp
outdir\Patient2\image_seg.nii
outdir\Patient2\image.nii
outdir\Patient2\01_EXTERNAL_mask.mha
outdir\Patient2\02_PROSTATE_mask.mha
outdir\Patient2\03_PELVIS_LT_mask.mha
outdir\Patient2\04_PELVIS_RT_mask.mha
outdir\Patient2\01_EXTERNAL.vtp
outdir\Patient2\02_PROSTATE.vtp
outdir\Patient2\03_PELVIS_LT.vtp
outdir\Patient2\04_PELVIS_RT.vtp
```

The `--jobs` option will set the number of data sets to process in parallel. During a `--statsOnly` run, this value can be rather high because memory usage is fairly low (disk I/O is the bottleneck). If not a `--statsOnly` execution, memory usage is much, much higher, so the job number must be lower, e.g., 4-6.

12.4.9 Misc

12.4.9.1 StatsOnly The `--statsOnly` "mode switch" is somewhat strange and doesn't seem to fit the theme of this utility. However, this utility began as a way to bulk convert and organize a LOT of DICOM into formats suitable for deep learning training. For ADMIRE's current autoseg models, determining the largest common subsets of ROIs was a time-consuming operation. Since the two activities are done together (often iteratively) and the directory scanning and intermediate collation data structures are the same, the two activities are combined into the same program for technical simplicity. The goal being that once `--stats-only` runs were done, removing just this parameter would leave a commandline suitable for obtaining the correct subset of the converted data.

12.4.9.2 Autocreation of EXTERNAL contours `--addExternalROI` works best on CT images, less well on CBCT, requires tweaking the threshold value, and is unsupported on MR.

12.5 DoseWarpUtil

Standalone Dose warp tool

12.5.1 Overview

A utility for warping RT Dose files by applying Deformable Spatial Registration Objects to RT Dose files.

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE> "C:\Program Files\ADMIRE\DoseWarpUtil.exe" --help
-----
Dose Warp:
  --src-dose <file>           Source dose to be warped
  --src-img <dir>             Source image the dose is defined on.
                              Required if method is EMT
  --src-plan <file>           (optional) Plan to be warped
  --method arg (=Legacy)      Dose warp method to use. Options are:
                              DDM
                              EMT
                              Legacy
  --transform <file>          Transformation
  --transform-inv <file>      Inverse transformation
  --output-dose <file>        Filename to store warped result
  --output-plan <file>        Optional filename to store plan result (if
                              --src-plan is used)
  --output-plan-rtss-ref <file> Optional RTSS the warped plan should
                              reference (if --output-plan is used)
  --target-grid <file|dir>     Optional path to a DICOM/ITK object that
                              defines origin/size/spacing/orientation of
                              the warped dose
  --target-spacing <N,N,N>    Override the warped dose spacing if
                              --target-grid was used. The size will be
                              recalculated.
  --param <file|param setting> ... Override ADMIREParameters.json
  --dump <dir>                Debugging directory to dump to
  -? [ --help ]               Print all program options
  --version                   Print program version information and exit.
  -@ [ --response-file ] <file> Read in additional commandline arguments
                              from a file
```

12.5.2 Dose Warping Algorithm

There are 3 dose warping algorithms to choose from by using the `--method [arg]`

12.5.2.1 DDM: Direct Dose Mapping Implements the method described by [Haisen S Li et al.](#) It is most similar traditional image warping. Summarized:

1. Upsample source dose match the accompanying `--srcImage` resolution.
2. Warp the dose corner points (push forward) using the forward DVF to define the warped dose image volume.

3. The warped dose volume is "pulled back" by using the inverse DVF to calculate the location in the upsampled source dose, using trilinear interpolation when necessary.
4. The warped dose is downsampled back to the original dose resolution.

Minimum inputs

- forward DVF
- inverse DVF
- planning image referenced by the dose
- planning dose

Example

```
DoseWarpUtil.exe --method DDM
--src-img c:\patient\planCT
--src-dose c:\patient\planCT\origDose.dcm
--transform c:\patient\planCT\dreg.dcm
--transform-inv c:\patient\planCT\dreg_inv.dcm
--output-dose warpedDose.dcm
```

12.5.2.2 EMT: Electron-Mass Transfer Implements the method described by [Ziegenhein, P., Kamerling, C.P., Fast, M.F. et al.](#) Instead of warping dose values directly, it determines the actual energy and mass which is transferred and then divides this energy by the mass to determine the final dose.

Unlike DDM, it requires only the forward DVF. However, it additionally requires a CT-to-ED curve.

Minimum inputs:

- forward DVF
- planning CT image referenced by the dose
- planning dose
- CT-to-ED curve

Example

```
DoseWarpUtil.exe --method EMT
--src-img c:\patient\planCT
--src-dose c:\patient\planCT\origDose.dcm
--transform c:\patient\planCT\dreg.dcm
--output-dose warpedDose.dcm
```

CT-to-ED curve A few curves are provided in `ADMIREParameters.json`. The user is expected to create their own curve(s) for their own CT machines. For users of Elekta's Monaco planning system, these values should be copied directly here and given a name (which must be unique). The default curve name can be changed permanently by editing the file, or overridden at runtime by specifying `--param DoseWarp.EMT.CTtoEDcurve=YourNewName`. The relevant parameters are excerpted below:

```
{
  "ADMIRE": {
    "Core": {
      // ...
      "CTtoEDcurves": {
        "DICOMerweitert": {
          "CT": [ -1040, -1000, -773, -516, -72, -34, -4, 42, 49, 238, 951, 2000, 5000 ],
          "ED": [ 0.001, 0.050, 0.190, 0.490, 0.950, 0.980, 1.000, 1.040, 1.050, 1.120, 1.510, 2.000,
3.500 ]
        },
        "SIEMENS": {
          "CT": [ -1000, -773, -516, -72, -34, -4, 42, 49, 238, 951, 2000 ],
          "ED": [ 0.000, 0.190, 0.490, 0.950, 0.980, 1.000, 1.040, 1.050, 1.120, 1.510, 2.000 ]
        }
      }
    },
    // ...
    "Case": {
      // Default dose warp method when the GUI is used. Value must be one of the following:
      //
      // Legacy
      // DDM
      // EMT
      "DoseWarpMethod": "DDM"
    }
  },
  // ...
  "DoseWarp": {
    "EMT": {
      // Default curve to use when EMT method is used
      "CTtoEDcurve": "DICOMerweitert"
    }
  }
}
```

Examples Temporarily use "SIEMENS" CT-to-ED curve instead of the default (which is "DICOMerweitert")

```
DoseWarpUtil.exe --method EMT
--param "DoseWarp.EMT.CTtoEDcurve=SIEMENS"
--src-img c:\patient\planCT
--src-dose c:\patient\planCT\origDose.dcm
--transform c:\patient\planCT\dreg.dcm
--output-dose warpedDose.dcm
```

12.5.2.3 Legacy The method initially implemented by ADMIRE. Warps the dose grid using the traditional ITK image warp filter, using trilinear interpolation. Remains only for comparison purposes. Use DDM or EMT instead. It is a "pull-back" warp method.

Minimum inputs:

- inverse DVF
- planning dose

Examples

```
DoseWarpUtil.exe --method Legacy
--src-dose origDose.dcm
--transform-inv dreg_inv.dcm
--output-dose warpedDose.dcm
```

12.5.3 Plan Duplication

Some planning systems do not allow multiple RT Dose files to refer to the same RT Plan. To work around this, the user may derive a new plan by specifying `--src-plan`. A new plan will be derived and written to `--output-plan` with RT Plan Label of "COPY_FOR_WARP". Any Beam Dose Specification Point (300a,0082) and **Isocenter Position (300a,012c)** values will be transformed according to the `--transform` (if it is present). The warped dose will then reference this new "warped" derived plan.

```
DoseWarpUtil.exe --method EMT
--param "DoseWarp.EMT.CTtoEDcurve=SIEMENS"
--src-img c:\patient\planCT
--src-dose c:\patient\planCT\origDose.dcm
--src-plan origPlan.dcm
--transform c:\patient\planCT\dreg.dcm
--output-dose warpedDose.dcm
--output-plan warpedPlan.dcm
```

If the DVFs were generated by ADMIRE, there likely exists an RT Structure Set that contains the warped structures. Since RT Plan objects reference a specific structure set, the derived plan should reference this structure set by adding:

```
--output-plan-rtss-ref warpedRTSS.dcm
```

If this is not provided, the derived plan will reference the same RT Structure Set as the source RT Plan (there is no alternative). This is unlikely what you want.

12.5.4 Target Grid Considerations

By default, the output grid of the warped dose will have the source dose voxel spacing and is obtained by warping the corners of the source dose and taking the bounding box. This is why the DDM and Legacy methods use both the forward and inverse DVF, even though technically only the inverse is needed to warp (pull-back) the dose voxels. Since EMT is a push-forward method (and only requires the forward DVF), the warped dose will have the same grid as if DDM were used. This enables easy comparison between the two methods since they will have the same origin, spacing, size, and orientation.

For dose accumulation workflows, this choice of target grid is not ideal. When warping multiple doses, the source dose grids vary, resulting in different warped image grids. This would necessitate resampling the warped doses into a common grid. Interpolation of dose values is precisely what EMT seeks to avoid.

To ensure each warped dose occupies the exact same grid, use `--target-grid` to specify a DICOM image set, dose, deformable spatial registration object, or any 3D ITK image file. The spacing of this object will be used unless overridden by `--target-spacing`.

12.5.4.1 Dose Back-Projection

An example of accumulating dose on the planning CT (aka "back-projection"):

```
# Warp fraction 1 to plan

DoseWarpUtil.exe --method EMT
--src-img c:\patient\Fx1
--src-dose c:\patient\Fx1\srcDose.dcm
--transform c:\patient\Fx1\transformToPlan.dcm
--target-grid c:\patient\Plan\planDose.dcm
--output-dose Fx1_dose_on_plan.dcm

# Warp fraction 2 to plan

DoseWarpUtil.exe --method EMT
--src-img c:\patient\Fx2
--src-dose c:\patient\Fx2\srcDose.dcm
--transform c:\patient\Fx2\transformToPlan.dcm
--target-grid c:\patient\Plan\planDose.dcm
--output-dose Fx2_dose_on_plan.dcm

# Warp fraction 3 to plan

DoseWarpUtil.exe --method EMT
--src-img c:\patient\Fx3
--src-dose c:\patient\Fx3\srcDose.dcm
--transform c:\patient\Fx3\transformToPlan.dcm
--target-grid c:\patient\Plan\planDose.dcm
--output-dose Fx3_dose_on_plan.dcm
```

At this point, `Fx1_dose_on_plan.dcm`, `Fx2_dose_on_plan.dcm`, `Fx3_dose_on_plan.dcm` all occupy the same image grid (same origin, size, spacing, orientation) as the original planning dose, `planDose.dcm`.

12.5.4.2 Dose Forward Projection

An example of accumulating previously delivered doses on a new daily scan. Since there's not yet a dose on the current scan, use the current scan *image set* as the target grid, but use `--target-spacing` to override the spacing to 3mm. Otherwise, the CT spacing would be used and this is too fine for EMT to operate correctly.

```
# Warp plan to current

DoseWarpUtil.exe --method EMT
--src-img c:\patient\Plan
--src-dose c:\patient\Plan\planDose.dcm
--transform c:\patient\current\transformPlanToCurrent.dcm
--target-grid c:\patient\current
--target-spacing 3 3 3
--output-dose plan_dose_on_current.dcm

# Warp fraction 1 to current

DoseWarpUtil.exe --method EMT
--src-img c:\patient\Fx1
--src-dose c:\patient\Fx1\srcDose.dcm
--transform c:\patient\current\transformFx1ToCurrent.dcm
--target-grid c:\patient\current
--target-spacing 3 3 3
--output-dose fx1_dose_on_current.dcm

# Warp fraction 2 to current

--src-img c:\patient\Fx2
--src-dose c:\patient\Fx2\srcDose.dcm
--transform c:\patient\current\transformFx2ToCurrent.dcm
--target-grid c:\patient\current
```



```
--target-spacing 3 3 3
--output-dose fx2_dose_on_current.dcm

# Warp fraction 3 to current

--src-img c:\patient\Fx3
--src-dose c:\patient\Fx3\srcDose.dcm
--transform c:\patient\current\transformFx3ToCurrent.dcm
--target-grid c:\patient\current
--target-spacing 3 3 3
--output-dose fx3_dose_on_current.dcm
```

At this point, `plan_dose_on_current.dcm`, `fx1_dose_on_current.dcm`, `fx2_dose_on_current.dcm`, and `fx3_dose_on_current.dcm` all occupy the same image grid (same origin, orientation) as the current CT, but at 3x3x3mm spacing.

12.5.4.3 EMT Spacing Requirements Since EMT is a "push-forward" deformation method and does not interpolate dose values, it is sensitive to the spacing of the target dose grid. Generally, the target grid in-plane spacing should have the same spacing as the source dose, or no finer than 2x the source dose in-plane spacing. Otherwise, holes, negative, or very large dose values can occur. Currently, the program will abort if too small a spacing is specified or if any warped dose values are negative.

12.5.5 Current limitations

- Only dose encoded as a multiframe image is supported. Isodose curves or points are not supported.
- The default method is "Legacy" in order to maintain backward compatibility with the Monaco Research Interface. It should not be used for new work.
- No consistency checking is done to verify that the DREG belongs to the same patient as the dose.
- No consistency checking is done to verify the DREG source frame of reference UID. If the resulting dose is blank or strange looking, try using the opposite DREG.
- Only the first non-empty vector grid in the `Deformable Registration Sequence (0064,0002)` is used;
- The `Pre Transformation Matrix (0064,000f)` is used; however, the `Post Transformation Matrix (0064,0010)` is ignored as per `IHE-RO DRRO TF 1.2` recommendations.
- All inputs and outputs are DICOM; support for ITK formats will be added in the future.

12.5.6 GUI Notes

- If dose warping is performed via the ADMIRE GUI, the default method is DDM. In order to use EMT, the "`ADMIRE.Case.DoseWarpMethod`" parameter value in `ADMIREParameters.json` must be modified as described above.
- All RT Plan duplication and structure set references are handled automatically since dose warping is done as part of the DIR workflow.

12.6 ImageSimilarityUtil

Utility for calculating status over a whole image and (optionally) within regions defined by structures.

12.6.1 Overview

One way of evaluating a deformable registration algorithm is by applying the transformation to one of the images (creating a "warped image"). Now that the images are in the same frame of reference, similarity metrics can be calculated. However, this a global metric; it can be useful to calculate metrics in only the region around a specific structure. This tool provides both abilities.

