DICOMautomaton Reference Manual

Contents

1 Overview

1.1 About

DICOMautomaton is a collection of software tools for processing and analyzing medical images. Once a workflow has been developed, the aim of DICOMautomaton is to require minimal interaction to perform the workflow in an automated way. However, some interactive tools are also included for workflow development, exploratory analysis, and contouring.

DICOMautomaton is meant to be flexible enough to adapt to a wide variety of situations and has been incorporated into projects to provide: a local PACs, image analysis for various types of QA, kinetic modeling of perfusion images, automated fuzzy mapping of ROI names to a standard lexicon, dosimetric analysis, TCP and NTCP modeling, ROI contour/volume manipulation, estimation of surface dose, ray casting through patient and phantom geometry, rudimentary linac beam optimization, radiomics, and has been used in various ways to explore the relationship between toxicity and dose in sub-organ compartments.

Note: DICOMautomaton should **NOT** be used for clinical purposes. It is experimental software. It is suitable for research or support tool purposes only. It comes with no warranty or guarantee of any kind, either explicit or implied. Users of DICOMautomaton do so fully at their own risk.

1.2 Project Home

This project's homepage can be found at http://www.halclark.ca/. The source code is available at either https://gitlab.com/hdeanclark/DICOMautomaton/ or https://github.com/hdclark/DICOMautomaton/.

1.3 Download

DICOMautomaton relies only on open source software and is itself open source software. Source code is available at https://github.com/hdclark/DICOMautomaton.

Currently, binaries are not provided. Only linux is supported and a recent C++ compiler is needed. A `PKGBUILD` file is provided for Arch Linux and derivatives, and CMake can be used to generate deb files for Debian derivatives. A docker container is available for easy portability to other systems. DICOMautomaton has successfully run on x86, x86_64, and most ARM systems. To maintain flexibility, DICOMautomaton is generally not ABI or API stable.

1.4 License and Copying

All materials herein which may be copywrited, where applicable, are. Copyright 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 hal clark. See the `LICENSE` file for details about the license. Informally, DICOMautomaton is available under a GPLv3+ license. The Imebra library is bundled for convenience and was not written by hal clark; consult its license file in `src/imebra/license.txt`.

All liability is herefore disclaimed. The person(s) who use this source and/or software do so strictly under their own volition. They assume all associated liability for use and misuse, including but not limited to damages, harm, injury, and death which may result, including but not limited to that arising from unforeseen and unanticipated implementation defects.

1.5 Dependencies

Dependencies are listed in the `PKGBUILD` file (using Arch Linux package naming conventions) and in the `CMakeLists.txt` file (Debian package naming conventions) bundled with the source code. See https://github.com/hdclark/DICOMautomaton. Broadly, DICOMautomaton depends on Boost, CGAL, SFML, Eigen, Asio, Wt, and PostgreSQL.

Notably, DICOMautomaton depends on the author's 'Ygor,' 'Explicator,' and 'YgorClustering' projects. See https://gitlab.com/hdeanclark/Ygor (mirrored at https://github.com/hdclark/Ygor), https://gitlab.com/hdeanclark/Explicator (mirrored at https://github.com/hdclark/Explicator), and (only for compilation) https://gitlab.com/hdeanclark/YgorClustering (mirrored at https://github.com/hdclark/YgorClustering).

1.6 Feedback

All feedback, questions, comments, and pull requests are welcomed.

1.7 FAQs

Q. What is the best way to use DICOMautomaton?
A. DICOMautomaton provides a command-line interface, SFML-based image viewer, and limited web interface. The command-line interface is most conducive to automation, the viewer works best for interactive tasks, and the web interface works well for specific installations.

Q. How do I contribute, report bugs, or contact the author?
A. All feedback, questions, comments, and pull requests are welcomed. Please find contact information at https://github.com/hdclark/DICOMautomaton.

1.8 Citing

Several publications and presentations refer to DICOMautomaton or describe some aspect of it. Here are a few:

- H. Clark, J. Beaudry, J. Wu, and S. Thomas. Making use of virtual dimensions for visualization and contouring. Poster presentation at the International Conference on the use of Computers in Radiation Therapy, London, UK. June 27-30, 2016.
- H. Clark, S. Thomas, V. Moiseenko, R. Lee, B. Gill, C. Duzenli, and J. Wu. Automated segmentation and dose-volume analysis with DICOMautomaton. In the Journal of Physics: Conference Series, vol. 489, no. 1, p. 012009. IOP Publishing, 2014.
- H. Clark, J. Wu, V. Moiseenko, R. Lee, B. Gill, C. Duzenli, and S. Thomas. Semi-automated contour recognition using DICOMautomaton. In the Journal of Physics: Conference Series, vol. 489, no. 1, p. 012088. IOP Publishing, 2014.
- H. Clark, J. Wu, V. Moiseenko, and S. Thomas. Distributed, asynchronous, reactive dosimetric and outcomes analysis using DICOMautomaton. Poster presentation at the COMP Annual Scientific Meeting, Banff, Canada. July 9–12,

2014.

If you use DICOMautomaton in an academic work, we ask that you please cite the most relevant publication for that work, if possible.

1.9 Components

1.9.1 dicomautomaton_dispatcher

1.9.1.1 Description

The core command-line interface to DICOMautomaton is the `dicomautomaton_dispatcher` program. It is presents an interface based on chaining of discrete operations on collections of images, DICOM images, DICOM radiotherapy files (RTSTRUCTS and RTDOSE), and various other types of files. `dicomautomaton_dispatcher` has access to all defined operations described in Operations. It can be used to launch both interactive and non-interactive tasks. Data can be sourced from a database or files in a variety of formats.

Name/label selectors in dicomautomaton_dispatcher generally support fuzzy matching via libexplicator or regular expressions. The operations and parameters that provide these options are documented in Operations.

Filetype support differs in some cases. A custom FITS file reader and writer are supported, and DICOM files are generally supported. There is currently no support for RTPLANs, though DICOM image, RTSTRUCT, and RTDOSE files are well supported. There is limited support for writing files – currently JPEG, PNG, and FITS images; RTDOSE files; and Boost.Serialize archive writing are supported.

1.9.1.2 Usage Examples

- `dicomautomaton_dispatcher --help`
  Print a listing of all available options.

- `dicomautomaton_dispatcher CT*dcm`
  Launch the default interactive viewer to inspect a collection of computed tomography images.

- `dicomautomaton_dispatcher MR*dcm`
  Launch the default interactive viewer to inspect a collection of magnetic resonance images.

- `dicomautomaton_dispatcher -o SFML_Viewer MR*dcm`
  *Launch the default interactive viewer to inspect a collection of magnetic resonance images. Note that files specified on the command line are always loaded **prior** to running any operations. Injecting files midway through the operation chain must make use of an operation designed to do so.*

- `dicomautomaton_dispatcher CT*dcm RTSTRUCT*dcm RTDOSE*dcm -o Average -o SFML_Viewer`
  Load DICOM files, perform an averaging operation, and then launch the SFML viewer to inspect the output.

- `dicomautomaton_dispatcher ./RTIMAGE.dcm -o AnalyzePicketFence:ImageSelection='last':Int`
  Perform a picket fence analysis of an RTIMAGE file.

- `dicomautomaton_dispatcher -f create_temp_view.sql -f select_records_from_temp_view.sql -o ComputeSomething`
  Load a SQL common file that creates a SQL view, issue a query involving the view which returns some DICOM file(s). Perform analysis 'ComputeSomething' with the files.

- `dicomautomaton_dispatcher -f common.sql -f seriesA.sql -n -f seriesB.sql -o SFML_Viewer`
  Load two distinct groups of data. The second group does not 'see' the file 'common.sql' side effects – the queries are totally separate.

- `dicomautomaton_dispatcher fileA fileB -s fileC adir/ -m PatientID=XYZ003 -o ComputeXYZ -o SFML_Viewer`
  Load standalone files and all files in specified directory. Inform the analysis 'ComputeXYZ' of the patient's ID, launch the analysis, and then interactively view.

- `dicomautomaton_dispatcher CT*dcm -o ModifyingOperation -o BoostSerializeDrover`
  Launch the default interactive viewer to inspect a collection of computed tomography images, perform an operation that modifies them, and serialize the internal state for later using the BoostSerializeDrover operation.

1.9.2 dicomautomaton_webserver

1.9.2.1 Description

This web server presents most operations in an interactive web page. Some operations are disabled by default (e.g., BuildLexiconInteractively because they are not designed to be operated via remote procedure calls. This routine should be run within a capability-limiting environment, but access to an X server is required. A Docker script is bundled with DICOMautomaton sources which includes everything needed to function properly.

1.9.2.2 Usage Examples

- `dicomautomaton_webserver --help`
  Print a listing of all available options. Note that most configuration is done via editing configuration files. See `/etc/DICOMautomaton/`.

-
  ```
  dicomautomaton_webserver --config /etc/DICOMautomaton/webserver.conf
  --http-address 0.0.0.0 --http-port 8080 --docroot='/etc/DICOMautomaton/'
  ```
  Launch the webserver on any interface and port 8080.

1.9.3 dicomautomaton_bsarchive_convert

1.9.3.1 Description

A program for converting Boost.Serialization archives types which DICOMautomaton can read. These archives need to be created by the BoostSerializeDrover operation. Some archive types are concise and not portable (i.e., binary archives), or verbose (and thus slow to read and write) and portable (i.e., XML, plain text). To combat verbosity, on-the-fly gzip compression and decompression is supported. This program can be used to convert archive types.

1.9.3.2 Usage Examples

-
  ```
  dicomautomaton_bsarchive_convert --help
  ```
  Print a listing of all available options.

-
  ```
  dicomautomaton_bsarchive_convert -i file.binary -o file.xml
  -t 'XML'
  ```
  Convert a binary archive to a portable XML archive.

-
  ```
  dicomautomaton_bsarchive_convert -i file.binary.gz -o file.xml.gz
  -t 'gzip-xml'
  ```
  Convert a binary archive to a gzipped portable XML archive.

-
  ```
  dicomautomaton_bsarchive_convert -i file.binary.gz -o file.xml
  -t 'XML'
  ```
  Convert a gzipped binary archive to a non-gzipped portable XML archive.

-
  ```
  dicomautomaton_bsarchive_convert -i file.xml.gz -o file.txt
  -t 'txt'
  ```
  Convert a gzipped binary archive to a non-gzipped, portable, and inspectable text archive.

-
  ```
  dicomautomaton_bsarchive_convert -i file.txt -o file.txt.gz
  -t 'gzip-txt'
  ```
  Convert an uncompressed text archive to a compressed text archive. Note that this conversion is effectively the same as simply `gzip file.txt`.

-
  ```
  dicomautomaton_bsarchive_convert -i file.xml.gz -o file.bin
  -t 'binary'
  ```
  Convert a compressed archive to a binary file. Note that binary archives should only expect to be readable on the same hardware with the same versions and are therefore best for checkpointing calculations that can fail or may need to be tweaked later.*

- `dicomautomaton_bsarchive_convert -i file.xml.gz -o file.bin.gz -t 'gzip-binary'`
  Convert a compressed archive to a compressed binary file.

1.9.4 dicomautomaton_dump

1.9.4.1 Description

This program is extremely simplistic. Given a single DICOM file, it prints to stdout the value of one DICOM tag. This program is best used in scripts, for example to check the modality or a file.

1.9.4.2 Usage Examples

- `dicomautomaton_dump afile.dcm 0x0008 0x0060`
  Print the value of the DICOM tag (0x0008,0x0060) aka (0008,0060).

1.9.5 pacs_ingress

1.9.5.1 Description

Given a DICOM file and some additional metadata, insert the data into a PACs system database. The file itself will be copied into the database and various bits of data will be deciphered. Note that at the moment a 'gdcmdump' file must be provided and is stored alongside the DICOM file in the database filestore. This sidecar file is meant to support ad-hoc DICOM queries without having to index the entire file. Also note that imports into the database are minimal, leaving files with multiple NULL values. This is done to improve ingress times. A separate database refresh (pacs_refresh) must be performed to replace NULL values.

1.9.5.2 Usage Examples

- `pacs_ingress --help`
  Print a listing of all available options.
- `pacs_ingress -f '/tmp/a.dcm' -g '/tmp/a.gdcmdump' -p 'XYZ Study 2019' -c 'Study concerning XYZ.'`
  Insert the file '/tmp/a.dcm' into the database.

1.9.6 pacs_refresh

1.9.6.1 Description

A program for trying to replace database NULLs, if possible, using stored files. This program is complementary to pacs_ingress. Note that the `--days-back/-d` parameter should always be specified.

1.9.6.2 Usage Examples

- `pacs_refresh --help`
  Print a listing of all available options.
- `pacs_refresh -d 7`
  Perform a refresh of the database, restricting to files imported within the previous 7 days.

1.9.7 pacs_duplicate_cleaner

1.9.7.1 Description

Given a DICOM file, check if it is in the PACS DB. If so, delete the file. Note that a full, byte-by-byte comparison is NOT performed – rather only the top-level DICOM unique identifiers are (currently) compared. No other metadata is considered. So this program is not suitable if DICOM files have been modified without re-assigning unique identifiers! (Which is non-standard behaviour.) Note that if an *exact* comparison is desired, using a traditional file de-duplicator will work.

1.9.7.2 Usage Examples

- `pacs_duplicate_cleaner --help`
  Print a listing of all available options.
- `pacs_duplicate_cleaner -f '/path/to/a/dicom/file.dcm'`
  Check if 'file.dcm' is already in the PACS DB. If so, delete it ('file.dcm').
- `pacs_duplicate_cleaner -f '/path/to/a/dicom/file.dcm' -n`
  Check if 'file.dcm' is already in the PACS DB, but do not delete anything.

2 Operations

2.1 AccumulateRowsColumns

2.1.1 Description

This operation generates row- and column-profiles of images in which the entire row or column has been summed together. It is useful primarily for detection of axes-aligned edges or ridges.

2.1.2 Notes

- It is often useful to pre-process inputs by computing an in-image-plane derivative, gradient magnitude, or similar (i.e., something to emphasize edges) before calling this routine. It is not necessary, however.

2.1.3 Parameters

- ImageSelection

2.1.3.1 ImageSelection

2.1.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.1.3.1.2 Default

- `"last"`

2.1.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`

- `"key1@.*value1.*;key2@^value2$;first"`

2.2 AnalyzeLightRadFieldCoincidence

2.2.1 Description

This operation analyzes the selected images to compare light and radiation field coincidence for fixed, symmetric field sizes. Coincidences are extracted automatically by fitting Gaussians to the peak nearest to one of the specified field boundaries and comparing offset from one another. So, for example, a 10x10cm MLC-defined field would be compared to a 15x15cm field if there are sharp edges (say, metal rulers) that define a 10x10cm field (i.e., considered to represent the light field). Horizontal and vertical directions (both positive and negative) are all analyzed separately.

2.2.2 Notes

- This routine assumes both fields are squarely aligned with the image axes. Alignment need not be perfect, but the Gaussians may be significantly broadened if there is misalignment. This should be fixed in a future revision.
- It is often useful to pre-process inputs by computing an in-image-plane derivative, gradient magnitude, or similar (i.e., something to emphasize edges) before calling this routine. It may not be necessary, however.

2.2.3 Parameters

- ImageSelection
- ToleranceLevel
- EdgeLengths
- SearchDistance
- PeakSimilarityThreshold
- UserComment
- OutputFileName
- InteractivePlots

2.2.3.1 ImageSelection

2.2.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or ‘4D’ time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.2.3.1.2 Default

- `"last"`

2.2.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.2.3.2 ToleranceLevel

2.2.3.2.1 Description

Controls detected edge visualization for easy identification of edges out of tolerance. Note: this value refers to edge-to-edge separation, not edge-to-nominal distances. This value is in DICOM units.

2.2.3.2.2 Default

- `"1.0"`

2.2.3.2.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`
- `"inf"`

2.2.3.3 EdgeLengths

2.2.3.3.1 Description

Comma-separated list of (symmetric) edge lengths fields should be analyzed at. For example, if 50x50, 100x100, 150x150, and 200x200 (all in mm) fields are to be analyzed, this argument would be '50,100,150,200' and it will be assumed that the field centre is at DICOM position (0,0,0). All values are in DICOM units.

2.2.3.3.2 Default

- `"100"`

2.2.3.3.3 Examples

- `"100.0"`
- `"50,100,150,200,300"`
- `"10.273,20.2456"`

2.2.3.4 SearchDistance

2.2.3.4.1 Description

The distance around the anticipated field edges to search for edges (actually sharp peaks arising from edges). If an edge is further away than this value from the anticipated field edge, then the coincidence will be ignored altogether. The value should be greater than the largest action/tolerance threshold with some additional margin (so gross errors can be observed), but small enough that spurious edges (i.e., unintended features in the image, such as metal fasteners, or artifacts near the field edge) do not replace the true field edges. The 'sharpness' of the field edge (resulting from the density of the material used to demarcate the edge) can impact this value; if the edge is not sharp, then the peak will be shallow, noisy, and may therefore travel around depending on how the image is pre-processed. Note that both radiation field and light field edges may differ from the 'nominal' anticipated edges, so this wobble factor should be incorporated in the search distance. This quantity must be in DICOM units.

2.2.3.4.2 Default

- `"3.0"`

2.2.3.4.3 Examples

- `"2.5"`
- `"3.0"`
- `"5.0"`

2.2.3.5 PeakSimilarityThreshold

2.2.3.5.1 Description

Images can be taken such that duplicate peaks will occur, such as when field sizes are re-used. Peaks are therefore de-duplicated. This value (as a %, ranging from [0,100]) specifies the threshold of disimilarity below which peaks are considered duplicates. A low value will make duplicates confuse the analysis, but a high value may cause legitimate peaks to be discarded depending on the attenuation cababilties of the field edge markers.

2.2.3.5.2 Default

- `"25"`

2.2.3.5.3 Examples

- `"5"`
- `"10"`
- `"15"`
- `"50"`

2.2.3.6 UserComment

2.2.3.6.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.2.3.6.2 Default

- `""`

2.2.3.6.3 Examples

- `""`
- `"6MV"`
- `"Using XYZ"`
- `"Test with thick metal edges"`

2.2.3.7 OutputFileName

2.2.3.7.1 Description

A filename (or full path) in which to append field edge coincidence data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.2.3.7.2 Default

- `""`

2.2.3.7.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.2.3.8 InteractivePlots

2.2.3.8.1 Description

Whether to interactively show plots showing detected edges.

2.2.3.8.2 Default

- `"false"`

2.2.3.8.3 Examples

- `"true"`
- `"false"`

2.3 AnalyzePicketFence

2.3.1 Description

This operation extracts MLC positions from a picket fence image.

2.3.2 Notes

- This routine requires data to be pre-processed. The gross picket area should be isolated and the leaf junction areas contoured (one contour per junction). Both can be accomplished via thresholding. Additionally, stray pixels should be filtered out using, for example, median or conservative filters.
- This routine analyzes the picket fences on the plane in which they are specified within the DICOM file, which often coincides with the image receptor ('RTImageSID'). Tolerances are evaluated on the isoplane, so the image is projected before measuring distances, but the image itself is not altered; a uniform magnification factor of SAD/SID is applied to all distances.

2.3.3 Parameters

- ImageSelection
- MLCModel
- MLCROILabel
- JunctionROILabel
- PeakROILabel
- MinimumJunctionSeparation
- ThresholdDistance
- LeafGapsFileName
- ResultsSummaryFileName
- UserComment
- InteractivePlots

2.3.3.1 ImageSelection

2.3.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals.

Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.3.3.1.2 Default

- `"last"`

2.3.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.3.3.2 MLCModel

2.3.3.2.1 Description

The MLC design geometry to use. ‘VarianMillenniumMLC80’ has 40 leafs in each bank; leaves are 10mm wide at isocentre; and the maximum static field size is 40cm x 40cm. ‘VarianMillenniumMLC120’ has 60 leafs in each bank; the 40 central leaves are 5mm wide at isocentre; the 20 peripheral leaves are 10mm wide; and the maximum static field size is 40cm x 40cm. ‘VarianHD120’ has 60 leafs in each bank; the 32 central leaves are 2.5mm wide at isocentre; the 28 peripheral leaves are 5mm wide; and the maximum static field size is 40cm x 22cm.

2.3.3.2.2 Default

- `"VarianMillenniumMLC120"`

2.3.3.2.3 Examples

- `"VarianMillenniumMLC80"`
- `"VarianMillenniumMLC120"`
- `"VarianHD120"`

2.3.3.3 MLCROILabel

2.3.3.3.1 Description

An ROI imitating the MLC axes of leaf pairs is created. This is the label to apply to it. Note that the leaves are modeled with thin contour rectangles of virtually zero area. Also note that the outline colour is significant and denotes leaf pair pass/fail.

2.3.3.3.2 Default

- `"Leaves"`

2.3.3.3.3 Examples

- `"MLC_leaves"`
- `"MLC"`
- `"approx_leaf_axes"`

2.3.3.4 JunctionROILabel

2.3.3.4.1 Description

An ROI imitating the junction is created. This is the label to apply to it. Note that the junctions are modeled with thin contour rectangles of virtually zero area.

2.3.3.4.2 Default

- `"Junction"`

2.3.3.4.3 Examples

- `"Junction"`
- `"Picket_Fence_Junction"`

2.3.3.5 PeakROILabel

2.3.3.5.1 Description

ROIs encircling the leaf profile peaks are created. This is the label to apply to it. Note that the peaks are modeled with small squares.

2.3.3.5.2 Default

- `"Peak"`

2.3.3.5.3 Examples

- `"Peak"`
- `"Picket_Fence_Peak"`

2.3.3.6 MinimumJunctionSeparation

2.3.3.6.1 Description

The minimum distance between junctions on the SAD isoplane in DICOM units (mm). This number is used to de-duplicate automatically detected junctions. Analysis results should not be sensitive to the specific value.

2.3.3.6.2 Default

- `"10.0"`

2.3.3.6.3 Examples

- `"5.0"`
- `"10.0"`
- `"15.0"`
- `"25.0"`

2.3.3.7 ThresholdDistance

2.3.3.7.1 Description

The threshold distance in DICOM units (mm) above which MLC separations are considered to 'fail'. Each leaf pair is evaluated separately. Pass/fail status is also indicated by setting the leaf axis contour colour (blue for pass, red for fail).