12.6.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE> "C:\Program Files\ADMIRE\ImageSimilarityUtil.exe" --help
ImageSimilarityUtil.exe 3.0
Compute various image similarity metrics over an entire volume and within bounding boxes of individual ROIs.
--csv arg                               Filename to store comparison results in
                                         CSV format
--truthRTSS arg                         Truth RTSS file (optional)
--truthImage arg                       Image file/dir for truth RTSS. Can be
                                         DICOM or any ITK format
--testImage arg                       Image file(s)/dir(s) for test RTSS; can
                                         be DICOM or any ITK format
--histogramBins arg (=120)             Mutual information joint histogram bins
                                         when calculating image similarity
                                         metrics (default is 120)
--roiPadding arg                      Number of voxels to pad the bounding
                                         boxes of truth surfaces (default is 5 5
                                         5)
--mappings arg                        Structure mappings file (optional).
                                         Maps structure names from both RTSSs
                                         before comparing them. If unset,
                                         structure names are used as-is.
--imageCompareIntensityTolerance arg (=0)
                                         Voxels are considered identical if
                                         intensity difference is within this
                                         value. Does NOT affect MI, NMI, NCC,
                                         and RMSE calculations.
--imageCompareIntensityRadius arg (=0)
                                         When comparing voxel intensities,
                                         search this radius for a matching
                                         intensity value. Does NOT affect MI,
                                         NMI, NCC, and RMSE calculations.
--dump arg                            Directory to dump intermediate meshes
                                         and images to (optional)
-? [ --help ]                         Print all program options
--version                             Print program version information and
                                         exit.
```

12.6.3 Simple Use Case

In the following example, we have a planning CT that is already contoured (the "truth") and some contours on a followup daily CT that were created by ADMIRE's inpatient algorithm. The "truth" image and ROIs serves as the frame of reference. Therefore, the daily image must be warped into the planning CT's frame of reference. This can be done via ADMIRE's `--deform-patient-images` commandline option or "Deform patient into atlas frame" configuration setting in the GUI. This option applies the transformation and resamples the patient image into the atlas frame of reference.

For example, run ADMIRE to segment the daily image

```
> ADMIRE.exe --atlas c:\data\planningCT\rtss.dcm
--patient c:\data\dailyImage
--type gen
--ip one-to-all
--results c:\data\dailyImageResults
--gpu
--deform-patient-images
```

Now calculate the statistics in the atlas frame of reference, using the atlas's ROIs:

```
> ImageSimilarityUtil.exe --truthRTSS c:\data\planningCT\rtss.dcm
--truthImage c:\data\planningCT
--testImage c:\data\dailyImageResults\warpedImages
--csv c:\data\dailyResults.csv
Installing ITK logging adapter and enabling GlobalWarningDisplay
Initializing dmctk logging system
Installing DCMtkLoggingAdapter; prefiltering DCMtk to DEBUG
Loading truth RTSS
Loading specific truth image set
Using mappings file: mappings.txt
Converting 15 structures into surfaces, in-plane isospacing 1.000000
Created surface for PTV_5400 (11 contours) in 0.04s
Created surface for PENILE BULB (3 contours) in 0.04s
Created surface for BLADDER-PTV (15 contours) in 0.08s
Created surface for HOT1 (10 contours) in 0.03s
Created surface for RECTUM (40 contours) in 0.09s
Created surface for CTV_6200 (19 contours) in 0.04s
Created surface for BLADDER (15 contours) in 0.04s
Created surface for PTV_6200 (22 contours) in 0.10s
Created surface for Femoral Head_R (33 contours) in 0.11s
Created surface for SV (16 contours) in 0.10s
Created surface for HOT2 (5 contours) in 0.03s
Created surface for Femoral Head_L (34 contours) in 0.09s
Created surface for Planning_PTV_5400 (12 contours) in 0.04s
Created surface for External (45 contours) in 0.24s
Created surface for NT_2700 (73 contours) in 0.28s
Loading specific test image set
Calculating whole image stats
Calculating PTV_5400 stats
Calculating HOT1 stats
Calculating HOT2 stats
Calculating CTV_6200 stats
Calculating NT_2700 stats
Calculating Femoral Head_L stats
Calculating External stats
Calculating Femoral Head_R stats
Calculating SV stats
Calculating BLADDER-PTV stats
Calculating PENILE BULB stats
Calculating Planning_PTV_5400 stats
Calculating PTV_6200 stats
Calculating RECTUM stats
Calculating BLADDER stats
roi=BLADDER, mi=0.257273, nmi=1.071160, ncc=0.590954, rmse=423.895893
roi=BLADDER-PTV, mi=0.257662, nmi=1.071348, ncc=0.594796, rmse=423.256854
roi=CTV_6200, mi=0.513170, nmi=1.160727, ncc=0.773745, rmse=446.817551
roi=External, mi=0.816758, nmi=1.186000, ncc=0.950688, rmse=325.414560
roi=Femoral Head_L, mi=0.648156, nmi=1.158407, ncc=0.910872, rmse=394.194441
roi=Femoral Head_R, mi=0.683389, nmi=1.163980, ncc=0.904015, rmse=373.563516
roi=HOT1, mi=0.401199, nmi=1.121451, ncc=0.665774, rmse=477.217200
roi=HOT2, mi=0.138874, nmi=1.064432, ncc=0.458622, rmse=460.675086
roi=NT_2700, mi=0.816758, nmi=1.186000, ncc=0.950688, rmse=325.414560
roi=PENILE BULB, mi=0.121112, nmi=1.065637, ncc=0.572697, rmse=434.473300
roi=PTV_5400, mi=0.256765, nmi=1.074550, ncc=0.489539, rmse=473.939617
roi=PTV_6200, mi=0.547868, nmi=1.160554, ncc=0.772651, rmse=447.863829
roi=Planning_PTV_5400, mi=0.257024, nmi=1.074566, ncc=0.455034, rmse=478.791423
roi=RECTUM, mi=0.240270, nmi=1.068126, ncc=0.344244, rmse=470.299584
roi=SV, mi=0.236248, nmi=1.074021, ncc=0.355337, rmse=470.082709
roi=WHOLE_IMAGE, mi=0.764928, nmi=1.204981, ncc=0.950012, rmse=250.223221
Done
```

If you wish to calculate the metrics using the patient's ROIs instead, reverse the direction such that the atlas is warped onto the patient, and then swap the truth and test arguments.

```
> ADMIRE.exe --atlas "c:\data\planningCT\rtss.dcm"
--patient "c:\data\dailyImage"
--type gen
--ip one-to-all
--results "c:\data\dailyImageResults"
--gpu
```

```
--deform-atlas-images
> ImageSimilarityUtil.exe --truthRTSS c:\data\dailyImage\rtss.dcm
--testImage c:\data\planningCT
--truthImage c:\data\dailyImageResults\warpedImages
--csv c:\data\daily1results.csv
```

12.6.4 Interpreting Results

Opening the `c:\data\daily1results.csv` from the above example reveals the following columns:

- **TruthRTSS:** The `--truthRTSS` argument
- **TruthImage:** The `--truthImage` argument
- **TestImage:** The `--testImage` argument
- **ROIName:** Name of the ROI in the TruthRTSS
- **ROIMappedName:** Semantic name after mappings have been applied
- **MI:** Mutual information of the region (as implemented by `itk::MutualInformationHistogramImageToImageMetric`)
- **NMI:** Normalized mutual information of the region (as implemented by `itk::NormalizedMutualInformationHistogramImageToImageMetric`)
- **NCC:** Normalized cross-correlation (as implemented by `itk::NormalizedCorrelationImageToImageMetric`)
- **RMSE:** Root-mean squared error (as implemented by `itk::MeanSquaresImageToImageMetric`)
- **MinDiff,MaxDiff,MeanDiff:** Minimum, maximum, and average intensity difference found between two voxels (as implemented by `itk::TestingComparisonImageFilter`)
- **CumulativeDiff:** Cumulative sum of the intensity differences found between voxels (as implemented by `itk::TestingComparisonImageFilter`)
- **DiffCount:** Total number of voxels that have intensity differences (as implemented by `itk::TestingComparisonImageFilter`)
- **ROIx,ROIy,ROIz:** Upper left hand corner of the ROI bounding box in voxel coordinates in the TruthRTSS image.
- **ROIxs,ROIys,ROIzs:** Dimensions of the ROI bounding box in voxels

Each row corresponds to an ROI found in the TruthRTSS. A pseudo-ROI named "WHOLE_IMAGE" is also created that represents the entire image volume. If no `--truthRTSS` was provided, then only the WHOLE_IMAGE roi will be present.

12.6.5 Additional Options

Additional commandline options are below:

- `histogramBins`: by default, the mutual information statistics use 120 bins for the joint histogram. The bin ranges are calculated based on the intensity range of the `truthImage`.
- `roiPadding`: by default, the ROI bounding box is padded by 5 voxels in each dimension.
- `mappings`: The path to the ADMIRE mappings file. The "ROIMappedName" column in the results is for display purposes only to help the user normalize ROI names across multiple runs or different data sets.
- `dump`: If specified, each ROI will be exported (VTK format), as well as the `testImage` cropped to the bounding box (in ITK's `MetalImage` format). This is the volume over which the statistics are calculated.
- `imageCompareIntensityTolerance`: Two voxels are considered equal if their intensity difference is less than or equal to this tolerance. Default is 0. *NOTE: Does not affect MI, NMI, NCC or RMSE metrics.*
- `imageCompareIntensityRadius`: For each voxel in `truthImage`, the radius around the corresponding `testImage` voxel is searched for a matching value. Default is 0. *NOTE: Does not affect MI, NMI, NCC or RMSE metrics.*

12.6.6 Return Codes

Exits with return code 0 if the images are identical; 1 if there are any voxel differences; > 1 if there is any other processing error.

12.6.7 Misc notes

- At this time, `TruthImage` and `TestImage` **must have same origin, spacing, and direction** for the voxel equality statistics. The similarity metrics will still work, however.
- Any bounding boxes that extend outside the `truthImage` volume will be cropped
- If a bounding box exists totally outside the `truthImage` volume, an error message will be printed.
- If most of the bounding boxes are outside the `truthImage`, verify the `testImage` was provided and that the correct deformed image was exported from ADMIRE (e.g., `--deform-patient-images` vs `--deform-atlas-images`).
- All surface creation and DICOM image conversion is the same as used within ADMIRE.

12.7 ImageWarpUtil

Standalone tool for applying various spatial and deformable registrations to images.

12.7.1 Overview

A simple utility for applying DVFs and linear registrations to images.

12.7.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE> "C:\Program Files\ADMIRE\ImageWarpUtil.exe" --help
Image Warp:
  --src arg          Image to be warped. May be any supported ITK format
                     or DICOM directory
  --transform arg    Filename of the DICOM Deformable or Spatial
                     Registration
  --output arg       Output path to export warped image in an ITK format;
                     if the directory doesn't exist, it will be created
  --outputDcm arg    Output directory to export warped DICOM images; if it
                     doesn't exist, it will be created
  --outputFrameOfRef arg (optional) DICOM image set that defines the
                     destination frame of reference--size, origin,
                     spacing, etc.
  --seriesDescription arg Series Description for the warped DICOM images
                     (optional)
  -? [ --help ]      Print all program options
  --version           Print program version information and exit.
```

Apply `dreg.dcm` to the DICOM image set in `c:\patient`, and export a new DICOM image set to `c:\warped_patient`.

```
--src c:\patient --transform c:\patient\dreg.dcm --outputDcm c:\warped_patient
```

Apply a spatial registration object (SRO) `sreg.dcm` to the DICOM image set in `c:\patient`, and save the result as an ITK (MetaImageIO) image:

```
--src c:\patient --transform c:\patient\sreg.dcm --output c:\warped_patient\image.mha
```

Apply an ITK DVF to an ITK image and save the result as an ITK image (Niftii format):

```
--src c:\patient\patient.nii --transform c:\patient\warpedField.mha --output c:\patient\warpedImage.nii
```

12.7.3 Frame of Reference

A DICOM Deformable Spatial Registration object (DREG) defines a vector field in physical space. It is, in fact, an image, whose pixels happen to be vectors. There is nothing that requires this field to have the same origin, size, and spacing as the patient images. Therefore, when an image is warped, the resulting image is resampled into this frame. Usually, the DREG exported by ADMIRE is **smaller** than the original atlas image set (because it is computationally expensive to calculate at original resolution). Therefore, the warped image set would be smaller than the original atlas image.

If this is not desired, you can specify another image set's frame of reference to use via the `--outputFrameOfRef` option.

```
> --src c:\patient --transform c:\patient\dreg.dcm --output c:\warped_patient --outputFrameOfRef c:\atlas
```

The resulting image set will have the same origin, spacing, and dimensions as `c:\atlas`, instead of the vector field. Furthermore, the slice locations will be the same.

12.7.4 Notes and Limitations

- No DICOM consistency checking is done to verify that the DREG actually refers to the src image; if the resulting image is blank, this is likely the cause.
- Only the first vector grid in the Deformable Registration Sequence (0064,0002) is used; Pre- and Post- transformations (0064,000f) and (0064,0010) are currently ignored.
- If there are multiple transformations in the SRO, they are composed together as defined by [DICOM C.20.↵ 2.1.1](#)
- Outputting the warped image in DICOM format is only supported if the source image is DICOM.

12.8 MAFusionUtil

Standalone Multi-Atlas Fusion tool

12.8.1 Overview

A simple utility for running the final phase of a multi-atlas fusion case from ADMIRE. Currently only supports STAPLE and Multilabel STAPLE. Patch Fusion and Random Forest label fusion and other methods will be added later.

12.8.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE> "C:\Program Files\ADMIRE\MAFusionUtil.exe" --help
--staple <roi binning>          Uses STAPLE algorithm; ROI binning
                                values include:
                                refonly: Limits to ROIs found in
                                        reference rtss (the 1st --rtss
                                        argument)
                                intersection: Limits to ROIs found in
                                        every rtss
                                union: All ROIs, regardless if found in
                                        each rtss
--mstaple [=arg(=refonly)] (=refonly) Uses Multilabel STAPLE algorithm
--ref-image <file>              Reference image/dir (usually the novel
                                patient image).
--rtss <file>                   Input RTSS; Can be specified multiple
                                times
--output-rtss <file>           Output RTSS to contain the fused
                                results; will be overwritten if it
                                already exists
--structures <roi, ...>         Limits to specific ROIs (optional)
--xstructures <roi, ...>        Exclude specific ROIs (optional)
--mappings <file>              Structure mappings file (optional).
                                Maps structure names from both RTSSs
                                before comparing them. If unset,
                                structure names are used as-is.
--staple-stats <file>          Filename to store sensitivity,
                                specificity, likelihood statistics. The
                                extension determines the format
                                .csv: Comma-Separated Value
                                .json: JavaScript Object Notation
                                .txt: Tabular
--dump <dir>                   Directory to dump intermediate
                                representations; usually only for
                                debugging purposes
--param <file|param setting> ... Path to a parameters file which
                                overrides default parameters. Can also
                                override values directly using
                                "name=val" syntax. If specified
                                multiple times, overrides are applied
                                in order.
-? [ --help ]                  Print all program options
--version                       Print program version information and
                                exit.
```

12.8.3 Simple Use Case

3 different segmentations were created for the patient whose images reside in `c:\data\patient`. The sensitivity and specificity of each atlas will be written to `c:\data\stats.txt` in a tabular text format. The mappings file from ADMIRE is used to map ROI names.