2.3.3.7.2 Default

- `"1.0"`

2.3.3.7.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`

2.3.3.8 LeafGapsFileName

2.3.3.8.1 Description

This file will contain gap and nominal-vs-actual offset distances for each leaf pair. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.3.3.8.2 Default

- `""`

2.3.3.8.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.3.3.9 ResultsSummaryFileName

2.3.3.9.1 Description

This file will contain a brief summary of the results. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.3.3.9.2 Default

- `""`

2.3.3.9.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.3.3.10 UserComment

2.3.3.10.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.3.3.10.2 Default

- `""`

2.3.3.10.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.3.3.11 InteractivePlots

2.3.3.11.1 Description

Whether to interactively show plots showing detected edges.

2.3.3.11.2 Default

- `"false"`

2.3.3.11.3 Examples

- `"true"`
- `"false"`

2.4 ApplyCalibrationCurve

2.4.1 Description

This operation applies a given calibration curve to voxel data inside the specified ROI(s). It is designed to apply calibration curves, but is useful for transforming voxel intensities using any supplied 1D curve.

2.4.2 Notes

- This routine can handle overlapping or duplicate contours.

2.4.3 Parameters

- Channel
- ImageSelection
- ContourOverlap
- Inclusivity
- CalibCurveFileName
- NormalizedROILabelRegex
- ROILabelRegex

2.4.3.1 Channel

2.4.3.1.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.4.3.1.2 Default

- `"-1"`

2.4.3.1.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.4.3.2 ImageSelection

2.4.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array.

Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.4.3.2.2 Default

- `"last"`

2.4.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.4.3.3 ContourOverlap

2.4.3.3.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.4.3.3.2 Default

- `"ignore"`

2.4.3.3.3 Examples

- `"ignore"`

- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.4.3.4 Inclusivity

2.4.3.4.1 Description

Controls how voxels are deemed to be ‘within’ the interior of the selected ROI(s). The default ‘center’ considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, ‘planar_corner_inclusive’, considers a voxel interior if ANY corner is interior. The second, ‘planar_corner_exclusive’, considers a voxel interior if ALL (four) corners are interior.

2.4.3.4.2 Default

- `"center"`

2.4.3.4.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.4.3.5 CalibCurveFileName

2.4.3.5.1 Description

The file from which a calibration curve should be read from. The format should be line-based with either 2 or 4 numbers per line. For 2 numbers: (current pixel value) (new pixel value) and for 4 numbers: (current pixel value) (uncertainty) (new pixel value) (uncertainty). Uncertainties refer to the prior number and may be uniformly zero if unknown. Lines beginning with ‘#’ are treated as comments and ignored. The curve is linearly interpolated, and must span the full range of pixel values. This is done to avoid extrapolation within the operation since the correct behaviour will differ depending on the specifics of the calibration.

2.4.3.5.2 Default

- `""`

2.4.3.5.3 Examples

- `"/tmp/calib.dat"`

2.4.3.6 NormalizedROILabelRegex

2.4.3.6.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.4.3.6.2 Default

- `".*"`

2.4.3.6.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.4.3.7 ROILabelRegex

2.4.3.7.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.4.3.7.2 Default

- `".*"`

2.4.3.7.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.5 AutoCropImages

2.5.1 Description

This operation crops image slices using image-specific metadata embedded within the image.

2.5.2 Parameters

- ImageSelection
- DICOMMargin
- RTIMAGE

2.5.2.1 ImageSelection

2.5.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.5.2.1.2 Default

- `"all"`

2.5.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.5.2.2 DICOMMargin

2.5.2.2.1 Description

The amount of margin (in the DICOM coordinate system) to spare from cropping.

2.5.2.2.2 Default

- `"0.0"`

2.5.2.2.3 Examples

- `"0.1"`
- `"2.0"`
- `"-0.5"`
- `"20.0"`

2.5.2.3 RTIMAGE

2.5.2.3.1 Description

If true, attempt to crop the image using information embedded in an RTIMAGE. This option cannot be used with the other options.

2.5.2.3.2 Default

- `"true"`

2.5.2.3.3 Examples

- `"true"`
- `"false"`

2.6 Average

2.6.1 Description

This operation averages image arrays/volumes. It can average over spatial or temporal dimensions. However, rather than relying specifically on time for temporal averaging, any images that have overlapping voxels can be averaged.

2.6.2 Notes

- This operation is typically used to create an aggregate view of a large volume of data. It may also increase SNR and can be used for contouring purposes.

2.6.3 Parameters

- ImageSelection
- AveragingMethod

2.6.3.1 ImageSelection

2.6.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.6.3.1.2 Default

- `"last"`

2.6.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.6.3.2 AveragingMethod

2.6.3.2.1 Description

The averaging method to use. Valid methods are ‘overlapping-spatially’ and ‘overlapping-temporally’.

2.6.3.2.2 Default

- `""`

2.6.3.2.3 Examples

- `"overlapping-spatially"`
- `"overlapping-temporally"`

2.7 BCCAExtractRadiomicFeatures

2.7.1 Description

This operation extracts radiomic features from an image and one or more ROIs.

2.7.2 Notes

- This is a ‘simplified’ version of the full radiomics extract routine that uses defaults that are expected to be reasonable across a wide range of scenarios.

2.7.3 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- FractionalAreaTolerance
- SimplificationMethod
- UserComment
- FeaturesFileName
- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- ScaleFactor
- ImageFileName
- ColourMapRegex
- WindowLow
- WindowHigh

2.7.3.1 NormalizedROILabelRegex

2.7.3.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.7.3.1.2 Default

- `".*"`

2.7.3.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.7.3.2 ROILabelRegex

2.7.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.7.3.2.2 Default

- `".*"`

2.7.3.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.7.3.3 FractionalAreaTolerance

2.7.3.3.1 Description

The fraction of area each contour will tolerate during simplified. This is a measure of how much the contour area can change due to simplification.

2.7.3.3.2 Default

- `"0.05"`

2.7.3.3.3 Examples

- `"0.001"`
- `"0.01"`
- `"0.02"`
- `"0.05"`
- `"0.10"`

2.7.3.4 SimplificationMethod

2.7.3.4.1 Description

The specific algorithm used to perform contour simplification. 'Vertex removal' is a simple algorithm that removes vertices one-by-one without replacement. It iteratively ranks vertices and removes the single vertex that has the least impact on contour area. It is best suited to removing redundant vertices or whenever new vertices should not be added. 'Vertex collapse' combines two adjacent vertices into a single vertex at their midpoint. It iteratively ranks vertex pairs and removes the single vertex that has the least total impact on contour area. Note that small sharp features that alternate inward and outward will have a small total area cost, so will be pruned early. Thus this technique acts as a low-pass filter and will defer simplification of high-curvature regions until necessary. It is more economical compared to vertex removal in that it will usually simplify contours more for a given tolerance (or, equivalently, can retain contour fidelity better than vertex removal for the same number of vertices). However, vertex collapse performs an averaging that may result in numerical imprecision.

2.7.3.4.2 Default

- `"vert-rem"`

2.7.3.4.3 Examples

- `"vertex-collapse"`
- `"vertex-removal"`

2.7.3.5 UserComment

2.7.3.5.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.7.3.5.2 Default

- `""`

2.7.3.5.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.7.3.6 FeaturesFileName

2.7.3.6.1 Description

Features will be appended to this file. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.7.3.6.2 Default

- `""`

2.7.3.6.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.7.3.7 ImageSelection

2.7.3.7.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.7.3.7.2 Default

- `"last"`

2.7.3.7.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.7.3.8 NormalizedROILabelRegex

2.7.3.8.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.7.3.8.2 Default

- `".*"`

2.7.3.8.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.7.3.9 ROILabelRegex

2.7.3.9.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them.

All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.7.3.9.2 Default

- `".*"`

2.7.3.9.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.7.3.10 ScaleFactor

2.7.3.10.1 Description

This factor is applied to the image width and height to magnify (larger than 1) or shrink (less than 1) the image. This factor only affects the output image size. Note that aspect ratio is retained, but rounding for non-integer factors may lead to small (1-2 pixel) discrepancies.

2.7.3.10.2 Default

- `"1.5"`

2.7.3.10.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`
- `"5.23"`

2.7.3.11 ImageFileName

2.7.3.11.1 Description

The file name to use for the image. If blank, a filename will be generated sequentially.

2.7.3.11.2 Default

- `""`

2.7.3.11.3 Examples

- `""`
- `"/tmp/an_image.png"`
- `"afile.png"`

2.7.3.12 ColourMapRegex

2.7.3.12.1 Description

The colour mapping to apply to the image if there is a single channel. The default will match the first available, and if there is no matching map found, the first available will be selected.

2.7.3.12.2 Default

- `".*"`

2.7.3.12.3 Examples

- `"Viridis"`
- `"Magma"`
- `"Plasma"`
- `"Inferno"`
- `"Jet"`
- `"MorelandBlueRed"`
- `"MorelandBlackBody"`
- `"MorelandExtendedBlackBody"`
- `"KRC"`
- `"ExtendedKRC"`
- `"Kovesi_LinKRYW_5-100_c64"`
- `"Kovesi_LinKRYW_0-100_c71"`
- `"Kovesi_Cyclic_cet-c2"`
- `"LANLOliveGreentoBlue"`
- `"YgorIncandescent"`
- `"LinearRamp"`

2.7.3.13 WindowLow

2.7.3.13.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or lower will be assigned the lowest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.7.3.13.2 Default

- `""`

2.7.3.13.3 Examples

- `""`
- `"-1.23"`
- `"0"`
- `"1E4"`

2.7.3.14 WindowHigh

2.7.3.14.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or higher will be assigned the highest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.7.3.14.2 Default

- `""`

2.7.3.14.3 Examples

- `""`
- `"1.23"`
- `"0"`
- `"10.3E4"`

2.8 BoostSerializeDrover

2.8.1 Description

This operation exports all loaded state to a serialized format that can be loaded again later. Is is especially useful for suspending long-running operations with intermittant interactive sub-operations.

2.8.2 Parameters

- Filename
- Components

2.8.2.1 Filename

2.8.2.1.1 Description

The filename (or full path name) to which the serialized data should be written. The file format is gzipped XML, which should be portable across most CPUs.

2.8.2.1.2 Default

- `"/tmp/boost_serialized_drover.xml.gz"`

2.8.2.1.3 Examples

- `"/tmp/out.xml.gz"`
- `"./out.xml.gz"`
- `"out.xml.gz"`

2.8.2.2 Components

2.8.2.2.1 Description

Which components to include in the output. Currently, any combination of (all images), (all contours), (all point clouds), (all surface meshes), and (all treatment plans) can be selected. Note that RTDOSEs are treated as images.

2.8.2.2.2 Default

- `"images+contours+pointclouds+surfacemeshes+tplans"`

2.8.2.2.3 Examples

- `"images"`
- `"images+pointclouds"`
- `"images+pointclouds+surfacemeshes"`
- `"pointclouds+surfacemeshes"`
- `"tplans+images+contours"`

- `"contours+images+pointclouds"`

2.9 BuildLexiconInteractively

2.9.1 Description

This operation interactively builds a lexicon using the currently loaded contour labels. It is useful for constructing a domain-specific lexicon from a set of representative data.

2.9.2 Parameters

- CleanLabels
- JunkLabel
- OmitROILabelRegex
- LexiconSeedFile

2.9.2.1 CleanLabels

2.9.2.1.1 Description

A listing of the labels of interest. These will be (some of) the 'clean' entries in the finished lexicon. You should only name ROIs you specifically care about and which have a single, unambiguous occurence in the data set (e.g., 'Left_Parotid' is good, but 'JUNK' and 'Parotids' are bad – you won't be able to select the single 'JUNK' label if all you care about are parotids.

2.9.2.1.2 Default

- `"Body,Brainstem,Chiasm,Cord,Larynx Pharynx,Left Eye,Left Optic Nerve,Left Parotid,Left Submand,Left Temp Lobe,Oral Cavity,Right Eye,Right Optic Nerve,Right Parotid,Right Submand,Right Temp Lobe"`

2.9.2.1.3 Examples

- `"Left Parotid,Right Parotid,Left Submand,Right Submand"`
- `"Left Submand,Right Submand"`

2.9.2.2 JunkLabel

2.9.2.2.1 Description

A label to apply to the un-matched labels. This helps prevent false positives by excluding names which are close to a desired clean label. For example, if you are looking for 'Left_Parotid' you will want to mark 'left-parotid_opti' and 'OLDLeftParotid' as junk. Passing an empty string disables junk labeling.

2.9.2.2.2 Default

- `"JUNK"`

2.9.2.2.3 Examples

- `""`
- `"Junk"`
- `"Irrelevant"`
- `"NA_Organ"`

2.9.2.3 OmitROILabelRegex

2.9.2.3.1 Description

A regex matching ROI labels/names to prune. Only matching ROIs will be pruned. The default will match no ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax. (Note: an exclusive approach is taken rather than an inclusive approach because regex negations are not easily supported in the POSIX syntax.)

2.9.2.3.2 Default

- `""`

2.9.2.3.3 Examples

- `".*left.*|.*right.*|.*eyes.*"`
- `".*PTV.*|.*CTV.*|.*GTV.*"`

2.9.2.4 LexiconSeedFile

2.9.2.4.1 Description

A file containing a ‘seed’ lexicon to use and add to. This is the lexicon that is being built. It will be modified.

2.9.2.4.2 Default

- `""`

2.9.2.4.3 Examples

- `"./some_lexicon"`
- `"/tmp/temp_lexicon"`

2.10 CT_Liver_Perfusion

2.10.1 Description

This operation performed dynamic contrast-enhanced CT perfusion image modeling on a time series image volume.

2.10.2 Notes

- This routine is used for research purposes only.

2.10.3 Parameters

No registered options.

2.11 CT_Liver_Perfusion_First_Run

2.11.1 Description

This operation performed dynamic contrast-enhanced CT perfusion image modeling on a time series image volume.

2.11.2 Notes

- Use this mode when peeking at the data for the first time. It avoids computing much, just lets you *look* at the data, find t_0, etc..

2.11.3 Parameters

No registered options.

2.12 CT_Liver_Perfusion_Ortho_Views

2.12.1 Description

This operation performed dynamic contrast-enhanced CT perfusion image modeling on a time series image volume.

2.12.2 Notes

- Use this mode when you are only interested in oblique/orthogonal views. The point of this operation is to keep memory low so image sets can be compared.

2.12.3 Parameters

No registered options.

2.13 CT_Liver_Perfusion_Pharmaco_1C2I_5Param

2.13.1 Description

This operation performed dynamic contrast-enhanced CT perfusion image modeling on a time series image volume.

2.13.2 Parameters

- AIFROINameRegex
- ExponentialKernelCoeffTruncation
- FastChebyshevMultiplication
- PlotAIFVIF
- PlotPixelModel
- PreDecimateOutSizeR
- PreDecimateOutSizeC
- TargetROINameRegex
- UseBasisSplineInterpolation
- BasisSplineCoefficients
- BasisSplineOrder
- UseChebyshevPolyMethod

- ChebyshevPolyCoefficients
- VIFROINameRegex

2.13.2.1 AIFROINameRegex

2.13.2.1.1 Description

Regex for the name of the ROI to use as the AIF. It should generally be a major artery near the trunk or near the tissue of interest.

2.13.2.1.2 Default

- `"Abdominal_Aorta"`

2.13.2.1.3 Examples

- `"Abdominal_Aorta"`
- `".*Aorta.*"`
- `"Major_Artery"`

2.13.2.2 ExponentialKernelCoeffTruncation

2.13.2.2.1 Description

Control the number of Chebyshev coefficients used to approximate the exponential kernel. Usually ~10 will suffice. ~20 is probably overkill, and ~5 is probably too few. It is probably better to err on the side of caution and enlarge this number if you're worried about loss of precision – this will slow the computation somewhat. (You might be able to offset by retaining fewer coefficients in Chebyshev multiplication; see 'FastChebyshevMultiplication' parameter.)

2.13.2.2.2 Default

- `"10"`

2.13.2.2.3 Examples

- `"20"`
- `"15"`
- `"10"`
- `"5"`

2.13.2.3 FastChebyshevMultiplication

2.13.2.3.1 Description

Control coefficient truncation/pruning to speed up Chebyshev polynomial multiplication. (This setting does nothing if the Chebyshev method is not being used.) The choice of this number depends on how much precision you are willing to forgo. It also strongly depends on the number of datum in the AIF, VIF, and the number of coefficients used to approximate the exponential kernel (usually ~10 suffices). Numbers are specified relative to max(N,M), where N and M are the number of coefficients in the two Chebyshev expansions taking part in the multiplication. If too many coefficients are requested (i.e., more than (N+M-2)) then the full non-approximate multiplication is carried out.

2.13.2.3.2 Default

- `"*10000000.0"`

2.13.2.3.3 Examples

- `"*2.0"`
- `"*1.5"`
- `"*1.0"`
- `"*0.5"`
- `"*0.3"`

2.13.2.4 PlotAIFVIF

2.13.2.4.1 Description

Control whether the AIF and VIF should be shown prior to modeling.

2.13.2.4.2 Default

- `"false"`

2.13.2.4.3 Examples

- `"true"`
- `"false"`

2.13.2.5 PlotPixelModel

2.13.2.5.1 Description

Show a plot of the fitted model for a specified pixel. Plotting happens immediately after the pixel is processed. You can supply arbitrary metadata, but must also supply Row and Column numbers. Note that numerical comparisons are performed lexically, so you have to be exact. Also note the sub-separation token is a semi-colon, not a colon.

2.13.2.5.2 Default

- `""`

2.13.2.5.3 Examples

- `"Row@12;Column@4;Description@.*k1A.*"`
- `"Row@256;Column@500;SliceLocation@23;SliceThickness@0.5"`
- `"Row@256;Column@500;Some@thing#Row@256;Column@501;Another@thing"`
- `"Row@0;Column@5#Row@4;Column@5#Row@8;Column@5#Row@12;Column@5"`

2.13.2.6 PreDecimateOutSizeR

2.13.2.6.1 Description

The number of pixels along the row unit vector to group into an outgoing pixel. This optional step can reduce computation effort by downsampling (decimating) images before computing fitted parameter maps (but *after* computing AIF and VIF time courses). Must be a multiplicative factor of the incoming image's row count. No decimation occurs if either this or 'PreDecimateOutSizeC' is zero or negative.

2.13.2.6.2 Default

- `"8"`

2.13.2.6.3 Examples

- `"0"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`
- `"32"`
- `"64"`
- `"128"`
- `"256"`

- "512"

2.13.2.7 PreDecimateOutSizeC

2.13.2.7.1 Description

The number of pixels along the column unit vector to group into an outgoing pixel. This optional step can reduce computation effort by downsampling (decimating) images before computing fitted parameter maps (but *after* computing AIF and VIF time courses). Must be a multiplicative factor of the incoming image's column count. No decimation occurs if either this or 'PreDecimateOutSizeR' is zero or negative.

2.13.2.7.2 Default

- "8"

2.13.2.7.3 Examples

- "0"
- "2"
- "4"
- "8"
- "16"
- "32"
- "64"
- "128"
- "256"
- "512"

2.13.2.8 TargetROINameRegex

2.13.2.8.1 Description

Regex for the name of the ROI to perform modeling within. The largest contour is usually what you want, but you can also be more focused.

2.13.2.8.2 Default

- ".*Body.*"

2.13.2.8.3 Examples

- `"Liver_Patches_For_Testing_Smaller"`
- `"Liver_Patches_For_Testing"`
- `"Suspected_Liver_Rough"`
- `"Rough_Body"`
- `".*body.*"`
- `".*something.*\|.*another.*thing.*"`

2.13.2.9 UseBasisSplineInterpolation

2.13.2.9.1 Description

Control whether the AIF and VIF should use basis spline interpolation in conjunction with the Chebyshev polynomial method. If this option is not set, linear interpolation is used instead. Linear interpolation may result in a less-smooth AIF and VIF (and therefore possibly slower optimizer convergence), but is safer if you cannot verify the AIF and VIF plots are reasonable. This option currently produces an effect only if the Chebyshev polynomial method is being used.

2.13.2.9.2 Default

- `"false"`

2.13.2.9.3 Examples

- `"true"`
- `"false"`

2.13.2.10 BasisSplineCoefficients

2.13.2.10.1 Description

Control the number of basis spline coefficients to use, if applicable. (This setting does nothing when basis splines are not being used.) Valid options for this setting depend on the amount of data and b-spline order. This number controls the number of coefficients that are fitted (via least-squares). You must verify that overfitting is not happening. If in doubt, use fewer coefficients. There are two ways to specify the number: relative and absolute. Relative means relative to the number of datum. For example, if the AIF and VIF have ~40 datum then generally ‘*0.5’ is safe. (*’0.5’ means there are half the number of coefficients as datum.) Inspect for overfitting and poor fit. Because this routine happens once and is fast, do not tweak to optimize for speed; the aim of this method is to produce a smooth and accurate AIF and VIF. Because an integer number of coefficients are needed, so rounding is used. You can also specify the absolute number of coefficients to use like ‘20’. It often makes more sense to use relative specification. Be aware that not all inputs can be honoured due to limits on b-spline knots and breaks, and may cause unpredictable behaviour or internal failure.

2.13.2.10.2 Default

- `"*0.5"`

2.13.2.10.3 Examples

- `"*0.8"`
- `"*0.5"`
- `"*0.3"`
- `"20.0"`
- `"10.0"`

2.13.2.11 BasisSplineOrder

2.13.2.11.1 Description

Control the polynomial order of basis spline interpolation to use, if applicable. (This setting does nothing when basis splines are not being used.) This parameter controls the order of polynomial used for b-spline interpolation, and therefore has ramifications for the computability and numerical stability of AIF and VIF derivatives. Stick with ‘4’ or ‘5’ if you’re unsure.

2.13.2.11.2 Default

- `"4"`

2.13.2.11.3 Examples

- `"1"`
- `"2"`
- `"3"`
- `"4"`
- `"5"`
- `"6"`
- `"7"`
- `"8"`
- `"9"`

- `"10"`

2.13.2.12 UseChebyshevPolyMethod

2.13.2.12.1 Description

Control whether the AIF and VIF should be approximated by Chebyshev polynomials. If this option is not set, a inear interpolation approach is used instead.