```
> MAFusionUtil.exe --staple refonly
--rtss c:\data\result1.dcm
--rtss c:\data\result2.dcm
--rtss c:\data\result3.dcm
--ref-image c:\data\patient
--output-rtss c:\data\staple_result.dcm
--staple-stats c:\data\stats.csv
--mappings "C:\Program Data\ADMIRE\mappings.txt"
```

12.8.4 ROI Binning

There is no guarantee that each input RTSS has the exact same ROIs. After the mappings file has been applied, the binning method determines how ROIs will be grouped and which will be present in the final result.

- **intersection:** Only ROIs that appear in all inputs will be used. Each resulting ROI will come from the N inputs. If an ROI doesn't appear in all inputs, it is discarded.
- **union:** ROIs are grouped by name, regardless of how many results they appear in. Each resulting ROI will come from 1 - N inputs.
- **refonly:** The 1st input is considered the "reference". Only ROIs that appear in the reference input will appear in the final output. Each resulting ROI will come from 1 - N inputs.

Consider the following inputs RTSSs and the effect of each binning behavior on the final result. The number in parentheses represents how many inputs were used to obtain the result for that ROI.

Input								
rtss1.↔ dcm	PROSTATE	FEMUR↔ _LT	FEMUR↔ _RT		SEM_VES	GTV	CTV	
rtss2.↔ dcm	PROSTATE			PENILE↔ _BULB	SEM_VES	GTV	CTV	PELVIS
rtss3.↔ dcm	PROSTATE	FEMUR↔ _LT	FEMUR↔ _RT	PENILE↔ _BULB	SEM_VES	GTV	CTV	
-----	-----	-----	-----	-----	-----	-----	-----	-----
refonly	PROSTATE(3)	FEMUR↔ _LT(2)	FEMUR↔ _RT(2)		SEM_↔ VES(3)	GTV(3)	CTV(3)	
union	PROSTATE(3)	FEMUR↔ _LT(2)	FEMUR↔ _RT(2)	PENILE↔ _BULB(2)	SEM_↔ VES(3)	GTV(3)	CTV(3)	PELVIS(1)
intersection	PROSTATE(3)				SEM_↔ VES(3)	GTV(3)	CTV(3)	

12.8.5 Multilabel STAPLE

Because STAPLE operates on each ROI independently, there is nothing to prevent the resulting consensus ROIs from overlapping. Multilabel STAPLE (aka mStaple) addresses this by operating on all ROIs at the same time. However, this

comes at the expense of significantly increased computation time and memory usage. The process is roughly:

1. Convert N ROIs from K result RTSSs (aka "observations") into K*N binary masks
2. For each observation, combine binary masks into a single 3D label map image where each voxel is assigned a label number, 1..N where N is the number of ROIs. This results in K label maps—one for each observation.
3. Run mStaple algorithm. This results in a 4D image, where each voxel is a vector of size N, s.t. each element is the probability [0.0-1.0] the voxel belongs to label N.
4. Select the highest probability at each voxel and generate N probability map images. Each voxel in these 3D images represent the probability it is part of ROI N.
5. Create 3D meshes by taking the isosurface of each probability map at 0.5.
6. Slice mesh at original image slice locations.
7. Export final RTSS.

12.8.5.1 Example > MAFusionUtil.exe --mstaple
 --mappings "C:\Program Data\ADMIRE\mappings.txt"
 --rtss c:\allresults*.dcm
 --ref-image c:\data\patient
 --output-rtss c:\data\consensus.dcm
 --xstructures SKIN

12.8.5.2 Caveats

Each result should have same ROIs Because mStaple operates on all labels at the same time, each result (observation), should have the same ROIs. If an observation is missing an ROI, it's the same as assigning those voxels to BACKGROUND (label 0) for that result. If there are enough missing observations, the consensus result will be BACKGROUND. Regular STAPLE, because it operates on each ROI independently, can simply operate on fewer observations for that one ROI.

Pay attention to intentional ROI overlap Because mStaple operates on label maps, each voxel can only have a single label. As a result, consensus ROIs will not overlap—usually a desirable property. However, it is currently incompatible with ROIs that contain other ROIs. For example, consider the SKIN surface; for voxels that are labeled Not-SKIN, their probability it is Skin is 0.0. When the isosurface of the SKIN probability map is taken, the resulting surface will have "holes" where the voxels were assigned to another label. For now, use the `--xstructures` parameter to omit these ROIs.

For the same reason, if planning volumes (CTV, GTV, PTV, etc) are present, the resulting consensus ROIs will be donut/keyhole structures for all but the innermost volume.

No Statistics Regular STAPLE can output sensitivity and specificity metrics for each observation via `--staple-stats`. mStaple does not yet have these.

12.8.6 Misc Notes

- All masks and label maps are created at resolution of the `--ref-image` image set
- 1D and 2D structures are ignored

12.9 REGtoITK

Convert DICOM deformable spatial registration objects to ITK vector fields

12.9.1 Overview

The **DICOM Deformable Spatial Registration IOD** stores deformation vector fields as a 3D grid of vectors, where each vector is a displacement. The displacement values are stored as IEEE-754:1985 32-bit floating point format. Therefore, the Vector Grid Data (0064,0009) equates to a C-array of `float[Z][Y][X][3]`, where X,Y,Z are the dimensions of the grid. The Deformable Registration Sequence (0064,0002) also contains an optional Pre Deformation Matrix Registration Sequence (0064,000F) and Post Deformation Matrix Registration Sequence (0064,0010). These optional 4x4 transformations are meant to be composed with the vector grid by the consumer of the IOD.

This utility will extract all of the transformations. Any format supported by ITK can be used, assuming the format's capability to store N-dimensional pixel data. The output filename extension determines the format. Common ones are:

- **.mha:** **ITK Meta Image IO**
- **.mhd:** **ITK Meta Image IO header + separate .raw for values**
- **.nii:** **NIFTI**
- **.nrrd:** **Nearly Raw Raster Data**

Any present affine transformations will always be written in 3D Slicer's .tfm (WRML) format. Simply open the file in a text editor to view the values of the 4x4 matrix.

Note

See [this FAQ](#) to learn why the matrix values are different after being read into Slicer.

12.9.2 Example Usage

REGtoITK requires the DSRO filename as input, and an output filename.

```
C:\temp> "C:\Program Files\ADMIRE\REGtoITK.exe" --help
Copyright (c) 2008 - 2016 Elekta, Inc. Internal use only.
-----
Convert DICOM registration objects to ITK format:
--input arg          Filename of the DICOM transformation
--output arg         Filename to store the ITK result. The extension
                     determines the format.
-? [ --help ]        Print all program options
--version             Print program version information and exit.
```

A DSRO with only a vector grid:

```
C:\temp> "C:\Program Files\ADMIRE\REGtoITK.exe" --input DVFOnly.dcm --output DVFOnly.mha
Copyright (c) 2008 - 2016 Elekta, Inc. Internal use only.
-----
Loading transform .\DVFOnly.dcm
Wrote vector grid: /DVFOnly.mha
```

If pre- and post- affine transforms are present, a numeric suffix is appended to the filename so that the order of composition is apparent.

```
C:\> "C:\Program Files\ADMIRE\REGtoITK.exe" .\PreAndPost.dcm --output .\PreAndPost.nii
Copyright (c) 2008 - 2016 Elekta, Inc. Internal use only.
-----
Loading transform .\PreAndPost.dcm
Wrote preTransformation: ./PreAndPost0.tfm
Wrote vector grid: ./PreAndPost1.nii
Wrote postTransformation: ./PreAndPost2.tfm
```

12.10 StandardizeMR

Standalone StandardizeMR tool

12.10.1 Overview

Standardizes MR images by applying N4 bias field correction, 2D anisotropic diffusion, and median normalization. Implementation is based on the materials presented by H. Al Seleh in [Automation of MR Image Standardization Strategy for MRI-gRT at ASTRO 2015](#).

12.10.2 Basic usage

Like ADMIRE, the program should be run from ADMIRE's program data as the working directory.

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE> "C:\Program Files\ADMIRE\StandardizeMR.exe" --help
StandardizeMR 3.40.1 Copyright (c) 2008 - 2022 Elekta, Inc. For Investigational Use Only
Contact Dan.Thill@elekta.com for help
-----
StandardizeMR:
  --src arg                Source MR to Standardize; can be a directory (for
                           DICOM) or any ITK image file
  --output arg             ITK filename to export the standardized image to;
                           (e.g., standardized_mr.mha)
  --outputDCM arg         Directory export the standardized DICOM images;
                           if it doesn't exist, it will be created.
  --dump arg              (optional) dir. to dump the intermediate images
                           (in ITK format) at various stages; usually only
                           needed for debugging purposes
  --scaleMedianTo arg (=4096) Standardize by scaling the median of intensity
                           values to this new median
  --seriesDescription arg  Series Description for the new DICOM images
                           (optional)
  --windowCenter arg       (optional) sets DICOM Window Center value of
                           output; if unset, the median will be used
  --windowWidth arg        (optional) sets DICOM Window Width value of
                           output; if unset, entire intensity range is used
  --reorientToDicomRCS     If specified, the final image will be resampled
                           into the DICOM patient RCS; (i.e., direction
                           cosines will be 1 0 0 0 1 0); otherwise, the
                           original orientation is used
  --skipBiasFieldCorrection If specified, N4 bias field correction is not
                           done
```

```
--skipDenoising      If specified, anisotropic diffusion is skipped
--skipMedianScaling  If specified, the intensity values are not scaled
                    to the median specified in ---scaleMedianTo
-? [ --help ]        Print all program options
--version            Print program version information and exit.
```

Standardize the MR series in `c:\patient` and export a derived series to `c:\standardized`, setting a new series description.

```
StandardizeMR.exe --src c:\patient --outputDCM c:\standardized --seriesDescription "Standardized"
```

Standardize an MR series and save the result as an ITK image:

```
StandardizeMR.exe --src c:\patient --output c:\standardized\patient_standardized.mha
```

Standardize an MR that's already in ITK format:

```
StandardizeMR.exe --src c:\patient\subject.mha --output c:\standardized\subject_standardized.mha
```

Standardize an MR so that the median is scaled to 500 (instead of the default 4096):

```
StandardizeMR.exe --src c:\patient --outputDCM c:\standardized --scaleMedianTo 500
```

Standardize an MR, and override the default DICOM window center/width. If unset, the default window center is the median, and the width is the entire intensity range.

```
StandardizeMR.exe --src c:\patient --outputDCM c:\standardized --windowCenter 4000 --windowWidth 10000
```

Perform only bias field correction:

```
StandardizeMR.exe --src c:\patient --outputDCM c:\standardized --skipDenoising --skipMedianScaling
```

12.10.3 Implementation

By default, MRs are standardized by the following process:

1. **ITK N4 bias field correction**. 3 levels, each max of 50 iterations or convergence threshold of .001. Otsu thresholding is used to generate a foreground mask.
2. Slicewise anisotropic diffusion
3. Standardize intensity value range by scaling the median value to 4096

Each phase can be disabled by using `--skipBiasFieldCorrection`, `--skipDenoising`, `--skipMedianScaling` respectively. Lastly, the DICOM window center/width and Series Description can be tweaked.

12.10.4 Known Limitations

- Unless `--skipMedianScaling` is provided, the median intensity will be scaled to 4096 by default. This is for backwards compatibility with the Monaco+ADMIRE Research Interface.
- If a DICOM series is output, new series is created and as DERIVED and MR machine parameters are omitted
- If the input image is ITK, the output must also be ITK

13 Command Line Usage

Command line mode is provided to help integrate ADMIRE into existing workflows. At this time, all input and output centers on DICOM data on disk. The purpose of this document is to provide additional assistance as there are many options and methods of using the program. As such, it will primarily use examples. Whenever possible, conventions of UNIX command line programs were used.

13.1 Differences from GUI

Every attempt is made to keep feature parity between the GUI and command line. However, since the majority of users rely on the GUI, functionality is usually implemented there initially. However, the following are permanent differences:

- The GUI allows multiple cases to be queued and run in a batch. For the command line, only a single case can be created and run. Use the facilities of your scripting environment (Powershell, batch files, etc) to implement this batch behavior.
- The GUI provides a dialog for editing the structure mappings file. For the command line, the mappings file must be edited manually. As the GUI saves its settings in the same file, this should not be an issue. Optionally, you may edit the mappings file with text editor; the format is simple.
- In the GUI, the Error Reporting mechanism will immediately prompt to submit an error report if a crash occurs. In command line mode, the error report will be saved to disk. The next time the GUI is started, the user will have the option to submit unsent reports.

13.2 Setup

The executable must be run with the current working directory set to the application's command data location. This is because the program looks for certain resources in the current working directory (model files, licensing data, etc). While this is somewhat inconvenient for script writers, it also helps the program remain self-contained and not require registry settings or more invasive system configuration for it to locate its components. The installer will separate the executable and its runtime data into the conventional locations (e.g., `C:\Program Files\ADMIRE`, and `C:\Documents and Settings\All Users\Application Data\ADMIRE` respectively). However, it is possible to collect everything into a single directory for deployment purposes.

For example, on Windows 7, when running from `cmd.exe`:

```
C:\> cd C:\ProgramData\ADMIRE
C:\ProgramData\ADMIRE>"C:\Program Files\ADMIRE\ADMIRE.exe" --version
1.14.1.1
C:\>
```

Powershell is similar, but requires a leading ampersand:

```
PS C:\> cd C:\ProgramData\ADMIRE
PS C:\ProgramData\ADMIRE>&"C:\Program Files\ADMIRE\ADMIRE.exe" --version
1.14.1.1
PS C:\>
```

13.3 Basic Usage Examples

In general, defining a case requires an atlas, a patient, and a segmentation type (which loosely corresponds to an anatomical region). All of the following examples use the long form of the arguments. Short forms are also available for many arguments (e.g., `--type`, vs `-t`).