2.13.2.12.2 Default

- `"true"`

2.13.2.12.3 Examples

- `"true"`
- `"false"`

2.13.2.13 ChebyshevPolyCoefficients

2.13.2.13.1 Description

Control the number of Chebyshev polynomial coefficients to use, if applicable. (This setting does nothing when the Chebyshev polynomial method is not being used.) This number controls the number of coefficients that are computed. There are two ways to specify the number: relative and absolute. Relative means relative to the number of datum. For example, if the AIF and VIF have ~40 datum then generally ‘*2’ is safe. (’*2’ means there are 2x the number of coefficients as datum; usually overkill.) A good middle-ground is ’*1’ which is faster but should produce similar results. For speed ‘/2’ is even faster, but can produce bad results in some cases. Because an integer number of coefficients are needed, rounding is used. You can also specify the absolute number of coefficients to use like ‘20’. It often makes more sense to use relative specification. Be aware that not all inputs can be honoured (i.e., too large, too small, or negative), and may cause unpredictable behaviour or internal failure.

2.13.2.13.2 Default

- `"*2.0"`

2.13.2.13.3 Examples

- `"*10.0"`
- `"*5.0"`
- `"*2.0"`
- `"*1.23"`
- `"*1.0"`
- `"/1.0"`
- `"/2.0"`
- `"/3.0"`
- `"/5.0"`
- `"100.0"`
- `"50.0"`
- `"20"`
- `"10.01"`

2.13.2.14 VIFROINameRegex

2.13.2.14.1 Description

Regex for the name of the ROI to use as the VIF. It should generally be a major vein near the trunk or near the tissue of interest.

2.13.2.14.2 Default

- `"Hepatic_Portal_Vein"`

2.13.2.14.3 Examples

- `"Hepatic_Portal_Vein"`
- `".*Portal.*Vein.*"`
- `"Major_Vein"`

2.14 CT_Liver_Perfusion_Pharmaco_1C2I_Reduced3Param

2.14.1 Description

This operation performed dynamic contrast-enhanced CT perfusion image modeling on a time series image volume.

2.14.2 Parameters

- AIFROINameRegex

- ExponentialKernelCoeffTruncation
- FastChebyshevMultiplication
- PlotAIFVIF
- PlotPixelModel
- PreDecimateOutSizeR
- PreDecimateOutSizeC
- TargetROINameRegex
- UseBasisSplineInterpolation
- BasisSplineCoefficients
- BasisSplineOrder
- ChebyshevPolyCoefficients
- VIFROINameRegex

2.14.2.1 AIFROINameRegex

2.14.2.1.1 Description

Regex for the name of the ROI to use as the AIF. It should generally be a major artery near the trunk or near the tissue of interest.

2.14.2.1.2 Default

- `"Abdominal_Aorta"`

2.14.2.1.3 Examples

- `"Abdominal_Aorta"`
- `".*Aorta.*"`
- `"Major_Artery"`

2.14.2.2 ExponentialKernelCoeffTruncation

2.14.2.2.1 Description

Control the number of Chebyshev coefficients used to approximate the exponential kernel. Usually ~10 will suffice. ~20 is probably overkill, and ~5 is probably too few. It is probably better to err on the side of caution and enlarge this number if you're worried about loss of precision – this will slow the computation somewhat. (You might be able to offset by retaining fewer coefficients in Chebyshev multiplication; see 'FastChebyshevMultiplication' parameter.)

2.14.2.2.2 Default

- `"10"`

2.14.2.2.3 Examples

- `"20"`
- `"15"`
- `"10"`
- `"5"`

2.14.2.3 FastChebyshevMultiplication

2.14.2.3.1 Description

Control coefficient truncation/pruning to speed up Chebyshev polynomial multiplication. (This setting does nothing if the Chebyshev method is not being used.) The choice of this number depends on how much precision you are willing to forgo. It also strongly depends on the number of datum in the AIF, VIF, and the number of coefficients used to approximate the exponential kernel (usually ~10 suffices). Numbers are specified relative to max(N,M), where N and M are the number of coefficients in the two Chebyshev expansions taking part in the multiplication. If too many coefficients are requested (i.e., more than (N+M-2)) then the full non-approximate multiplication is carried out.

2.14.2.3.2 Default

- `"*10000000.0"`

2.14.2.3.3 Examples

- `"*2.0"`
- `"*1.5"`
- `"*1.0"`
- `"*0.5"`
- `"*0.3"`

2.14.2.4 PlotAIFVIF

2.14.2.4.1 Description

Control whether the AIF and VIF should be shown prior to modeling.

2.14.2.4.2 Default

- `"false"`

2.14.2.4.3 Examples

- `"true"`
- `"false"`

2.14.2.5 PlotPixelModel

2.14.2.5.1 Description

Show a plot of the fitted model for a specified pixel. Plotting happens immediately after the pixel is processed. You can supply arbitrary metadata, but must also supply Row and Column numbers. Note that numerical comparisons are performed lexically, so you have to be exact. Also note the sub-separation token is a semi-colon, not a colon.

2.14.2.5.2 Default

- `""`

2.14.2.5.3 Examples

- `"Row@12;Column@4;Description@.*k1A.*"`
- `"Row@256;Column@500;SliceLocation@23;SliceThickness@0.5"`
- `"Row@256;Column@500;Some@thing#Row@256;Column@501;Another@thing"`
- `"Row@0;Column@5#Row@4;Column@5#Row@8;Column@5#Row@12;Column@5"`

2.14.2.6 PreDecimateOutSizeR

2.14.2.6.1 Description

The number of pixels along the row unit vector to group into an outgoing pixel. This optional step can reduce computation effort by downsampling (decimating) images before computing fitted parameter maps (but *after* computing AIF and VIF time courses). Must be a multiplicative factor of the incoming image's row count. No decimation occurs if either this or 'PreDecimateOutSizeC' is zero or negative.

2.14.2.6.2 Default

- `"8"`

2.14.2.6.3 Examples

- `"0"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`
- `"32"`
- `"64"`
- `"128"`
- `"256"`
- `"512"`

2.14.2.7 PreDecimateOutSizeC

2.14.2.7.1 Description

The number of pixels along the column unit vector to group into an outgoing pixel. This optional step can reduce computation effort by downsampling (decimating) images before computing fitted parameter maps (but *after* computing AIF and VIF time courses). Must be a multiplicative factor of the incoming image's column count. No decimation occurs if either this or 'PreDecimateOutSizeR' is zero or negative.

2.14.2.7.2 Default

- `"8"`

2.14.2.7.3 Examples

- `"0"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`
- `"32"`
- `"64"`
- `"128"`
- `"256"`
- `"512"`

2.14.2.8 TargetROINameRegex

2.14.2.8.1 Description

Regex for the name of the ROI to perform modeling within. The largest contour is usually what you want, but you can also be more focused.

2.14.2.8.2 Default

- `".*Body.*"`

2.14.2.8.3 Examples

- `"Liver_Patches_For_Testing_Smaller"`
- `"Liver_Patches_For_Testing"`
- `"Suspected_Liver_Rough"`
- `"Rough_Body"`
- `".*body.*"`
- `".*something.*\|.*another.*thing.*"`

2.14.2.9 UseBasisSplineInterpolation

2.14.2.9.1 Description

Control whether the AIF and VIF should use basis spline interpolation in conjunction with the Chebyshev polynomial method. If this option is not set, linear interpolation is used instead. Linear interpolation may result in a less-smooth AIF and VIF (and therefore possibly slower optimizer convergence), but is safer if you cannot verify the AIF and VIF plots are reasonable. This option currently produces an effect only if the Chebyshev polynomial method is being used.

2.14.2.9.2 Default

- `"false"`

2.14.2.9.3 Examples

- `"true"`
- `"false"`

2.14.2.10 BasisSplineCoefficients

2.14.2.10.1 Description

Control the number of basis spline coefficients to use, if applicable. (This setting does nothing when basis splines are not being used.) Valid options for this setting depend on the amount of data and b-spline order. This number controls the number of coefficients that are fitted (via least-squares). You must verify that overfitting is not happening. If in doubt, use fewer coefficients. There are two ways to specify the number: relative and absolute. Relative means relative to the number of datum. For example, if the AIF and VIF have ~40 datum then generally ‘*0.5’ is safe. (’*0.5’ means there are half the number of coefficients as datum.) Inspect for overfitting and poor fit. Because this routine happens once and is fast, do not tweak to optimize for speed; the aim of this method is to produce a smooth and accurate AIF and VIF. Because an integer number of coefficients are needed, so rounding is used. You can also specify the absolute number of coefficients to use like ‘20’. It often makes more sense to use relative specification. Be aware that not all inputs can be honoured due to limits on b-spline knots and breaks, and may cause unpredictable behaviour or internal failure.

2.14.2.10.2 Default

- `"*0.5"`

2.14.2.10.3 Examples

- `"*0.8"`
- `"*0.5"`
- `"*0.3"`
- `"20.0"`
- `"10.0"`

2.14.2.11 BasisSplineOrder

2.14.2.11.1 Description

Control the polynomial order of basis spline interpolation to use, if applicable. (This setting does nothing when basis splines are not being used.) This parameter controls the order of polynomial used for b-spline interpolation, and therefore has ramifications for the computability and numerical stability of AIF and VIF derivatives. Stick with ‘4’ or ‘5’ if you’re unsure.

2.14.2.11.2 Default

- `"4"`

2.14.2.11.3 Examples

- `"1"`
- `"2"`
- `"3"`
- `"4"`
- `"5"`
- `"6"`
- `"7"`
- `"8"`
- `"9"`
- `"10"`

2.14.2.12 ChebyshevPolyCoefficients

2.14.2.12.1 Description

Control the number of Chebyshev polynomial coefficients to use, if applicable. (This setting does nothing when the Chebyshev polynomial method is not being used.) This number controls the number of coefficients that are computed. There are two ways to specify the number: relative and absolute. Relative means relative to the number of datum. For example, if the AIF and VIF have ~40 datum then generally ‘*2’ is safe. (*’2’ means there are 2x the number of coefficients as datum; usually overkill.) A good middle-ground is ’*1’ which is faster but should produce similar results. For speed ‘/2’ is even faster, but can produce bad results in some cases. Because an integer number of coefficients are needed, rounding is used. You can also specify the absolute number of coefficients to use like ‘20’. It often makes more sense to use relative specification. Be aware that not all inputs can be honoured (i.e., too large, too small, or negative), and may cause unpredictable behaviour or internal failure.

2.14.2.12.2 Default

- `"*2.0"`

2.14.2.12.3 Examples

- `"*10.0"`
- `"*5.0"`
- `"*2.0"`
- `"*1.23"`
- `"*1.0"`
- `"/1.0"`
- `"/2.0"`
- `"/3.0"`

- `"/5.0"`
- `"100.0"`
- `"50.0"`
- `"20"`
- `"10.01"`

2.14.2.13 VIFROINameRegex

2.14.2.13.1 Description

Regex for the name of the ROI to use as the VIF. It should generally be a major vein near the trunk or near the tissue of interest.

2.14.2.13.2 Default

- `"Hepatic_Portal_Vein"`

2.14.2.13.3 Examples

- `"Hepatic_Portal_Vein"`
- `".*Portal.*Vein.*"`
- `"Major_Vein"`

2.15 ClusterDBSCAN

2.15.1 Description

This routine performs DBSCAN clustering on an image volume. The clustering is limited within ROI(s) and also within a range of voxel intensities. Voxels values are overwritten with the cluster ID (if applicable) or a generic configurable background value.

2.15.2 Notes

- This operation will work with single images and image volumes. Images need not be rectilinear.

2.15.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex

- ContourOverlap
- Inclusivity
- Channel
- Lower
- Upper
- MinPoints
- MaxPoints
- Eps
- BackgroundValue
- Reduction

2.15.3.1 ImageSelection

2.15.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.15.3.1.2 Default

- `"last"`

2.15.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`

- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.15.3.2 NormalizedROILabelRegex

2.15.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.15.3.2.2 Default

- `".*"`

2.15.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.15.3.3 ROILabelRegex

2.15.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.15.3.3.2 Default

- `".*"`

2.15.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.15.3.4 ContourOverlap

2.15.3.4.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.15.3.4.2 Default

- `"ignore"`

2.15.3.4.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.15.3.5 Inclusivity

2.15.3.5.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.15.3.5.2 Default

- `"center"`

2.15.3.5.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.15.3.6 Channel

2.15.3.6.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.15.3.6.2 Default

- `"0"`

2.15.3.6.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.15.3.7 Lower

2.15.3.7.1 Description

Lower threshold (inclusive) below which voxels will be ignored by this routine.

2.15.3.7.2 Default

- `"-inf"`

2.15.3.7.3 Examples

- `"-inf"`
- `"0.0"`
- `"1024"`

2.15.3.8 Upper

2.15.3.8.1 Description

Upper threshold (inclusive) above which voxels will be ignored by this routine.

2.15.3.8.2 Default

- `"inf"`

2.15.3.8.3 Examples

- `"inf"`
- `"1.0"`
- `"2048"`

2.15.3.9 MinPoints

2.15.3.9.1 Description

DBSCAN algorithm parameter representing the minimum number of points that must appear in the vicinity for a cluster to be recognized. Sanders, et al. (1998) recommend a default of twice the dimensionality, but what is considered to be a reasonable value depends on the sparsity of the inputs and geometry. For regular grids, a slightly smaller value might be more appropriate.

2.15.3.9.2 Default

- `"5"`

2.15.3.9.3 Examples

- `"4"`
- `"6"`
- `"15"`

2.15.3.10 MaxPoints

2.15.3.10.1 Description

Reject clusters if they would contain more than this many members. This parameter can be used to reject irrelevant background clusters or to help search for disconnected clusters. Setting this parameter appropriately will improve both memory usage and runtime considerably.

2.15.3.10.2 Default

- `"inf"`

2.15.3.10.3 Examples

- `"10"`
- `"1000"`
- `"1E6"`
- `"inf"`

2.15.3.11 Eps

2.15.3.11.1 Description

DBSCAN algorithm parameter representing the threshold separation distance (in DICOM units; mm) between members of a cluster. All members in a cluster must be separated from at least MinPoints points within a distance of Eps. There is a standard way to determine an optimal value from the data itself, but requires generating a k-nearest-neighbours clustering first, and then visually identifying an appropriate 'kink' in the k-distances plot. This approach is not implemented here. Alternatively, the sparsity of the data and the specific problem domain must be used to estimate a desirable separation Eps.

2.15.3.11.2 Default

- `"4.0"`

2.15.3.11.3 Examples

- `"1.5"`
- `"2.5"`
- `"4.0"`
- `"10.0"`

2.15.3.12 BackgroundValue

2.15.3.12.1 Description

The voxel intensity that will be assigned to all voxels that are not members of a cluster. Note that this value can be anything, but cluster numbers are zero-based, so a negative background is probably desired.

2.15.3.12.2 Default

- `"-1.0"`

2.15.3.12.3 Examples

- `"-1.0"`
- `"0.0"`
- `"100.23"`
- `"nan"`
- `"-inf"`

2.15.3.13 Reduction

2.15.3.13.1 Description

Voxels within a cluster can be marked as-is, or reduced in a variety of ways. If reduction is not used, voxels in a valid cluster will have their values replaced with the cluster ID number. If 'median' reduction is specified, the component-wise median is reported for each cluster; the x-, y-, and z-coordinates of all voxels in each individual cluster will be reduced to the median coordinate.

2.15.3.13.2 Default

- `"none"`

2.15.3.13.3 Examples

- `"none"`
- `"median"`

2.16 ComparePixels

2.16.1 Description

This operation compares images ('test' images and 'reference' images) on a per-voxel/per-pixel basis. Any combination of 2D and 3D images is supported, including images which do not fully overlap, but the reference image array must be rectilinear (this property is verified).

2.16.2 Notes

- Images are overwritten, but ReferenceImages are not. Multiple Images may be specified, but only one ReferenceImages may be specified.
- The reference image array must be rectilinear. (This is a requirement specific to this implementation, a less restrictive implementation could overcome the issue.)
- For the fastest and most accurate results, test and reference image arrays should spatially align. However, alignment is **not** necessary. If test and reference image arrays are aligned, image adjacency can be precomputed and the analysis will be faster. If not, image adjacency must be evaluated for every voxel.
- The distance-to-agreement comparison will tend to overestimate the distance, especially when the DTA value is low, because voxel size effects will dominate the estimation. Reference images should be supersampled as necessary.
- This operation optionally makes use of interpolation for sub-voxel distance estimation. However, interpolation is currently limited to be along the edges connecting nearest- and next-nearest voxel centres. In other words, true volumetric interpolation is **not** available. Implicit interpolation is also used (via the intermediate value theorem) for the distance-to-agreement comparison, which results in distance estimation that may vary up to the largest caliper distance of a voxel. For this reason, the accuracy of all comparisons should be expected to be limited by image spatial resolution (i.e., voxel dimensions). Reference images should be supersampled as necessary.

2.16.3 Parameters

- ImageSelection
- ReferenceImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Method
- Channel
- TestImgLowerThreshold
- TestImgUpperThreshold
- RefImgLowerThreshold
- RefImgUpperThreshold

- DiscType
- DTAVoxValEqAbs
- DTAVoxValEqRelDiff
- DTAMax
- DTAInterpolationMethod
- GammaDTAThreshold
- GammaDiscThreshold
- GammaTerminateAboveOne

2.16.3.1 ImageSelection

2.16.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.16.3.1.2 Default

- `"all"`

2.16.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.16.3.2 ReferenceImageSelection

2.16.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.16.3.2.2 Default

- `"all"`

2.16.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.16.3.3 NormalizedROILabelRegex

2.16.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.16.3.3.2 Default

- `".*"`

2.16.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.16.3.4 ROILabelRegex

2.16.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.16.3.4.2 Default

- `".*"`

2.16.3.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.16.3.5 Method

2.16.3.5.1 Description

The comparison method to compute. Three options are currently available: distance-to-agreement (DTA), discrepancy, and gamma-index. All three are fully 3D, but can also work for 2D or mixed 2D-3D comparisons. DTA is a measure of how far away the nearest voxel (in the reference images) is with a voxel intensity sufficiently close to each voxel in the test images. This comparison ignores pixel intensities except to test if the values match within the specified tolerance. The voxel neighbourhood is exhaustively explored until a suitable voxel is found. Implicit interpolation is used to detect when the value could be found via interpolation, but explicit interpolation is not used. Thus distance might be overestimated. A discrepancy comparison measures the point intensity discrepancy without accounting for spatial shifts. A gamma analysis combines distance-to-agreement and point differences into a single index which is best used to test if both DTA and discrepancy criteria are satisfied (gamma <= 1 iff both pass). It was proposed by Low et al. in 1998 ((doi:10.1118/1.598248). Gamma analyses permits trade-offs between spatial and dosimetric discrepancies which can arise when the image arrays slightly differ in alignment or pixel values.

2.16.3.5.2 Default

- `"gamma-index"`

2.16.3.5.3 Examples

- `"gamma-index"`
- `"DTA"`
- `"discrepancy"`

2.16.3.6 Channel

2.16.3.6.1 Description

The channel to compare (zero-based). Note that both test images and reference images will share this specifier.

2.16.3.6.2 Default

- `"0"`

2.16.3.6.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.16.3.7 TestImgLowerThreshold

2.16.3.7.1 Description

Pixel lower threshold for the test images. Only voxels with values above this threshold (inclusive) will be altered.

2.16.3.7.2 Default

- `"-inf"`

2.16.3.7.3 Examples

- `"-inf"`
- `"0.0"`
- `"200"`

2.16.3.8 TestImgUpperThreshold

2.16.3.8.1 Description

Pixel upper threshold for the test images. Only voxels with values below this threshold (inclusive) will be altered.

2.16.3.8.2 Default

- `"inf"`

2.16.3.8.3 Examples

- `"inf"`
- `"1.23"`
- `"1000"`

2.16.3.9 RefImgLowerThreshold

2.16.3.9.1 Description

Pixel lower threshold for the reference images. Only voxels with values above this threshold (inclusive) will be altered.

2.16.3.9.2 Default

- `"-inf"`

2.16.3.9.3 Examples

- `"-inf"`
- `"0.0"`
- `"200"`

2.16.3.10 RefImgUpperThreshold

2.16.3.10.1 Description

Pixel upper threshold for the reference images. Only voxels with values below this threshold (inclusive) will be altered.

2.16.3.10.2 Default

- `"inf"`

2.16.3.10.3 Examples

- `"inf"`
- `"1.23"`
- `"1000"`

2.16.3.11 DiscType

2.16.3.11.1 Description

Parameter for all comparisons estimating the direct, voxel-to-voxel discrepancy. There are currently three types available. ‘Relative’ is the absolute value of the difference of two voxel values divided by the largest of the two values. ‘Difference’ is the difference of two voxel values. ‘PinnedToMax’ is the absolute value of the difference of two voxel values divided by the largest voxel value in the selected images.

2.16.3.11.2 Default

- `"relative"`

2.16.3.11.3 Examples

- `"relative"`
- `"difference"`
- `"pinned-to-max"`

2.16.3.12 DTAVoxValEqAbs

2.16.3.12.1 Description

Parameter for all comparisons involving a distance-to-agreement (DTA) search. The difference in voxel values considered to be sufficiently equal (absolute; in voxel intensity units). Note: This value CAN be zero. It is meant to help overcome noise. Note that this value is ignored by all interpolation methods.

2.16.3.12.2 Default

- `"1.0E-3"`

2.16.3.12.3 Examples

- `"1.0E-3"`
- `"1.0E-5"`
- `"0.0"`
- `"0.5"`

2.16.3.13 DTAVoxValEqRelDiff

2.16.3.13.1 Description

Parameter for all comparisons involving a distance-to-agreement (DTA) search. The difference in voxel values considered to be sufficiently equal (~relative difference; in %). Note: This value CAN be zero. It is meant to help overcome noise. Note that this value is ignored by all interpolation methods.

2.16.3.13.2 Default

- `"1.0"`

2.16.3.13.3 Examples

- `"0.1"`
- `"1.0"`
- `"10.0"`

2.16.3.14 DTAMax

2.16.3.14.1 Description

Parameter for all comparisons involving a distance-to-agreement (DTA) search. Maximally acceptable distance-to-agreement (in DICOM units: mm) above which to stop searching. All voxels within this distance will be searched unless a matching voxel is found. Note that a gamma-index comparison may terminate this search early if the gamma-index is known to be greater than one. It is recommended to make this value approximately 1 voxel width larger than necessary in case a matching voxel can be located near the boundary. Also note that some voxels beyond the DTA_max distance may be evaluated.