Head/Neck

```
ADMIRE.exe --type hn --atlas c:\a\demoHNatlas --patient c:\p\demoHNpat
```

Head/Neck using the GPU implementation

```
ADMIRE.exe --type hn --atlas c:\a\demoHNatlas --patient c:\p\demoHNpat --gpu
```

Head/Neck, but don't do any region-specific refinement.

```
ADMIRE.exe --type gen --atlas c:\a\demoHNatlas --patient c:\p\demoHNpat
```

Breast (there is no specific refinement available for this region)

```
ADMIRE.exe --type gen --atlas c:\a\breast1 --patient c:\p\patient1
```

Prostate, writing the resulting structure set to a different location:

```
ADMIRE.exe --type pr --atlas c:\a\demoPRatlas --patient c:\a\demoPRpat --results c:\output
```

13.4 Advanced Usage

13.4.1 Patient, Atlas paths

Paths are relative to the current working directory. Absolute paths are acceptable as well. If multiple patient series are in the same folder, the first series found will be the one used. To specify a specific series, use the path to a single image in the series you want. The Series UID will be extracted and only those images in the same series will be used.

13.4.2 Results

By default, the resulting structure set will be written to the patient input directory. Use the `--results` parameter to specify an alternate location. The filename is automatically generated and follows the same naming conventions as the GUI (e.g., filenames are incremented to avoid overwriting).

13.4.3 Structures

By default, the mappings file, `mappings.txt`, is searched for in the current working directory. This is the same file that the GUI stores its settings. `--mappings` argument can be used to specify a different file.

By default, all of the structures in the atlas will be present in the result. Use `--structures` to restrict to a subset. The structure names can either be the original names, or the mapped names.

13.5 STAPLE

STAPLE is a statistical method of combining the results from multiple segmentations into a single result. To define multiple atlases, specify `--atlas` multiple times. `--staple-stats` can be specified to output a text file with various statistics related to the STAPLE process. The file will be written to the same location as the structure set. At this time, the filename is automatically determined.

The first atlas parameter will be chosen as the *reference atlas*. The reference atlas determines which structures will be present in the final result. Any structures present in the other atlases that aren't in the reference atlas will be ignored. If a structure that is present in the reference atlas isn't found in all of the other atlases, the structure will be skipped with a warning message written to the log.

Head/Neck STAPLE, use the GPU segmentation algorithm.

```
ADMIRE.exe --type hn --patient c:\p\hnpatl1 --atlas c:\a\hn1 --atlas c:\a\hn2 --atlas c:\a\hn3
           --staple --staple-stats --gpu
```

Head/Neck STAPLE, only do eyes and skin, use hn2 as the reference atlas.

```
ADMIRE.exe --type hn --patient c:\p\hnpatl1 --atlas c:\a\hn2 --atlas c:\a\hn1 --atlas c:\a\hn3
           --staple --structures EYE_L, EYE_R, SKIN
```

13.6 Patch-Based Fusion

Also referred to as Patch Fusion or Intensity-Weighted Fusion, patch-based fusion is similar to STAPLE in that it combines multiple results to obtain a better segmentation. Unlike STAPLE, patch-based fusion uses image intensity information to weight the contributions from each result for each voxel. At this time, GPU support is required because all the images must be in the patient frame of reference. This requires inverting the deformation field calculated during normal segmentation. At this time, inverting the deformation field requires GPU support.

The `--patch-fusion` command line options behave the same way as STAPLE.

13.7 Random Forest Label Fusion

Like patch fusion, RF Label Fusion operates on ambiguous voxel labels by utilizing the intensity information from all the intermediate deformed atlas images. It classifies ambiguous voxels by training a random forest classifier on the fly.

The `--rflabel-fusion` command line options behave the same way as STAPLE.

13.8 Intrasubject and RCCT

Intrasubject and respiratory-correlated CT cases (`--ip` and `--rc` respectively) are designed for multiple patient image sets. The option values for these define the transitions between each segmentation.

13.8.1 one-to-all

The single `--atlas` is used as the atlas for all of the phases/fractions.

13.8.2 cascade

The `--atlas` is used as the atlas for the first phase/fraction, and the resulting segmentation is used as the atlas for the second phase/fraction. Then, the resulting structure set from the second, is used as the atlas for the third, and so on. The primary use for this method is for segmenting the phases of respiration.

13.8.3 one-to-many

The `--atlas` is used as the atlas for the first phase/fraction. The resulting structure set then used as the atlas for the remaining phases/fractions.

Prostate region, three fractions, each segmentation result is used as the atlas for the next.

```
ADMIRE.exe --type pr --atlas c:\p\patientRefCT --patient c:\p\patientFrac1
           --patient c:\p\patientFrac2 --patient c:\p\patientFrac3 --ip cascade --gpu
```

13.8.4 Single Case

At this time, to run an inpatient case with a single atlas and patient, you must still provide an argument, e.g., `one-to-all`, even though it is redundant and will be ignored. This will be fixed in a future release. An example of registering a patient's reference CT scan to a CT taken after fraction 1.

```
ADMIRE.exe --type gen --atlas c:\a\patientRefCT --patient c:\p\patientFraction1 --ip cascade
           --gpu --results d:\out
```

13.8.5 Adaptive Lung

To use the feature-constrained adaptive lung algorithm, use the `lung` region instead of `gen`. `--cbct` can also be appended if CBCT modality detection fails.

```
ADMIRE.exe --type lung --atlas c:\a\patientRefCT --patient c:\p\patientFraction1 --ip cascade
           --gpu --results d:\out
```


13.8.6 Constraints

Inpatient cases support using pre-existing structures as constraints to help guide the registration. For example, the bladder and prostate from fraction 1 might be used to constrain registration to fraction 2.

```
ADMIRE.exe --type gen --atlas c:\a\patientRefCT --patient c:\p\patientFraction1 --ip one-to-all
--constrain c:\p\frac1results.dcm --gpu --results d:\out
```

If multiple patients are part of the case, append "1", "2", etc to the option name to link the constraint to the proper patient.

```
ADMIRE.exe --type gen --atlas c:\a\patientRefCT --patient c:\p\patientFraction1
--patient c:\p\patientFraction2 --ip one-to-all --constrain1 c:\p\frac1results.dcm
--constrain2 c:\p\frac2results.dcm --gpu --results d:\out
```

To limit to specific ROIs within the constraint file, include them after the filename:

```
ADMIRE.exe --type gen --atlas c:\a\patientRefCT --patient c:\p\patientFraction1
--patient c:\p\patientFraction2 --ip one-to-all
--constrain1 c:\p\frac1.dcm PROSTATE BLADDER --constrain2 c:\p\frac2.dcm BLADDER
--gpu --results d:\out
```

13.9 Modalities

For the most part, it is not necessary to explicitly specify modalities of the image sets, as it can be determined from the DICOM data. However, not all segmentation types are valid with all modality combinations. When this occurs, the program will exit with an error. One exception is the `--cbct` parameter. For `--ip` and `--rc` segmentations, cone-beam CT images are often present. However, there is no 100% unambiguous way to identify them from the DICOM information. Therefore, use the `--cbct` to indicate that the patient images are cone-beam CT.

13.10 Combining Arguments

In general, a command line execution has four groups of options

- Configure the segmentation
- Define the transition from segmentation to segmentation
- Combine (STAPLE, patch fusion) the segmentations together
- Export information regarding the segmentations

For the most part, each category's options operate independently. For example, `--deform-patient-images` will output deformed patient images for every intermediate segmentation of an inpatient/RCCT case. Or, for basic case, it will export just one image set.

13.11 Output

Logging functionality is identical to GUI with the addition that the same information that would be written to the console log is also written to the STDOUT and STDERR streams. However, currently, there is problem redirecting these streams using standard shell redirections. Until this is fixed, all the same output is available in the most recent `main_2020XXXX.log`.

13.12 Exit Codes

- 0: No error; case completed successfully
- 1: Exception; case did not complete successfully. See log for details.
- 2: Not licensed for requested case type or feature

13.13 Cygwin

Passing in paths from cygwin will require using `cygpath` to translate cygwin paths to the original Windows path. Enclose the results in double quotes as `cygpath` will return paths with non-escaped spaces. For example:

```
"/cygdrive/c/Program Files/ADMIRE/ADMIRE.exe" \
-a "$(cygpath -aw "/path/to/atlas")" \
-p "$(cygpath -aw "/path/to/patient")" \
-r "$(cygpath -aw "/path/to/resultsdir")" \
--type gen
```

13.14 Other ADMIRE Binaries

`ADMIRE.exe` is compiled as Win32 Console Application. If started with no commandline arguments, the GUI loads and frees the console window. This is why a console window is briefly visible when starting from a shortcut or the Start Menu. Two other binaries are provided:

- `ADMIREc.exe` is a command line mode only executable. It has no .NET dependencies or GUI components. It is compiled as a Win32 Console Application. It has a smaller memory footprint and may have faster startup times because the .NET runtime is not needed. It is intended for service-type or scripting-only deployments.
- `ADMIREcw.exe` is the same as `ADMIREc.exe`, but compiled as a standard Win32 application. It has no standard input/output streams, but it can be useful in certain scripting environments where having an attached console for any reason is problematic.

	ADMIRE.exe	ADMIREc.exe	ADMIREcw.exe
Language	C++/CLI	C++	C++
Entry-point	main()	main()	wWinMain()
Supports commandline params	✓	✓	✓
Attached Console	✓	✓	
Standard streams	✓	✓	
GUI	✓		
.NET Runtime dependency	✓		

13.15 Parameter Reference

This can be seen at any time by using the `--help` parameter.

```
*****
*                               ADMIRE                               *
*                               Version 3.28                         *
*                               Copyright (c) 2008 - 2021           *
*****
```

Allowed options:

<code>-t [--type] <opt></code>	Segmentation type to perform. Valid values are: gen: Normal deformable registration with no additional anatomy-specific refinements. Suitable for any region (but optimized for none) hn: Deformable registration with refinements for the head/neck region. pr: Deformable registration with refinements for the prostate region. lung: Deformable registration with refinements for the lung region (intrapatient only). brain: MR brain segmentation demons: Use demons algorithm (not recommended) dl_autoseg: Deep learning autosegmentation (atlas determines region) dl_synimage: Deep Learning synthetic image creation; atlas determines anatomical region and output modality; subject is input modality
<code>-p [--patient] <dir></code>	Path to one or more patient folders or a single patient image file. An image file name is required when there are multiple image series in the same folder. If there are multiple image series in the same folder and only the folder name is provided, the first image series will be chosen.
<code>-a [--atlas] <dir></code>	Path to atlas folder. Must be specified multiple times when using a fusion algorithm to combine results.
<code>--staple</code>	Uses STAPLE algorithm to combine results from multiple atlases. Requires the <code>--atlas</code> option to be specified for each atlas. The first <code>--atlas</code> option will be used as the reference atlas.
<code>--patch-fusion</code>	Uses patch fusion (a.k.a. intensity weighted fusion) algorithm to combine results from multiple atlases (similar to STAPLE).
<code>--rflabel-fusion</code>	Uses random forest label fusion algorithm to combine results from multiple atlases (similar to STAPLE).
<code>--staple-stats</code>	Export statistics resulting from a segmentation using <code>--staple</code> .
<code>-r [--results] <dir></code>	Path to the results folder location. If this option is omitted, the patient image folder is used.
<code>-m [--mappings] <file></code>	Path to a structure name mappings file. If none is specified, mappings.txt will be searched for in current working directory.
<code>--param <file param setting> ...</code>	Path to a parameters file which overrides default parameters. Can also override values directly using "name=val" syntax. If specified multiple times, overrides are applied in order.
<code>--merge-unique-structures <dicomfile></code>	Path to an existing structure set to merge into the final results. In the

	event of a structure name collision, the structure from the pre-existing structure set is kept. Can be specified multiple times if the --rc option is in effect. When in this mode, the existing structure sets are associated automatically with phases by correlating the study and series instance UIDs.
--merge-all-structures <dicomfile>	Path to an existing structure set to merge into the final results. In the event of a structure name collision, the structure from the new structure set is renamed with a suffix (e.g., PROSTATE_1 -> PROSTATE_2).
-s [--structures] <roi> ...	Restricts segmentation to this list of structure names. Separate each structure name with a space or comma.
-g [--gpu]	Use GPU acceleration where applicable. If no supported GPU hardware is present, an error will occur.
--cbct	Forces patient CT images to be treated as CBCT. Use when heuristics fail to detect CBCT.
-I [--ip] <opt>	Method of processing inpatient series. Valid values are:
	<ul style="list-style-type: none"> one-to-all: The atlas will be used to segment each series independently. cascade: The atlas will be used to segment the first provided series. The results of that segmentation will be as an atlas for segmenting the second series. The results from segmenting each remaining series will be used with that series's images as an atlas in the segmentation of the next series. one-to-many: The atlas will be used to segment the first provided series. The results of that segmentation will be used as an atlas for the remaining series.
-R [--rc] <opt>	Method of processing respiration-correlated phases. Requires the --phases option to be specified. Valid values are:
	<ul style="list-style-type: none"> one-to-all: The atlas will be used to segment each series independently. cascade: The atlas will be used to segment the first provided series. The results of that segmentation will be as an atlas for segmenting the second series. The results from segmenting each remaining series will be used with that series's images as an atlas in the segmentation of the next series. one-to-many: The atlas will be used to segment the first provided series. The results of that segmentation will be used as an atlas for the remaining series.
-P [--phases] <opt>	Specifies 4D phases in the order they will be processed. Separate each phase with a space. Requires the --rc option to be specified.
--constrain <dicomfile> [<roi> ...]	Path to a structure set of existing patient structures which will be used to constrain the segmentation. Currently only valid for inpatient.

May be either specified multiple times or linked to a specific patient in a multi-patient case by appending 1, 2, etc. I.e., '--constrain2 patient.dcm'. Limiting to specific ROIs is also valid: '--constrain2 patient.dcm BLADDER PROSTATE'

--skip-linear-reg Skip linear/rigid registration; requires there be an implicit registration between atlas and patient. Use with caution.

Other outputs:

--deform-patient-images Apply deformation field to patient image. This has the effect of resampling the patient onto the atlas frame of reference. Exports the result as a new DICOM image set. Generally, this only makes sense for inpatient cases.

--deform-atlas-images Apply deformation field to atlas image. This has the effect of resampling the atlas onto the patient frame of reference. Exports the result as a new DICOM image set. Generally, this only makes sense for inpatient cases.

--transform-patient-images Apply linear registration to patient image. This has the effect of resampling the patient onto the atlas frame of reference. Exports the result as a new DICOM image set.

--transform-atlas-images Apply linear registration to atlas image. This has the effect of resampling the atlas onto the patient frame of reference. Exports the result as a new DICOM image set.

--keep-empty-result-structures If true, deformed structures that map outside the patient FOV or do not intersect any patient image slices will be included in the results as an ROI with no contours. Otherwise, by default, these empty structures are filtered out.