2.16.3.14.2 Default

- `"30.0"`

2.16.3.14.3 Examples

- `"3.0"`
- `"5.0"`
- `"50.0"`

2.16.3.15 DTAInterpolationMethod

2.16.3.15.1 Description

Parameter for all comparisons involving a distance-to-agreement (DTA) search. Controls how precisely and how often the space between voxel centres are interpolated to identify the exact position of agreement. There are currently three options: no interpolation ('None'), nearest-neighbour ('NN'), and next-nearest-neighbour ('NNN'). (1) If no interpolation is selected, the agreement position will only be established to within approximately the reference image voxels dimensions. To avoid interpolation, voxels that straddle the target value are taken as the agreement distance. Conceptually, if you view a voxel as having a finite spatial extent then this method may be sufficient for distance assessment. Though it is not precise, it is fast. This method will tend to over-estimate the actual distance, though it is possible that it slightly under-estimates it. This method works best when the reference image grid size is small in comparison to the desired spatial accuracy (e.g., if computing gamma, the tolerance should be much larger than the largest voxel dimension) so supersampling is recommended. (2) Nearest-neighbour interpolation considers the line connecting directly adjacent voxels. Using linear interpolation along this line when adjacent voxels straddle the target value, the 3D point where the target value appears can be predicted. This method can significantly improve distance estimation accuracy, though will typically be much slower than no interpolation. On the other hand, this method lower amounts of supersampling, though it is most reliable when the reference image grid size is small in comparison to the desired spatial accuracy. Note that nearest-neighbour interpolation also makes use of the ‘no interpolation’ methods. If you have a fine reference image, prefer either no interpolation or nearest-neighbour interpolation. (3) Finally, next-nearest-neighbour considers the diagonally-adjacent neighbours separated by taxi-cab distance of 2 (so in-plane diagonals are considered, but 3D diagonals are not). Quadratic (i.e., bi-linear) interpolation is analytically solved to determine where along the straddling diagonal the target value appears. This method is more expensive than linear interpolation but will generally result in more accurate distance estimates. This method may require lower amounts of supersampling than linear interpolation, but is most reliable when the reference image grid size is small in comparison to the desired spatial accuracy. Use of this method may not be appropriate in all cases considering that supersampling may be needed and a quadratic equation is solved for every voxel diagonal. Note that next-nearest-neighbour interpolation also makes use of the nearest-neighbour and ‘no interpolation’ methods.

2.16.3.15.2 Default

- `"NN"`

2.16.3.15.3 Examples

- `"None"`
- `"NN"`
- `"NNN"`

2.16.3.16 GammaDTAThreshold

2.16.3.16.1 Description

Parameter for gamma-index comparisons. Maximally acceptable distance-to-agreement (in DICOM units: mm). When the measured DTA is above this value, the gamma index will necessarily be greater than one. Note this parameter can differ from the DTA_max search cut-off, but should be <= to it.

2.16.3.16.2 Default

- `"5.0"`

2.16.3.16.3 Examples

- `"3.0"`
- `"5.0"`
- `"10.0"`

2.16.3.17 GammaDiscThreshold

2.16.3.17.1 Description

Parameter for gamma-index comparisons. Voxel value discrepancies lower than this value are considered acceptable, but values above will result in gamma values >1. The specific interpretation of this parameter (and the units) depend on the specific type of discrepancy used. For percentage-based discrepancies, this parameter is interpretted as a percentage (i.e., '5.0' = '5%'). For voxel intensity measures such as the absolute difference, this value is interpretted as an absolute threshold with the same intensity units (i.e., '5.0' = '5 HU' or similar).

2.16.3.17.2 Default

- `"5.0"`

2.16.3.17.3 Examples

- `"3.0"`
- `"5.0"`
- `"10.0"`

2.16.3.18 GammaTerminateAboveOne

2.16.3.18.1 Description

Parameter for gamma-index comparisons. Halt spatial searching if the gamma index will necessarily indicate failure (i.e., gamma >1). Note this can parameter can drastically reduce the computational effort required to compute the gamma index, but the reported gamma values will be invalid whenever they are >1. This is often tolerable since the magnitude only matters when it is <1. In lieu of the true gamma-index, a value slightly >1 will be assumed.

2.16.3.18.2 Default

- `"true"`

2.16.3.18.3 Examples

- `"true"`
- `"false"`

2.17 ContourBasedRayCastDoseAccumulate

2.17.1 Description

This operation performs ray-casting to estimate the dose of a surface. The surface is represented as a set of contours (i.e., an ROI).

2.17.2 Parameters

- DoseLengthMapFileName
- LengthMapFileName
- NormalizedROILabelRegex
- ROILabelRegex
- CylinderRadius
- RaydL
- Rows
- Columns

2.17.2.1 DoseLengthMapFileName

2.17.2.1.1 Description

A filename (or full path) for the (dose)*(length traveled through the ROI peel) image map. The format is TBD. Leave empty to dump to generate a unique temporary file.

2.17.2.1.2 Default

- `""`

2.17.2.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.img"`
- `"derivative_data.img"`

2.17.2.2 LengthMapFileName

2.17.2.2.1 Description

A filename (or full path) for the (length traveled through the ROI peel) image map. The format is TBD. Leave empty to dump to generate a unique temporary file.

2.17.2.2.2 Default

- `""`

2.17.2.2.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.img"`
- `"derivative_data.img"`

2.17.2.3 NormalizedROILabelRegex

2.17.2.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.17.2.3.2 Default

- `".*"`

2.17.2.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.17.2.4 ROILabelRegex

2.17.2.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.17.2.4.2 Default

- `".*"`

2.17.2.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.17.2.5 CylinderRadius

2.17.2.5.1 Description

The radius of the cylinder surrounding contour line segments that defines the 'surface'. Quantity is in the DICOM coordinate system.

2.17.2.5.2 Default

- `"3.0"`

2.17.2.5.3 Examples

- `"1.0"`
- `"2.0"`
- `"0.5"`
- `"5.0"`

2.17.2.6 RaydL

2.17.2.6.1 Description

The distance to move a ray each iteration. Should be << img_thickness and << cylinder_radius. Making too large will invalidate results, causing rays to pass through the surface without registering any dose accumulation. Making too small will cause the run-time to grow and may eventually lead to truncation or round-off errors. Quantity is in the DICOM coordinate system.

2.17.2.6.2 Default

- `"0.1"`

2.17.2.6.3 Examples

- `"0.1"`
- `"0.05"`
- `"0.01"`
- `"0.005"`

2.17.2.7 Rows

2.17.2.7.1 Description

The number of rows in the resulting images.

2.17.2.7.2 Default

- `"256"`

2.17.2.7.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.17.2.8 Columns

2.17.2.8.1 Description

The number of columns in the resulting images.

2.17.2.8.2 Default

- `"256"`

2.17.2.8.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.18 ContourBooleanOperations

2.18.1 Description

This routine performs 2D Boolean operations on user-provided sets of ROIs. The ROIs themselves are planar contours embedded in R^3, but the Boolean operation is performed once for each 2D plane where the selected ROIs reside. This routine can only perform Boolean operations on co-planar contours. This routine can operate on single contours (rather than ROIs composed of several contours) by simply presenting this routine with a single contour to select.

2.18.2 Notes

- This routine DOES support disconnected ROIs, such as left- and right-parotid contours that have been joined into a single 'parotids' ROI.
- Many Boolean operations can produce contours with holes. This operation currently connects the interior and exterior with a seam so that holes can be represented by a single polygon (rather than a separate hole polygon). It *is* possible to export holes as contours with a negative orientation, but this was not needed when writing.
- Only the common metadata between contours is propagated to the product contours.

2.18.3 Parameters

- ROILabelRegexA
- ROILabelRegexB
- NormalizedROILabelRegexA
- NormalizedROILabelRegexB
- Operation
- OutputROILabel

2.18.3.1 ROILabelRegexA

2.18.3.1.1 Description

A regex matching ROI labels/names that comprise the set of contour polygons ‘A’ as in f(A,B) where f is some Boolean operation. The default with match all available ROIs, which is probably not what you want.

2.18.3.1.2 Default

- `".*"`

2.18.3.1.3 Examples

- `".*"`
- `".*[pP]rostate.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.18.3.2 ROILabelRegexB

2.18.3.2.1 Description

A regex matching ROI labels/names that comprise the set of contour polygons ‘B’ as in f(A,B) where f is some Boolean operation. The default with match all available ROIs, which is probably not what you want.

2.18.3.2.2 Default

- `".*"`

2.18.3.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.18.3.3 NormalizedROILabelRegexA

2.18.3.3.1 Description

A regex matching ROI labels/names that comprise the set of contour polygons 'A' as in f(A,B) where f is some Boolean operation. The regex is applied to normalized ROI labels/names, which are translated using a user-provided lexicon (i.e., a dictionary that supports fuzzy matching). The default with match all available ROIs, which is probably not what you want.

2.18.3.3.2 Default

- `".*"`

2.18.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.18.3.4 NormalizedROILabelRegexB

2.18.3.4.1 Description

A regex matching ROI labels/names that comprise the set of contour polygons 'B' as in f(A,B) where f is some Boolean operation. The regex is applied to normalized ROI labels/names, which are translated using a user-provided lexicon (i.e., a dictionary that supports fuzzy matching). The default with match all available ROIs, which is probably not what you want.

2.18.3.4.2 Default

- `".*"`

2.18.3.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.18.3.5 Operation

2.18.3.5.1 Description

The Boolean operation (e.g., the function ‘f’) to perform on the sets of contour polygons ‘A’ and ‘B’. ‘Symmetric difference’ is also known as ‘XOR’.

2.18.3.5.2 Default

- `"join"`

2.18.3.5.3 Examples

- `"intersection"`
- `"join"`
- `"difference"`
- `"symmetric_difference"`

2.18.3.6 OutputROILabel

2.18.3.6.1 Description

The label to attach to the ROI contour product of f(A,B).

2.18.3.6.2 Default

- `"Boolean_result"`

2.18.3.6.3 Examples

- `"A+B"`
- `"A-B"`
- `"AuB"`
- `"AnB"`
- `"AxB"`
- `"A^B"`
- `"union"`
- `"xor"`
- `"combined"`
- `"body_without_spinal_cord"`

2.19 ContourSimilarity

2.19.1 Description

This operation estimates the similarity or overlap between two sets of contours. The comparison is based on point samples. It is useful for comparing contouring styles.

2.19.2 Notes

- This routine requires an image grid, which is used to control where the contours are sampled. Images are not modified.

2.19.3 Parameters

- ImageSelection
- NormalizedROILabelRegexA
- ROILabelRegexA
- NormalizedROILabelRegexB
- ROILabelRegexB
- FileName
- UserComment

2.19.3.1 ImageSelection

2.19.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.19.3.1.2 Default

- `"last"`

2.19.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.19.3.2 NormalizedROILabelRegexA

2.19.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.19.3.2.2 Default

- `".*"`

2.19.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.19.3.3 ROILabelRegexA

2.19.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.19.3.3.2 Default

- `".*"`

2.19.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.19.3.4 NormalizedROILabelRegexB

2.19.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.19.3.4.2 Default

- `".*"`

2.19.3.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.19.3.5 ROILabelRegexB

2.19.3.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.19.3.5.2 Default

- `".*"`

2.19.3.5.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.19.3.6 FileName

2.19.3.6.1 Description

A filename (or full path) in which to append similarity data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.19.3.6.2 Default

- `""`

2.19.3.6.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.19.3.7 UserComment

2.19.3.7.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.19.3.7.2 Default

- `""`

2.19.3.7.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.20 ContourViaGeometry

2.20.1 Description

This operation constructs ROI contours using geometrical primitives.