--save-sreg [=<opt> (=associate_with_patient)] Save spatial registration (linear registration) in DICOM format

--save-inv-sreg [=<opt> (=associate_with_patient)] Save inverse spatial registration (linear registration) in DICOM format

--save-dreg [=<opt> (=associate_with_patient)] Save deformable spatial registration (DVF) in DICOM format

--save-inv-dreg [=<opt> (=associate_with_patient)] Save inverse deformable spatial registration (DVF) in DICOM format

--save-dreg-raw Save deformable spatial registration (DVF) in ITK format

--save-inv-dreg-raw Save inverse deformable spatial registration (DVF) in ITK format

--save-rtss Save contours as RT Structure Set. Defaults to enabled; use 'no' value to disable.

--save-intermediate-results Save intermediate atlases, DVFs, etc created during multi-atlas cases (STAPLE, patch fusion)

Miscellaneous:

--gpu-info Display information on all available GPU devices on the system.

-@ [--response-file] <file> Read additional options from this text file as if they were part of the original commandline. This is required if the total commandline length becomes too large (typically, 8K chars).

--version Display version number and exit.

-? [--help] Display this usage message and exit.

14 DICOM Notes

Miscellaneous notes regarding DICOM compatibility and usage.

As a rule, ADMIRE tries to be as permissive as is reasonable when reading DICOM files. Violations and ambiguities with the standard are corrected whenever possible as long as referential integrity can be preserved. Any DICOM files written by ADMIRE, however, try to adhere to the generally stricter [IHE-RO standards](#).

14.1 Cone-Beam CT

At this time, there is no standard method to differentiate Cone-Beam CT (CBCT) images from CT images. There exists a [Correction Proposal](#) that proposes defining a new term for Value 4 of the Image Type (0008,0008) tag with the value of CONEBEAM. ADMIRE implements this proposal when reading/writing CBCT images.

Additionally, a heuristic is employed based on the Manufacturer (0008,0070) tag. If a CT image comes from manufacturer with ELEKTA, SIEMENS CBCT, SIEMENS MV CBCT, or Varian Imaging Laboratories, Switzerland the image is treated as CBCT because these manufacturers do not sell regular CT scanners.

If this heuristic is not valid for your data, use the *Adaptive (CBCT)* algorithm type to force CT data to be treated as CBCT. We are looking to add user-definable heuristics in a future release.

14.2 ROI Interpreted Types

ADMIRE uses the ROI Interpreted Type (3006,00a4) values from the atlas when writing out the new structure set. If it's marked as EXTERNAL in the atlas, the resulting structure will be marked as EXTERNAL.

If there is no ROI Interpreted Type for the structure in the atlas, ADMIRE uses the mappings file to set any structure that maps to "SKIN" as EXTERNAL. That is the only interpreted type ADMIRE attempts to generate on its own. This is because many other products which may consume ADMIRE's results require any skin ROI to have an Interpreted Type of EXTERNAL.

14.3 Spatial Registration

Frame of Reference Transformation Matrix Type (0070,030c) is always AFFINE, even if intra-subject segmentation is performed which uses rigid registration.

14.4 ROI Contour Sequence - Contour Image Sequence

Within the ROI Contour Sequence (3006,0039), each contour in the Contour Sequence (3006,0040), has an optional Contour Image Sequence (3006,0016) which stores a reference to the Class and Instance UID of the image the contour was drawn upon. IHE-RO states that each contour shall be associated with an image. However, for point/marker ROIs which consist of a single point, after deformation, it is most likely the transformed point will not end up on a slice. Therefore, single point/marker ROIs will not have Contour Image Sequence item.

15 Differences from Product Version

Outlines differences between the research and production versions

15.1 Overview

ADMIRE is a research platform/product for automatic segmentation and registration. It is not certified for clinical use. It was previously called ABASProto8 (the 8th ABAS prototype). To avoid confusion with the Elekta ABAS product which is commercially sold, supported, and FDA-approved, it was renamed ADMIRE. It primarily serves to investigate new algorithms with research collaborators. Ultimately, many of the features within the research version appear in the product version. This document serves to briefly cover the features in the research version which are *not* present in the production version. For detailed version-to-version change notes within ADMIRE, please refer to the [Changelog](#).

15.2 Algorithms

- Inpatient
- [Patch fusion](#)
- Random Forest Label Fusion

15.3 Workflows

- Multi-series processing for inpatient and respiratory-correlated CT (RCCT). Simply put, it allows segmentation of a fraction/phase to be propagated to other fractions/phases.
 - *One-to-all*: same atlas is applied to each fraction/phase independently
 - *Cascade*: atlas applied to first fraction/phase, result is used as atlas for second fraction/phase, and so on
 - *One-to-many*: atlas applied to first fraction/phase, result is used as atlas for remaining fraction/phases
- Deforming atlas or patient image set with deformation field onto the frame of reference of the other (results are exported as a new DICOM image series).

15.4 DICOM

- affine/rigid spatial registration export
- [Deformable Spatial Registration export](#)

15.5 Misc

- Improved support for multiple cores/CPU's
- Improved commandline capabilities

16 Windows 7 Notes

Notes on ADMIRE deployment on Windows 7.

16.1 Displaying the ProgramData folder

On Windows 7, the configuration file, mappings file, log files, licensing-related files, demo atlases, and demo patients are all located in `C:\ProgramData\ADMIRE`. This folder is hidden by default in Windows Explorer. There are two ways to access it. One method is permanent and affects the display of all file system elements, while the other is a temporary way to access only a folder within the `ProgramData` subtree.

16.1.1 Permanently Show

Permanently show all hidden file system elements.

- To show all hidden file system elements, click the Organize button in Windows Explorer and select Folder and Search Options from the menu

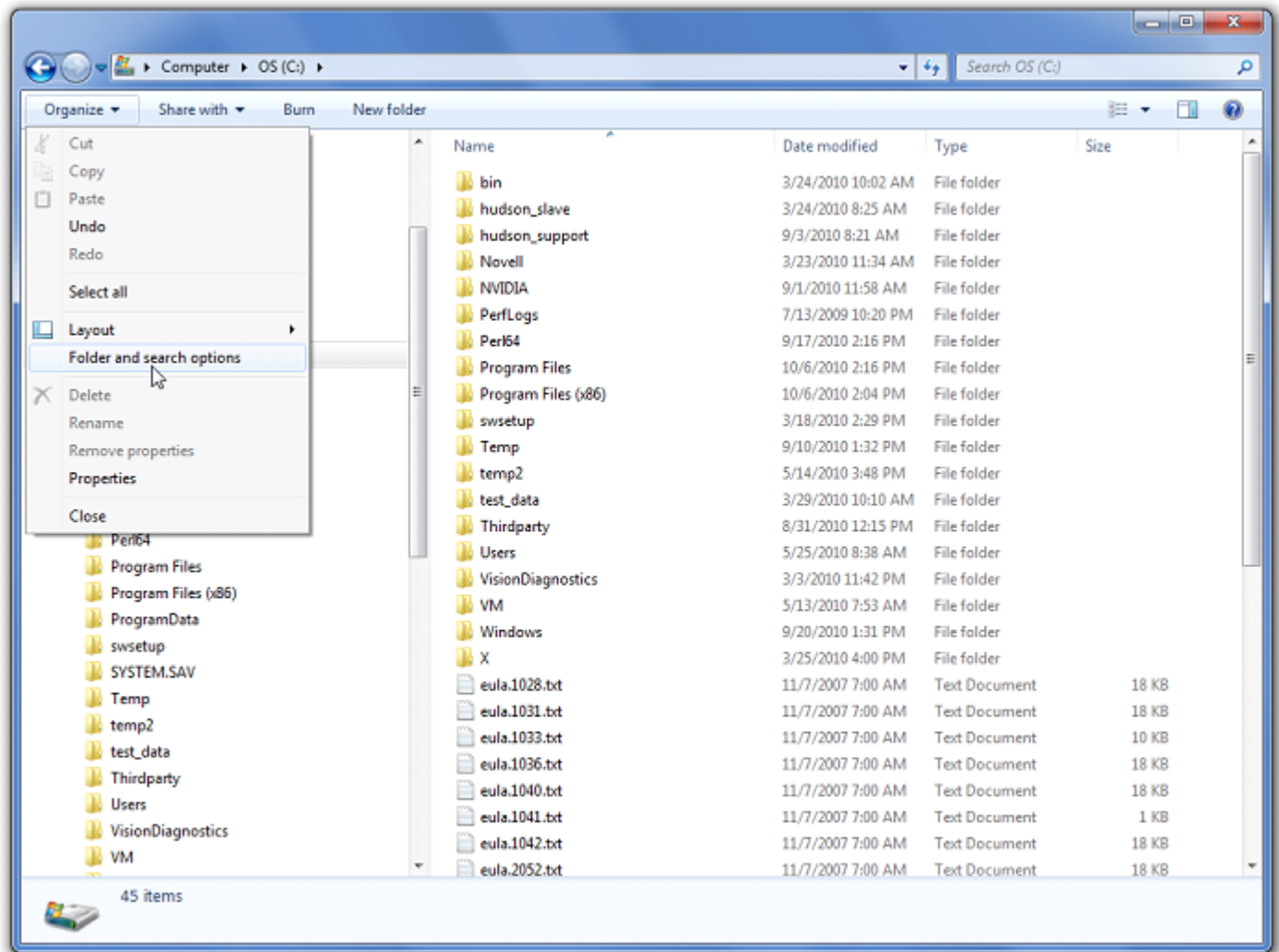


Figure 10 Folder options

- Click the View tab in the dialog that appears, then select the Show hidden files, folders, and drives radio button that is in the Advanced settings list box:

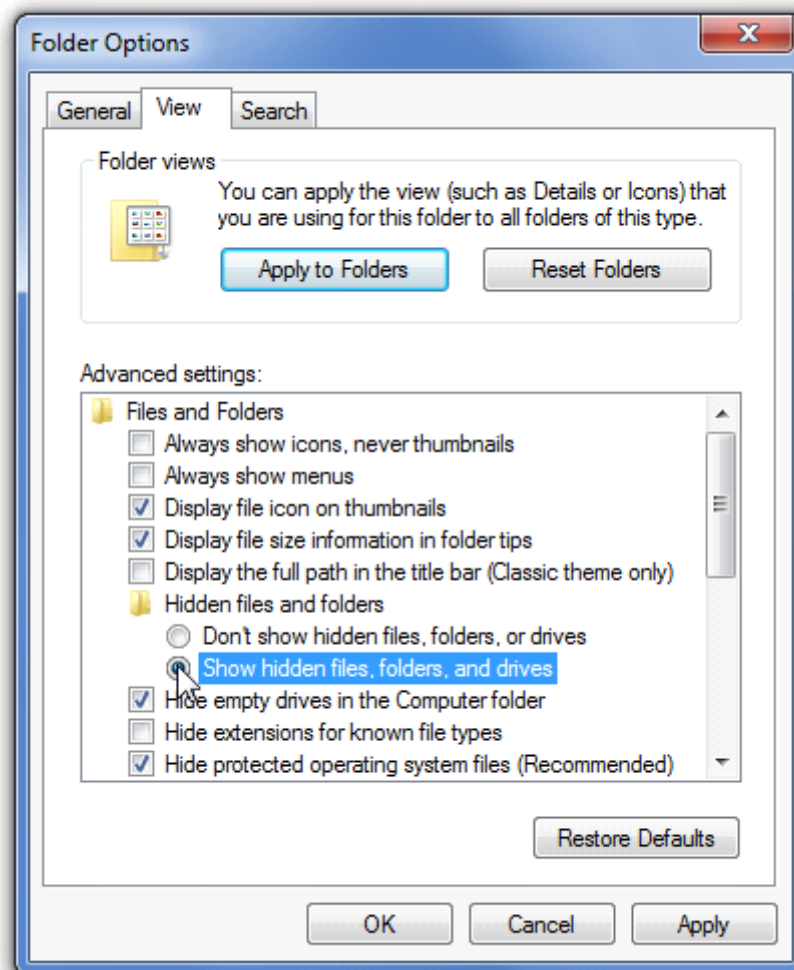


Figure 11 Show hidden files

- Click OK and the ProgramData directory should be visible in the main folder list:

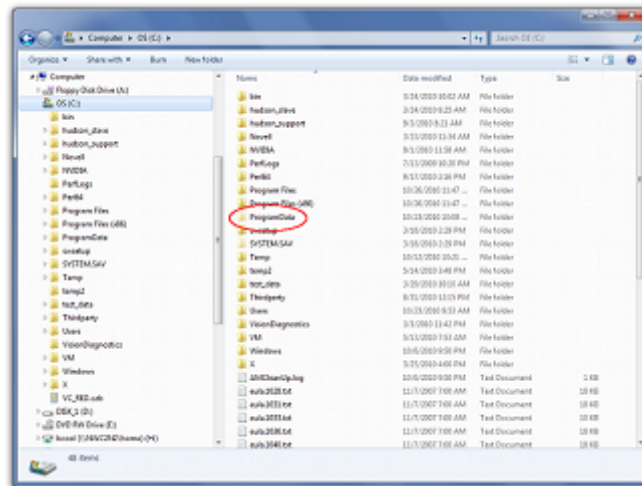


Figure 12 ProgramData

16.1.2 Temporary

Temporarily show the contents of the ProgramData folder without affecting other file system elements by entering the path in the address bar.

- Instead of displaying all hidden items, directly type the path `C:\ProgramData` in the address bar. To do this, click in the blank area on the right side of the address bar to edit the current address:

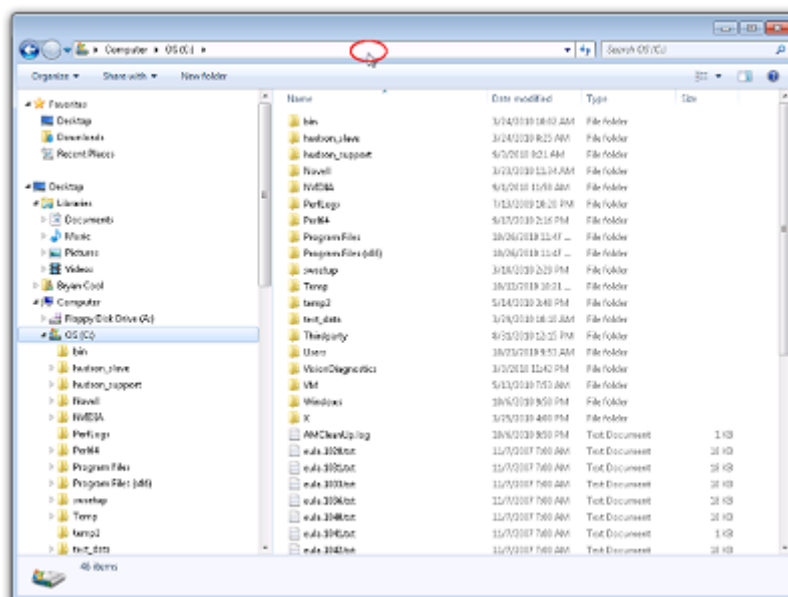


Figure 13 Entering path manually

- The address bar will change to display the full path. Type `C:\ProgramData` into the address bar:

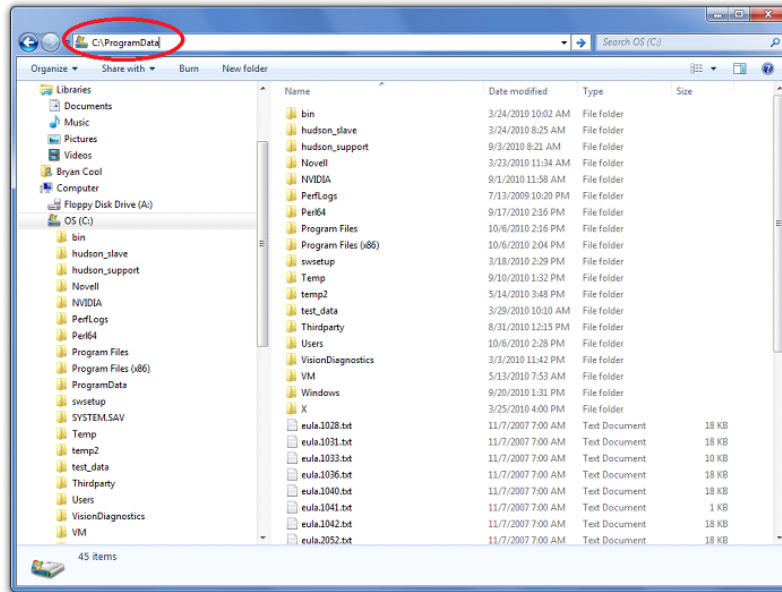


Figure 14 Entering path manually

- Hit Enter to go directly to the ProgramData folder, or type the rest of the desired path.

Or simply use the "Configuration Data" shortcut that's installed into the ADMIRE folder in the Start Menu and the File Menu within ADMIRE.

16.2 Plug-in Location

ADMIRE plug-ins are located in the application directory, which is by default `C:\Program Files\ADMIRE`.

17 Whitepapers & References

The following papers describe the algorithms implemented by ADMIRE. Click on the [pdf] link to open the paper in your default PDF application. The functional areas of ADMIRE covered by the paper are listed immediately following. Refer to [Segmenter Detail](#) page for the full enumeration of ADMIRE's capabilities with respect to modality and computing hardware.

17.1 Elekta

- *Feature-constrained Nonlinear Registration of Pulmonary CT Images* [[pdf](#)]
 - Inpatient Lung CT & CBCT (GPU-only)
- *GPU-accelerated, Gradient-free MI Deformable Registration for Atlas-based MR Brain Image Segmentation* [[pdf](#)]

- Inter-patient CT-CT, MR-MR (GPU-only)
- *Atlas-based Auto-segmentation of CT Images for Radiotherapy Planning* [[pdf](#)]
 - Inter-patient CT-CT, MR-MR (GPU-only) (specifically head/neck region)
- *Intra-Patient Deformable Image Registration for Adaptive Radiotherapy* [[pdf](#)]
 - Intra-patient CT-CT, CT-CBCT, CBCT-CBCT, CBCT-CT, MR-MR (same sequence)
- *An Efficient Inverse-Consistent Diffeomorphic Image Registration Method for Adaptive Radiotherapy* [[pdf](#)]
 - Intra-patient CT-MR, MR-CT
- *Intensity-weighted Multi-Atlas Label Fusion for ADMIRE* [[pdf](#)]
 - Multi-atlas cases created using "Patch Fusion" setting
- *ADMIRE with Multiple Atlases - The STAPLE Approach* [[pdf](#)]
 - Multi-atlas cases created using "STAPLE" setting
- *Learning-boosted Multi-Atlas Label Fusion for Atlas-based Auto-segmentation* [[pdf](#)]
 - Multi-atlas cases created using "RF Label Fusion" setting
- *Surface Reconstruction from a Stack of Planar Contours* [[pdf](#)]
 - All DICOM structure sets used as atlases or constraints
- *Automatic Image Segmentation using Deep Convolutional Neural Networks* [[pdf](#)]
 - [Deep learning based autosegmentation](#)
- *Synthetic CT modules/models for Cone-beam CT images using CycleGAN-based framework* [[pdf](#)]
 - [Synthetic CT](#)

17.2 Non-Elekta

- Tom Vercauteren et al. *Deformation field validation and inversion applied to adaptive radiation therapy* 2013 Phys. Med. Biol. 58 5269. [[http](#)]
 - Inverse trilinear interpolation DVF inversion method
- Warfield SK, Zou KH, Wells WM. *Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation*. IEEE Trans Med Imaging. 2004 Jul;23(7):903-21. [[http](#)].
 - Multilabel STAPLE implementation
- Haisen S Li et al. *Direct dose mapping versus energy/mass transfer mapping for 4D dose accumulation: fundamental differences and dosimetric consequences*. 2014 Phys. Med. Biol. 59 173. [[http](#)]
 - [Dose Warping](#)
- Ziegenhein, P., Kamerling, C.P., Fast, M.F. et al. *Real-time energy/mass transfer mapping for online 4D dose reconstruction*. Sci Rep 8, 3662 (2018). [[http](#)]
 - [Dose Warping](#)

17.3 Comparison and Performance Studies

Elekta partners with various academic and health institutions to help develop and validate the algorithms contained within ADMIRE.

- Ying Suna, Xiao-Li Yua, Wei Luo, et al. *Recommendation for a contouring method and atlas of organs at risk in nasopharyngeal carcinoma patients receiving intensity-modulated radiotherapy*. Radiotherapy of NPC. Volume 110, Issue 3, p390-397, March 01, 2014. [[http](#)]
- Amjad, A, Xu, J, Thill, D, et al. *General and custom deep learning autosegmentation models for organs in head and neck, abdomen, and male pelvis*. Med Phys. 2022; 49: 1686-1700. [[http](#)]

18 Licenses and Acknowledgments

License acknowledgments for third party software used in ADMIRE.

18.1 Tensorflow

Copyright 2018 The TensorFlow Authors. All rights reserved.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or

Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of

the Derivative Works; and

- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

- 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
- 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
- 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
- 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
- 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify,

defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

18.2 ADV

Written by Sebastien Andrivet - Copyright (C) 2010-2017 Sebastien Andrivet.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

18.3 Botan

Copyright (C) 1999-2018 The Botan Authors
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

18.4 DCMTK

```
/*
 * Copyright (C) 1994-2004, OFFIS
 *
 * This software and supporting documentation were developed by
 *
 * Kuratorium OFFIS e.V.
 * Healthcare Information and Communication Systems
 * Escherweg 2
 * D-26121 Oldenburg, Germany
 *
 * THIS SOFTWARE IS MADE AVAILABLE, AS IS, AND OFFIS MAKES NO WARRANTY
 * REGARDING THE SOFTWARE, ITS PERFORMANCE, ITS MERCHANTABILITY OR
 * FITNESS FOR ANY PARTICULAR USE, FREEDOM FROM ANY COMPUTER DISEASES OR
 * ITS CONFORMITY TO ANY SPECIFICATION. THE ENTIRE RISK AS TO QUALITY AND
 * PERFORMANCE OF THE SOFTWARE IS WITH THE USER.
 *
 * Copyright of the software and supporting documentation is, unless
 * otherwise stated, owned by OFFIS, and free access is hereby granted as
 * a license to use this software, copy this software and prepare
 * derivative works based upon this software. However, any distribution
 * of this software source code or supporting documentation or derivative
 * works (source code and supporting documentation) must include the
 * three paragraphs of this copyright notice.
 */
```

```
/*
 * Copyright (C) 1993/1994, OFFIS, Oldenburg University and CERIU
 *
 * This software and supporting documentation were
 * developed by
 *
 * Institut OFFIS
 * Bereich Kommunikationssysteme
 * Westerstr. 10-12
 * 26121 Oldenburg, Germany
 *
 * Fachbereich Informatik
 * Abteilung Prozessinformatik
 * Carl von Ossietzky Universitaet Oldenburg
 * Ammerlaender Heerstr. 114-118
 * 26111 Oldenburg, Germany
 *
 * CERIU
 * Laboratoire SIM
 * Faculte de Medecine
 * 2 Avenue du Pr. Leon Bernard
 * 35043 Rennes Cedex, France
 *
 * for CEN/TC251/WG4 as a contribution to the Radiological
 * Society of North America (RSNA) 1993 Digital Imaging and
 * Communications in Medicine (DICOM) Demonstration.
 *
 * THIS SOFTWARE IS MADE AVAILABLE, AS IS, AND NEITHER OFFIS,
 * OLDENBURG UNIVERSITY NOR CERIU MAKE ANY WARRANTY REGARDING
 * THE SOFTWARE, ITS PERFORMANCE, ITS MERCHANTABILITY OR
 * FITNESS FOR ANY PARTICULAR USE, FREEDOM FROM ANY COMPUTER
 * DISEASES OR ITS CONFORMITY TO ANY SPECIFICATION. THE
 * ENTIRE RISK AS TO QUALITY AND PERFORMANCE OF THE SOFTWARE
 * IS WITH THE USER.
 *
 * Copyright of the software and supporting documentation
 * is, unless otherwise stated, jointly owned by OFFIS,
 * Oldenburg University and CERIU and free access is hereby
```

```
* granted as a license to use this software, copy this
* software and prepare derivative works based upon this
* software. However, any distribution of this software
* source code or supporting documentation or derivative
* works (source code and supporting documentation) must
* include the three paragraphs of this copyright notice.
*
*/
```

18.5 VTK

```
/*=====

Program:   Visualization Toolkit
Module:    $RCSfile: Copyright.txt,v $

Copyright (c) 1993-2006 Ken Martin, Will Schroeder, Bill Lorensen
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names
  of any contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

* Modified source versions must be plainly marked as such, and must not be
  misrepresented as being the original software.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS''
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====*/
```

18.6 ITK

```
Copyright (c) 1999-2003 Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
```

and/or other materials provided with the distribution.

- * The name of the Insight Software Consortium, or the names of any consortium members, or of any contributors, may not be used to endorse or promote products derived from this software without specific prior written permission.
- * Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

18.7 Elastix

Copyright (c) 2004-2010 University Medical Center Utrecht
All rights reserved.

License:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the University Medical Center Utrecht nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

18.8 ITK

Copyright (c) 2006 Instituto Tecnológico de Canarias - Gobierno de Canarias
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * The name of the Instituto Tecnológico de Canarias - Gobierno de Canarias, may be used to endorse or promote products derived from this software without specific prior written permission.
- * Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

18.9 Boost

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

18.10 ZLib

Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly Mark Adler
jloup@gzip.org madler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files <http://www.ietf.org/rfc/rfc1950.txt> (zlib format), [rfc1951.txt](http://www.ietf.org/rfc/rfc1951.txt) (deflate format) and [rfc1952.txt](http://www.ietf.org/rfc/rfc1952.txt) (gzip format).

18.11 libpng

```
/*
 * COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:
 *
 * If you modify libpng you may insert additional notices immediately following
 * this sentence.
 *
 * This code is released under the libpng license.
 *
 * libpng versions 1.2.6, August 15, 2004, through 1.4.3, June 26, 2010, are
 * Copyright (c) 2004, 2006-2010 Glenn Randers-Pehrson, and are
 * distributed according to the same disclaimer and license as libpng-1.2.5
 * with the following individual added to the list of Contributing Authors:
 *
 *   Cosmin Truta
 *
 * libpng versions 1.0.7, July 1, 2000, through 1.2.5, October 3, 2002, are
 * Copyright (c) 2000-2002 Glenn Randers-Pehrson, and are
 * distributed according to the same disclaimer and license as libpng-1.0.6
 * with the following individuals added to the list of Contributing Authors:
 *
 *   Simon-Pierre Cadieux
 *   Eric S. Raymond
 *   Gilles Vollant
 *
 * and with the following additions to the disclaimer:
 *
 *   There is no warranty against interference with your enjoyment of the
 *   library or against infringement.  There is no warranty that our
 *   efforts or the library will fulfill any of your particular purposes
 *   or needs.  This library is provided with all faults, and the entire
 *   risk of satisfactory quality, performance, accuracy, and effort is with
 *   the user.
 *
 * libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
 * Copyright (c) 1998, 1999, 2000 Glenn Randers-Pehrson, and are
 * distributed according to the same disclaimer and license as libpng-0.96,
 * with the following individuals added to the list of Contributing Authors:
 *
 *   Tom Lane
 *   Glenn Randers-Pehrson
 *   Willem van Schaik
 *
 * libpng versions 0.89, June 1996, through 0.96, May 1997, are
 * Copyright (c) 1996, 1997 Andreas Dilger
 * Distributed according to the same disclaimer and license as libpng-0.88,
 * with the following individuals added to the list of Contributing Authors:
```

```

*
*   John Bowler
*   Kevin Brace
*   Sam Bushell
*   Magnus Holmgren
*   Greg Roelofs
*   Tom Tanner
*
* libpng versions 0.5, May 1995, through 0.88, January 1996, are
* Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.
*
* For the purposes of this copyright and license, "Contributing Authors"
* is defined as the following set of individuals:
*
*   Andreas Dilger
*   Dave Martindale
*   Guy Eric Schalnat
*   Paul Schmidt
*   Tim Wegner
*
* The PNG Reference Library is supplied "AS IS". The Contributing Authors
* and Group 42, Inc. disclaim all warranties, expressed or implied,
* including, without limitation, the warranties of merchantability and of
* fitness for any purpose. The Contributing Authors and Group 42, Inc.
* assume no liability for direct, indirect, incidental, special, exemplary,
* or consequential damages, which may result from the use of the PNG
* Reference Library, even if advised of the possibility of such damage.
*
* Permission is hereby granted to use, copy, modify, and distribute this
* source code, or portions hereof, for any purpose, without fee, subject
* to the following restrictions:
*
* 1. The origin of this source code must not be misrepresented.
*
* 2. Altered versions must be plainly marked as such and
* must not be misrepresented as being the original source.
*
* 3. This Copyright notice may not be removed or altered from
* any source or altered source distribution.
*
* The Contributing Authors and Group 42, Inc. specifically permit, without
* fee, and encourage the use of this source code as a component to
* supporting the PNG file format in commercial products. If you use this
* source code in a product, acknowledgment is not required but would be
* appreciated.
*/

```

18.12 Thrust

```

/*
 * Copyright 2008-2012 NVIDIA Corporation
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
*/

```

18.13 Segmentation Evaluation

<https://github.com/codalab/EvaluateSegmentation>

```
// VISERAL Project http://www.viceral.eu
// VISCERAL received funding from EU FP7, contract 318068
// Created by Abdel Aziz Taha (taha@ifs.tuwien.ac.at)
// on 20.03.2013
// Copyright 2013 Vienna University of Technology
// Institute of Software Technology and Interactive Systems
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
```

19 Data Structure Index

19.1 Data Structures

Here are the data structures with brief descriptions:

CmsAlgorithm::SurfaceComparisonStatistics
Simple container for all comparison statistics

114

20 File Index

20.1 File List

Here is a list of all documented files with brief descriptions:

C:/wa/admire/BusinessLayer/SegmentationCore/CmsAlgorithm/SurfaceComparisonStatistics.h

124

21 Data Structure Documentation

21.1 CmsAlgorithm::SurfaceComparisonStatistics Class Reference

Simple container for all comparison statistics.

```
#include <SurfaceComparisonStatistics.h>
```


Public Member Functions

- double **voxelVolume** () const
Volume (mm^3) of a voxel/sample.

Other mask-derived statistics

- double **referenceVolume** () const
Volume of the test surface (mm^3).
- double **testVolume** () const
Volume of the test surface (mm^3).
- double **intersectionVolume** () const
*Simply **voxelVolume**() * **intersectionVoxels**. (mm^3)*
- double **unionVolume** () const
*Simply **voxelVolume**() * **unionVoxels**. (mm^3)*

Data Fields

- std::string **referenceSource**
Source of the reference surface, e.g., a filename.
- std::string **testSource**
Source of the test surface, e.g., a filename.
- std::string **referenceName**
Name of the reference surface, e.g., Prostate.
- std::string **testName**
Name of the test surface.
- std::string **referenceSemanticName**
Name of the surface after name mapping was applied, e.g., PROSTATE.
- std::string **testSemanticName**
Name of the surface after name mapping was applied, e.g., PROSTATE.

Mesh-derived statistics

If both reference and test surface mesh data is available, these statistics are calculated on the mesh directly.
Most of the other statistics are calculated on binary mask representations of the surfaces.

- double **testVolume_mesh**
volume of the test surface (mm^3)
- double **referenceVolume_mesh**
volume of the reference surface (mm^3)
- double **testSurfaceArea_mesh**
Returns the surface area of the test surface. (mm^2)
- double **referenceSurfaceArea_mesh**
surface area of the reference surface (mm^2)
- double **testPointCount**
Number of mesh vertices in the test surface.
- double **referencePointCount**
Number of mesh vertices in the reference surface.
- double **meanDistance_mesh**
Mean distance error.
- double **stdDevDistance_mesh**

- *Standard deviation of the vertex-to-face distances.*
- double **maxDistance_mesh**
Maximum of the vertex-to-face distances.
- std::vector< double > [referenceToTestDistances](#)
Distance from each vertex in the reference surface to the nearest test surface face (mm).
- std::vector< double > [testToReferenceDistances](#)
Distance from each vertex in the test surface to the nearest reference surface face (mm).
- double [hausdorff_mesh](#)
Hausdorff distance is the maximum of the min distances between the reference and test (in both directions).
- double [distance_percentile](#)
Mesh-related distance metrics can operate on a percentile of the inputs in order to discard outliers.

Other distance statistics

- double **mahanabolisDistance**
- double **probabilisticDistance**
- double **interclassCorrelation**

Overlap statistics

Traditional statistics that are calculated from a contingency table/confusion matrix.

Each surface is converted from a mesh to a binary mask (at sampleSpacing). These comprise the "sets" being compared. The bounding box of the 2 surfaces is computed before conversion s.t. the resulting two binary mask images will have the same size and dimensions.

- std::array< double, 3 > [sampleSpacing](#)
Voxel size (mm) used when creating the binary masks representing the surfaces.
- double **trueNegative**
Number of background test voxels that correctly do not overlap the truth voxels (TN)
- double **truePositive**
Number of foreground test voxels that correctly do overlap the truth voxels (TP)
- double **falsePositive**
Number of foreground test voxels that incorrectly overlap the truth voxels (FP)
- double **falseNegative**
Number of background test voxels that incorrectly do not overlap the truth voxels (FN)
- double **referenceVoxels**
Total number of voxels inside the reference surface (i.e., non-zero label value)
- double **testVoxels**
Total number of voxels inside the test surface (i.e., non-zero label value)
- double **intersectionVoxels**
Total number of voxels shared by the reference and test surfaces.
- double **unionVoxels**
Total number of voxels in either reference or test surfaces.
- double [hausdorff](#)
Hausdorff distance.
- double [hausdorff_avg](#)
Average Hausdorff distance.

- double [hausdorff_avg_balanced](#)

Balanced Average Hausdorff distance.

- double **randIndex**
- double **randIndexAdjusted**
- double **cohenKappa**
- double **mutualInformation**
- double [variationOfInformation](#)

Defines the distance between two segmentations as the average conditional entropy of a segmentation given the other, and thus roughly measures the amount of randomness in a segmentation which cannot be explained by the other.

- double [jaccard](#) () const

The Jaccard coefficient $J(T, R) = \frac{|T \cap R|}{|T \cup R|}$.

- double [dice](#) () const

The Dice coefficient $DC(T, R) = \frac{2|T \cap R|}{|T| + |R|}$.

- double [falseNegativeDice](#) () const

False-Negative Dice: The volume of the reference that was missed by the Test volume relative to the size of the Reference plus the size of the Test.

- double [falsePositiveDice](#) () const

False-Positive Dice: The volume that the test segmented that was not part of reference (FP) relative to the size of the reference plus the size of the test.

- double [sensitivity](#) () const

Portion of voxels in the reference volume that are also identified as part of the test surface.

- double [specificity](#) () const

Portion of negative voxels(background) in the ground truth segmentation that are also identified as negative by the segmentation being evaluated.

- double [positivePredictiveValue](#) () const

$PPV = \frac{TP}{TP + FP}$

- double **negativePredictiveValue** () const

- double [falsePositiveRate](#) () const

Proportion of test volume incorrectly not covering the reference volume: $FP = \frac{|T \setminus R|}{T}$.

- double [falseNegativeRate](#) () const

Proportion of reference volume that is incorrectly not covered by the test volume.

- double **falseDiscoveryRate** () const
- double **accuracy** () const
- double **matthewsCorrelationCoefficient** () const
- double **informedness** () const
- double **markedness** () const
- double [fMeasure](#) (double beta=1.0) const

Also known as Harmonic mean.

- double [globalConsistencyError](#) () const

measures the extent to which one segmentation can be viewed as a refinement of the other.

- double **ROCAUC** () const

Area under ROC (Receiver operating characteristic) Curve (one system state)

- double [totalVolumeOverlap](#) () const

The Total Volume Overlap $TO(T, R) = \frac{|T \cap R|}{R}$.

- double [volumeSimilarity](#) () const

Measures the similarity between test and reference volumes VALUES, completely independent of overlap.

ROI image intensity statistics

If both reference and test image data is available, these statistics are calculated on the voxels within the reference and test surface respectively.

Note that the underlying image voxels are selected by applying the surface binary masks that were created for overlap statistics—not by using a bounding box. Therefore, the number of voxels are typically different between the reference and test. This is mainly to support other statistics that may be calculated outside this program.

The ref and test image volumes are first resampled to match the surface binary mask, using linear interpolation. E.g., if a sample spacing of 0.5mm is chosen, the image volume will first be resampled to 0.5mm.

- double **referenceMeanIntensity**
Mean intensity of the image voxels within the ref surface volume (e.g., where surface mask values are = 1)
- double **referenceStdDevIntensity**
Standard deviation of the intensity of the voxels within the ref surface volume (e.g., where surface mask values are = 1)
- double **testMeanIntensity**
mean intensity of the voxels within the test surface volume (e.g., where surface mask values are = 1)
- double **testStdDevIntensity**
Standard deviation of the intensity of the voxels within the test surface volume (e.g., where surface mask values are = 1)
- double **sdnr** () const
Signal difference-to-noise ratio of the image intensity within the reference and test volume.

21.1.1 Detailed Description

Simple container for all comparison statistics.

Many statistics are derived from others, so they are implemented as functions rather than public member variables.

Most statistics are implemented by the reference code provided by [Taha, A.A., Hanbury, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imaging 15, 29 \(2015\).](#)

If there is not specific documentation for the metric, refer to the paper for its formula.

21.1.2 Member Function Documentation

21.1.2.1 **dice()** double CmsAlgorithm::SurfaceComparisonStatistics::dice () const

The Dice coefficient $DC(T, R) = \frac{2|T \cap R|}{|T| + |R|}$.

Also known as Mean Overlap or Sorensen-Dice. It is the agreement between ref and test labels divided by the mean volume of the two regions.

Return values

-1	Not relevant for these surfaces
0	zero overlap
0.2- 0.4	fair agreement
0.4- 0.6	moderate agreement
0.6- 0.8	good agreement
1	perfect overlap

21.1.2.2 falseNegativeDice() `double CmsAlgorithm::SurfaceComparisonStatistics::falseNegativeDice () const`

False-Negative Dice: The volume of the reference that was missed by the Test volume relative to the size of the Reference plus the size of the Test.

Can be thought of as a measure of under-segmentation. Lower is better. Saturates at 2.0.

$$FND(T, R) = \frac{2|\overline{T} \cap R|}{|T| + |R|}$$

Return values

-1	Not relevant/invalid for these surfaces
0.↔ 0	No undersegmentation
2.↔ 0	All undersegmentation (no overlap)

Implemented as per "Quality Assurance Assessment of Diagnostic and Radiation Therapy–Simulation CT Image Registration for Head and Neck Radiation Therapy: Anatomic Region of Interest–based Comparison of Rigid and Deformable Algorithms" Abdallah S.R.Mohamed, et al. <https://doi.org/10.1148/radiol.14132871>

21.1.2.3 falseNegativeRate() `double CmsAlgorithm::SurfaceComparisonStatistics::falseNegativeRate () const`

Proportion of reference volume that is incorrectly not covered by the test volume.

$$FN = \frac{|R \setminus T|}{R}$$

21.1.2.4 falsePositiveDice() `double CmsAlgorithm::SurfaceComparisonStatistics::falsePositiveDice () const`

False-Positive Dice: The volume that the test segmented that was not part of reference (FP) relative to the size of the reference plus the size of the test.

Can be thought of as a measure of over-segmentation. Lower is better. Saturates at 2.0.

$$FND(T, R) = \frac{2|T \cap \bar{R}|}{|T| + |R|}$$

Note that the values can range from [0, 2]. A value of 1.0 does not unambiguously indicate anything. As long as the two volumes are roughly the same size, a value of 1 will tend to indicate zero overlap (i.e., all oversegmentation). However, the metric is proportional to the combined volume, so there are many situations where there can be zero or some overlap and the FPD will range below or above 1.0.

Return values

-1	Not relevant/invalid for these surfaces
0	No oversegmentation
2.↔ 0	All oversegmentation (no overlap)

Implemented as per "Quality Assurance Assessment of Diagnostic and Radiation Therapy–Simulation CT Image Registration for Head and Neck Radiation Therapy: Anatomic Region of Interest–based Comparison of Rigid and Deformable Algorithms" Abdallah S.R.Mohamed, et al. <https://doi.org/10.1148/radiol.14132871>

21.1.2.5 falsePositiveRate() `double CmsAlgorithm::SurfaceComparisonStatistics::falsePositiveRate () const`

Proportion of test volume incorrectly not covering the reference volume: $FP = \frac{|T \setminus R|}{T}$.

Also known as Fallout.

21.1.2.6 fMeasure() `double CmsAlgorithm::SurfaceComparisonStatistics::fMeasure (double beta = 1.0) const`

Also known as Harmonic mean.

At beta value 1.0, it's identical to [dice\(\)](#).

21.1.2.7 globalConsistencyError() `double CmsAlgorithm::SurfaceComparisonStatistics::globalConsistencyError () const`

measures the extent to which one segmentation can be viewed as a refinement of the other.

Segmentations which are related are considered to be consistent, since they could represent the same image segmented at different scales

21.1.2.8 jaccard() `double CmsAlgorithm::SurfaceComparisonStatistics::jaccard () const`

The Jaccard coefficient $J(T, R) = \frac{|T \cap R|}{|T \cup R|}$.

Also known as Union Overlap. Quantifies the intersection between ref and test labels divided by the union of both regions.

Return values

0	zero overlap
0.2-0.4	fair agreement
0.4-0.6	moderate agreement
0.6-0.8	good agreement
1	perfect overlap

21.1.2.9 positivePredictiveValue() `double CmsAlgorithm::SurfaceComparisonStatistics::positivePredictiveValue () const`

$$PPV = \frac{TP}{TP+FP}$$

Also known as Precision.

21.1.2.10 referenceVolume() `double CmsAlgorithm::SurfaceComparisonStatistics::referenceVolume () const`

Volume of the test surface (mm³).

`voxelVolume()` * referenceVoxels

21.1.2.11 sdnr() `double CmsAlgorithm::SurfaceComparisonStatistics::sdnr () const`

Signal difference-to-noise ratio of the image intensity within the reference and test volume.

$$SDNR = 2 \frac{|\mu_R - \mu_T|}{\sigma_R + \sigma_T}$$

21.1.2.12 sensitivity() `double CmsAlgorithm::SurfaceComparisonStatistics::sensitivity () const`

Portion of voxels in the reference volume that are also identified as part of the test surface.

$$Sensitivity = \frac{TP}{TP+FN}$$

Also known as True Positive Rate or Recall. Not as commonly used because it penalizes errors greater in smaller segmentations more than larger ones.

21.1.2.13 specificity() `double CmsAlgorithm::SurfaceComparisonStatistics::specificity () const`

Portion of negative voxels(background) in the ground truth segmentation that are also identified as negative by the segmentation being evaluated.

$$Specificity = \frac{TN}{TN+FP}$$

Also known as True Negative Rate. Not as commonly used because it penalizes errors greater in smaller segmentations more than larger ones.

21.1.2.14 testVolume() `double CmsAlgorithm::SurfaceComparisonStatistics::testVolume () const`

Volume of the test surface (mm³).

`voxelVolume()` * testVoxels

21.1.2.15 totalVolumeOverlap() `double CmsAlgorithm::SurfaceComparisonStatistics::totalVolumeOverlap () const`

The Total Volume Overlap $TO(T, R) = \frac{|T \cap R|}{R}$.

This is the percentage of the Reference volume that is overlapped by the Test volume.

Return values

-1	Not relevant for these surfaces
0	zero overlap
1.↔ 0	perfect overlap

21.1.2.16 volumeSimilarity() `double CmsAlgorithm::SurfaceComparisonStatistics::volumeSimilarity () const`

Measures the similarity between test and reference volumes VALUES, completely independent of overlap.

It's defined as absolute volume difference divided by the sum of the compared volumes. It primarily useful as a fast check to see if two surfaces have roughly the same size and shape. Further away from zero means more difference.

$$VS = 2 \frac{|T| - |R|}{|T| + |R|}$$

$$VS = 1 - \frac{|FN - FP|}{2TP + FP + FN}$$

Return values

<0	Test volume is smaller than reference
0	exact same volume
>0	Test volume is larger than reference

21.1.3 Field Documentation

21.1.3.1 distance_percentile `double CmsAlgorithm::SurfaceComparisonStatistics::distance_percentile`

Mesh-related distance metrics can operate on a percentile of the inputs in order to discard outliers.

Range is [0.0,1.0]. Default is 1.0, which means no outliers are discarded.

21.1.3.2 hausdorff `double CmsAlgorithm::SurfaceComparisonStatistics::hausdorff`

Hausdorff distance.

Calculated on binary mask. Applied to distance_percentile of the distance values

21.1.3.3 hausdorff_avg `double CmsAlgorithm::SurfaceComparisonStatistics::hausdorff_avg`

Average Hausdorff distance.

Calculated on binary mask. Applied to distance_percentile of the distance values

$$d_{AHD}(X, Y) = \left(\frac{1}{X} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{Y} \sum_{y \in Y} \min_{x \in X} d(x, y) \right) / 2$$

21.1.3.4 hausdorff_avg_balanced `double CmsAlgorithm::SurfaceComparisonStatistics::hausdorff_avg_balanced`

Balanced Average Hausdorff distance.

Calculated on binary mask. DOES NOT YET HONOR PERCENTILE.

O. U. Aydin, A. A. Taha, A. Hilbert, A. A. Khalil, I. Galinovic, J. B. Fiebach, D. Frey, and V. I. Madai, "On the usage of average hausdorff distance for segmentation performance assessment: Hidden bias when used for ranking," European Radiology Experimental, vol. 5, 2021. Available: <https://doi.org/10.1186/s41747-020-00200-2>

21.1.3.5 hausdorff_mesh `double CmsAlgorithm::SurfaceComparisonStatistics::hausdorff_mesh`

Hausdorff distance is the maximum of the min distances between the reference and test (in both directions).

For each vertex in the reference, find the closest point in the test. Repeat for each vertex in the test to the reference. Take the maximum of these two sets.

$$HD_{T \rightarrow R} = \max_{t \in T} \left\{ \min_{r \in R} \{ \text{distance}(t, r) \} \right\}$$

Note

unlike hausdorff and hausdorff_avg, this is calculated using the mesh vertices. The others are calculated using the binary mask. Both will be subject to the discretization error of sampleSpacing, albeit in slightly different ways.

21.1.3.6 meanDistance_mesh `double CmsAlgorithm::SurfaceComparisonStatistics::meanDistance_mesh`

Mean distance error.

Returns the mean of the vertex-to-face distances in mm.

21.1.3.7 referenceToTestDistances `std::vector<double> CmsAlgorithm::SurfaceComparisonStatistics::referenceToTestDistances`

Distance from each vertex in the reference surface to the nearest test surface face (mm).

Excludes distances > distance_percentile.

21.1.3.8 sampleSpacing `std::array<double, 3> CmsAlgorithm::SurfaceComparisonStatistics::sampleSpacing`

Voxel size (mm) used when creating the binary masks representing the surfaces.

Smaller sample spacing increases accuracy, but increases runtime

21.1.3.9 testToReferenceDistances `std::vector<double> CmsAlgorithm::SurfaceComparisonStatistics::testToReferenceDistances`

Distance from each vertex in the test surface to the nearest reference surface face (mm).

Excludes distances > distance_percentile.

21.1.3.10 variationOfInformation `double CmsAlgorithm::SurfaceComparisonStatistics::variationOfInformation`

Defines the distance between two segmentations as the average conditional entropy of a segmentation given the other, and thus roughly measures the amount of randomness in a segmentation which cannot be explained by the other.

Lower is better.

The documentation for this class was generated from the following file:

- [C:/wa/admire/BusinessLayer/SegmentationCore/CmsAlgorithm/SurfaceComparisonStatistics.h](#)

22 File Documentation

22.1 C:/wa/admire/BusinessLayer/SegmentationCore/CmsAlgorithm/SurfaceComparisonStatistics.h File Reference

```
#include <array>
#include <vector>
```

Data Structures

- class [CmsAlgorithm::SurfaceComparisonStatistics](#)
Simple container for all comparison statistics.

22.1.1 Detailed Description

Copyright

(c) 2012 - 2021 Elekta, Inc.

22.2 SurfaceComparisonStatistics.h

[Go to the documentation of this file.](#)

```

1
2 // THIS SOFTWARE IS THE PROPERTY OF ELEKTA AND THE PROPRIETARY RIGHTS OF ELEKTA
3 // INVOLVED IN THIS SUBJECT MATTER ARE EXPRESSLY RESERVED. IT IS SUBMITTED IN CONFIDENCE
4 // FOR A SPECIFIED PURPOSE PURSUANT TO A LICENSE AGREEMENT AND SUBJECT TO THE FOLLOWING
5 // RESTRICTION:
6 //
7 // THE RECIPIENT BY ACCEPTING THIS MATERIAL AGREES THAT THE MATERIAL
8 // WILL NOT BE USED COPIED OR REPRODUCED IN WHOLE OR IN PART NOR ITS
9 // CONTENTS REVEALED IN ANY MANNER OR TO ANY PERSON, NOR WILL THE
10 // SOFTWARE BE USED, COPIED OR REPRODUCED IN WHOLE OR IN PART FOR OR
11 // ON EQUIPMENT OTHER THAN THE DESIGNATED EQUIPMENT FOR WHICH IT WAS
12 // FURNISHED WITHOUT THE EXPRESS WRITTEN PERMISSION OF ELEKTA.
13 //
14
15
16
17
18
19 #ifndef CMSALGORITHM_SURFACECOMPARISONSTATISTICS_H_INCLUDED
20 #define CMSALGORITHM_SURFACECOMPARISONSTATISTICS_H_INCLUDED
21
22 #include <array>
23 #include <vector>
24
25 namespace CmsAlgorithm
26 {
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41 class SurfaceComparisonStatistics
42 {
43 public:
44
45     // CREATORS
46
47     SurfaceComparisonStatistics();
48     SurfaceComparisonStatistics(const SurfaceComparisonStatistics& orig) = default;
49     ~SurfaceComparisonStatistics() = default;
50     SurfaceComparisonStatistics& operator=(const SurfaceComparisonStatistics& rhs) = default;
51
52
53
54     std::string referenceSource;
55
56
57     std::string testSource;
58
59
60     std::string referenceName;
61
62
63     std::string testName;
64
65
66     std::string referenceSemanticName;
67
68
69     std::string testSemanticName;
70
71
72     double voxelVolume() const;
73
74
75
76
77
78
79
80
81     double testVolume_mesh;
82
83
84     double referenceVolume_mesh;
85

```

```
87     double testSurfaceArea_mesh;
88
89     double referenceSurfaceArea_mesh;
90
91     double testPointCount;
92
93     double referencePointCount;
94
95     double meanDistance_mesh;
96
97     double stdDevDistance_mesh;
98
99     double maxDistance_mesh;
100
101     std::vector<double> referenceToTestDistances;
102
103     std::vector<double> testToReferenceDistances;
104
105
106     double hausdorff_mesh;
107
108     double distance_percentile;
109
110
111
112
113     std::array<double, 3> sampleSpacing;
114
115     double trueNegative;
116     double truePositive;
117     double falsePositive;
118     double falseNegative;
119
120
121     double referenceVoxels;
122
123     double testVoxels;
124
125     double intersectionVoxels;
126
127     double unionVoxels;
128
129     double hausdorff;
130
131     double hausdorff_avg;
132
133     double hausdorff_avg_balanced;
134
135     double jaccard() const;
136
137     double dice() const;
138
139     double falseNegativeDice() const;
140
141     double falsePositiveDice() const;
142
143     // Other stats derived from the contingency table
144
145     double sensitivity() const;
146
147     double specificity() const;
148
149     double positivePredictiveValue() const;
150
151     double negativePredictiveValue() const;
```

```
331     double falsePositiveRate() const;
332
339
340     double falseNegativeRate() const;
341     double falseDiscoveryRate() const;
342     double accuracy() const;
343     double matthewsCorrelationCoefficient() const;
344     double informedness() const;
345     double markedness() const;
346
347
353
354     double fMeasure(double beta = 1.0) const;
355
356
364     double globalConsistencyError() const;
365
367     double ROCAUC() const;
368
379
380     double totalVolumeOverlap() const;
381
382
389     //
400
401     double volumeSimilarity() const;
402
403     double randIndex;
404     double randIndexAdjusted;
405     double cohenKappa;
406
407
408
409     double mutualInformation;
410
418
419     double variationOfInformation;
420
422
423
426
427
428     double mahanabolisDistance;
429     double probabilisticDistance;
430     double interclassCorrelation;
431
433
434
435
436
437
439
441
444     double referenceVolume() const;
445
448     double testVolume() const;
449
451     double intersectionVolume() const;
452
454     double unionVolume() const;
455
457
458
470
472
474     double referenceMeanIntensity;
475
477     double referenceStdDevIntensity;
478
480     double testMeanIntensity;
481
483     double testStdDevIntensity;
484
493
494     double sdnr() const;
495
497
498     bool operator<(const SurfaceComparisonStatistics& rhs) const;
499
500 };
501
```

```
502 }    // CmsAlgorithm
503
504 #endif
```

Index

Aero, [33](#)

ATI, [33](#)

C:/wa/admire/BusinessLayer/SegmentationCore/CmsAlgorithm/SurfaceComparisonStatistics.h,
[124](#), [125](#)

CBCT, [96](#)

CmsAlgorithm::SurfaceComparisonStatistics, [114](#)

 dice, [118](#)

 distance_percentile, [122](#)

 falseNegativeDice, [119](#)

 falseNegativeRate, [119](#)

 falsePositiveDice, [119](#)

 falsePositiveRate, [120](#)

 fMeasure, [120](#)

 globalConsistencyError, [120](#)

 hausdorff, [123](#)

 hausdorff_avg, [123](#)

 hausdorff_avg_balanced, [123](#)

 hausdorff_mesh, [123](#)

 jaccard, [120](#)

 meanDistance_mesh, [123](#)

 positivePredictiveValue, [121](#)

 referenceToTestDistances, [124](#)

 referenceVolume, [121](#)

 sampleSpacing, [124](#)

 sdnr, [121](#)

 sensitivity, [121](#)

 specificity, [121](#)

 testToReferenceDistances, [124](#)

 testVolume, [121](#)

 totalVolumeOverlap, [122](#)

 variationOfInformation, [124](#)

 volumeSimilarity, [122](#)

Contour Image Sequence, [96](#)

CUDA, [33](#)

current working directory, [4](#), [51](#)

Deep Learning, [51](#)

deep learning, [104](#)

deformation field, [47](#)

deformation vector field, [47](#)

dice

 CmsAlgorithm::SurfaceComparisonStatistics, [118](#)

DICOM 2009, [47](#)

diffeomorphic, [104](#)

Direction Cosines, [37](#)

distance_percentile

 CmsAlgorithm::SurfaceComparisonStatistics, [122](#)

dose, [52](#)

dose warping, [104](#)

driver, [33](#)

duplicate plan, [52](#)

DVF, [47](#), [104](#)

EMT, [104](#)

falseNegativeDice

 CmsAlgorithm::SurfaceComparisonStatistics, [119](#)

falseNegativeRate

 CmsAlgorithm::SurfaceComparisonStatistics, [119](#)

falsePositiveDice

 CmsAlgorithm::SurfaceComparisonStatistics, [119](#)

falsePositiveRate

 CmsAlgorithm::SurfaceComparisonStatistics, [120](#)

fMeasure

 CmsAlgorithm::SurfaceComparisonStatistics, [120](#)

Frame of Reference Transformation Matrix Type, [96](#)

globalConsistencyError

 CmsAlgorithm::SurfaceComparisonStatistics, [120](#)

hardware, [33](#)

hausdorff

 CmsAlgorithm::SurfaceComparisonStatistics, [123](#)

hausdorff_avg

 CmsAlgorithm::SurfaceComparisonStatistics, [123](#)

hausdorff_avg_balanced

 CmsAlgorithm::SurfaceComparisonStatistics, [123](#)

hausdorff_mesh

 CmsAlgorithm::SurfaceComparisonStatistics, [123](#)

IHE-RO, [96](#)

Image Type, [96](#)

InnoSetup, [4](#)

Installation, [2](#)

installation packages, [4](#)

intensity weighted fusion, [104](#)

inverse consistent, [104](#)

inverse deformation field, [47](#)

itk::MetaImage, [47](#)

jaccard

 CmsAlgorithm::SurfaceComparisonStatistics, [120](#)

license, [4](#)

lung, [104](#)

mappings, [41](#)

meanDistance_mesh

 CmsAlgorithm::SurfaceComparisonStatistics, [123](#)

memory, [33](#)

multiple GPUs, [33](#)

Nonuniform image spacing, [37](#)

NVIDIA, [33](#)

- nvidia-smi.exe, [33](#)
- OpenCL, [33](#)
- OrientationCheck, [37](#)
- Output file conventions, [54](#)
- patch fusion, [41](#)
- Phi, [33](#)
- plan, [52](#)
- positivePredictiveValue
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- Production version differences, [97](#)
- rackmount, [33](#)
- random forest, [104](#)
- recommended GPUs, [33](#)
- referenceToTestDistances
 - CmsAlgorithm::SurfaceComparisonStatistics, [124](#)
- referenceVolume
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- Remote Desktop, [33](#)
- RF label fusion, [104](#)
- ROI Interpreted Type, [96](#)
- RTDOSE, [52](#)
- RTPLAN, [52](#)
- sampleSpacing
 - CmsAlgorithm::SurfaceComparisonStatistics, [124](#)
- sdnr
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- sensitivity
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- specificity
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- STAPLE, [41](#), [104](#)
- structures, [41](#)
- surface reconstruction, [104](#)
- Tensorflow, [51](#)
- testToReferenceDistances
 - CmsAlgorithm::SurfaceComparisonStatistics, [124](#)
- testVolume
 - CmsAlgorithm::SurfaceComparisonStatistics, [121](#)
- totalVolumeOverlap
 - CmsAlgorithm::SurfaceComparisonStatistics, [122](#)
- Uninstalling, [3](#)
- UnitaryDirectionCosineTolerance, [37](#)
- Unity MR, [37](#)
- variationOfInformation
 - CmsAlgorithm::SurfaceComparisonStatistics, [124](#)
- Visual C++ Runtime, [4](#)
- volumeSimilarity
 - CmsAlgorithm::SurfaceComparisonStatistics, [122](#)