2.20.2 Notes

- This routine requires an image array onto which the contours will be written.
- This routine expects images to be non-overlapping. In other words, if images overlap then the contours generated may also overlap. This is probably not what you want (but there is nothing intrinsically wrong with presenting this routine with multiple images if you intentionally want overlapping contours).
- Existing contours are ignored and unaltered.
- Small and degenerate contours produced by this routine are suppressed. If a specific number of contours must be generated, provide a slightly larger radius to compensate for the degenerate cases at the extrema.

2.20.3 Parameters

- ROILabel
- ImageSelection
- Shapes

2.20.3.1 ROILabel

2.20.3.1.1 Description

A label to attach to the ROI contours.

2.20.3.1.2 Default

- `"unspecified"`

2.20.3.1.3 Examples

- `"unspecified"`
- `"body"`
- `"air"`
- `"bone"`
- `"invalid"`
- `"above_zero"`
- `"below_5.3"`

2.20.3.2 ImageSelection

2.20.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.20.3.2.2 Default

- `"last"`

2.20.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.20.3.3 Shapes

2.20.3.3.1 Description

This parameter is used to specify the shapes to consider. There is currently a single supported shape: sphere. However, it is likely that more shapes will be accepted in the future. Spheres have two configurable parameters: centre and radius. A sphere with centre (1.0,2.0,3.0) and radius 12.3 can be specified as 'sphere(1.0, 2.0, 3.0, 12.3)'.

2.20.3.3.2 Default

- `""`

2.20.3.3.3 Examples

- `"sphere(-1.0, 2.0, 3.0,  12.3)"`

2.21 ContourViaThreshold

2.21.1 Description

This operation constructs ROI contours using images and pixel/voxel value thresholds. There are two methods of contour generation available: a simple binary method in which voxels are either fully in or fully out of the contour, and a method based on marching cubes that will provide smoother contours. The marching cubes method does **not** construct a full surface mesh; rather each individual image slice has their own mesh constructed in parallel.

2.21.2 Notes

- This routine expects images to be non-overlapping. In other words, if images overlap then the contours generated may also overlap. This is probably not what you want (but there is nothing intrinsically wrong with presenting this routine with multiple images if you intentionally want overlapping contours).
- Existing contours are ignored and unaltered.
- Contour orientation is (likely) not properly handled in this routine, so 'pinches' and holes will produce contours with inconsistent or invalid topology. If in doubt, disable merge simplifications and live with the computational penalty. The marching cubes approach will properly handle 'pinches' and contours should all be topologically valid.

2.21.3 Parameters

- ROILabel
- Lower
- Upper
- Channel
- ImageSelection
- Method
- SimplifyMergeAdjacent

2.21.3.1 ROILabel

2.21.3.1.1 Description

A label to attach to the ROI contours.

2.21.3.1.2 Default

- `"unspecified"`

2.21.3.1.3 Examples

- `"unspecified"`
- `"body"`
- `"air"`
- `"bone"`
- `"invalid"`
- `"above_zero"`
- `"below_5.3"`

2.21.3.2 Lower

2.21.3.2.1 Description

The lower bound (inclusive). Pixels with values < this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.21.3.2.2 Default

- `"-inf"`

2.21.3.2.3 Examples

- `"0.0"`
- `"-1E-99"`
- `"1.23"`
- `"0.2%"`
- `"23tile"`
- `"23.123 tile"`

2.21.3.3 Upper

2.21.3.3.1 Description

The upper bound (inclusive). Pixels with values > this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.21.3.3.2 Default

- `"inf"`

2.21.3.3.3 Examples

- `"1.0"`
- `"1E-99"`
- `"2.34"`

- `"98.12%"`
- `"94tile"`
- `"94.123 tile"`

2.21.3.4 Channel

2.21.3.4.1 Description

The image channel to use. Zero-based.

2.21.3.4.2 Default

- `"0"`

2.21.3.4.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.21.3.5 ImageSelection

2.21.3.5.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.21.3.5.2 Default

- `"last"`

2.21.3.5.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.21.3.6 Method

2.21.3.6.1 Description

There are currently two supported methods for generating contours: (1) a simple (and fast) binary inclusivity checker, that simply checks if a voxel is within the ROI by testing the value at the voxel centre, and (2) a robust (but slow) method based on marching cubes. The binary method is fast, but produces extremely jagged contours. It may also have problems with 'pinches' and topological consistency. The marching method is more robust and should reliably produce contours for even the most complicated topologies, but is considerably slower than the binary method.

2.21.3.6.2 Default

- `"binary"`

2.21.3.6.3 Examples

- `"binary"`
- `"marching"`

2.21.3.7 SimplifyMergeAdjacent

2.21.3.7.1 Description

Simplify contours by merging adjacent contours. This reduces the number of contours dramatically, but will cause issues if there are holes (two contours are generated if there is a single hole, but most DICOMautomaton code disregards orientation – so the pixels within the hole will be considered part of the ROI, possibly even doubly so depending on the algorithm). Disabling merges is always safe (and is therefore the default) but can be extremely costly for large images. Furthermore, if you know the ROI does not have holes (or if you don't care) then it is safe to enable merges.

2.21.3.7.2 Default

- `"false"`

2.21.3.7.3 Examples

- `"true"`
- `"false"`

2.22 ContourVote

2.22.1 Description

This routine pits contours against one another using various criteria. A number of 'closest' or 'best' or 'winning' contours are copied into a new contour collection with the specified ROILabel. The original ROIs are not altered, even the winning ROIs.

2.22.2 Notes

- This operation considers individual contours only at the moment. It could be extended to operate on whole ROIs (i.e., contour_collections), or to perform a separate vote within each ROI. The individual contour approach was taken for relevance in 2D image (e.g., RTIMAGE) analysis.
- This operation currently cannot perform voting on multiple criteria. Several criteria could be specified, but an awkward weighting system would also be needed.

2.22.3 Parameters

- WinnerROILabel
- ROILabelRegex
- NormalizedROILabelRegex
- Area
- Perimeter
- CentroidX
- CentroidY
- CentroidZ
- WinnerCount

2.22.3.1 WinnerROILabel

2.22.3.1.1 Description

The ROI label to attach to the winning contour(s). All other metadata remains the same.

2.22.3.1.2 Default

- `"unspecified"`

2.22.3.1.3 Examples

- `"closest"`
- `"best"`
- `"winners"`
- `"best-matches"`

2.22.3.2 ROILabelRegex

2.22.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.22.3.2.2 Default

- `".*"`

2.22.3.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.22.3.3 NormalizedROILabelRegex

2.22.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.22.3.3.2 Default

- `".*"`

2.22.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.22.3.4 Area

2.22.3.4.1 Description

If this option is provided with a valid positive number, the contour(s) with an area closest to the specified value is/are retained. Note that the DICOM coordinate space is used. (Supplying the default, NaN, will disable this option.) Note: if several criteria are specified, it is not specified in which order they are considered.

2.22.3.4.2 Default

- `"nan"`

2.22.3.4.3 Examples

- `"nan"`
- `"100.0"`
- `"1000"`
- `"10.23E8"`

2.22.3.5 Perimeter

2.22.3.5.1 Description

If this option is provided with a valid positive number, the contour(s) with a perimeter closest to the specified value is/are retained. Note that the DICOM coordinate space is used. (Supplying the default, NaN, will disable this option.) Note: if several criteria are specified, it is not specified in which order they are considered.

2.22.3.5.2 Default

- `"nan"`

2.22.3.5.3 Examples

- `"nan"`
- `"0.0"`
- `"123.456"`
- `"1E6"`

2.22.3.6 CentroidX

2.22.3.6.1 Description

If this option is provided with a valid positive number, the contour(s) with a centroid closest to the specified value is/are retained. Note that the DICOM coordinate space is used. (Supplying the default, NaN, will disable this option.) Note: if several criteria are specified, it is not specified in which order they are considered.

2.22.3.6.2 Default

- `"nan"`

2.22.3.6.3 Examples

- `"nan"`
- `"0.0"`
- `"123.456"`
- `"-1E6"`

2.22.3.7 CentroidY

2.22.3.7.1 Description

If this option is provided with a valid positive number, the contour(s) with a centroid closest to the specified value is/are retained. Note that the DICOM coordinate space is used. (Supplying the default, NaN, will disable this option.) Note: if several criteria are specified, it is not specified in which order they are considered.

2.22.3.7.2 Default

- `"nan"`

2.22.3.7.3 Examples

- `"nan"`
- `"0.0"`
- `"123.456"`
- `"-1E6"`

2.22.3.8 CentroidZ

2.22.3.8.1 Description

If this option is provided with a valid positive number, the contour(s) with a centroid closest to the specified value is/are retained. Note that the DICOM coordinate space is used. (Supplying the default, NaN, will disable this option.) Note: if several criteria are specified, it is not specified in which order they are considered.

2.22.3.8.2 Default

- `"nan"`

2.22.3.8.3 Examples

- `"nan"`
- `"0.0"`
- `"123.456"`
- `"-1E6"`

2.22.3.9 WinnerCount

2.22.3.9.1 Description

Retain this number of ‘best’ or ‘winning’ contours.

2.22.3.9.2 Default

- `"1"`

2.22.3.9.3 Examples

- `"0"`
- `"1"`
- `"3"`
- `"10000"`

2.23 ContourWholeImages

2.23.1 Description

This operation constructs contours for an ROI that encompasses the whole of all specified images. It is useful for operations that operate on ROIs whenever you want to compute something over the whole image. This routine avoids having to manually contour anything. The output is ‘ephemeral’ and is not commited to any database.

2.23.2 Notes

- This routine will attempt to avoid repeat contours. Generated contours are tested for intersection with an image before the image is processed.
- Existing contours are ignored and unaltered.
- Contours are set slightly inside the outer boundary so they can be easily visualized by overlaying on the image. All voxel centres will be within the bounds.

2.23.3 Parameters

- ROILabel
- ImageSelection

2.23.3.1 ROILabel

2.23.3.1.1 Description

A label to attach to the ROI contours.

2.23.3.1.2 Default

- `"everything"`

2.23.3.1.3 Examples

- `"everything"`
- `"whole_images"`
- `"unspecified"`

2.23.3.2 ImageSelection

2.23.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.23.3.2.2 Default

- `"last"`

2.23.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`

- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.24 ContouringAides

2.24.1 Description

This operation attempts to prepare an image for easier contouring.

2.24.2 Notes

- At the moment, only logarithmic scaling is applied.

2.24.3 Parameters

No registered options.

2.25 ConvertDoseToImage

2.25.1 Description

This operation converts all loaded images from RTDOSE modality to CT modality. Image contents will not change, but the intent to treat as an image or dose matrix will of course change.

2.25.2 Parameters

- Modality

2.25.2.1 Modality

2.25.2.1.1 Description

The modality that will replace ‘RTDOSE’.

2.25.2.1.2 Default

- `"CT"`

2.25.2.1.3 Examples

- `"CT"`
- `"MR"`
- `"UNKNOWN"`

2.26 ConvertImageToDose

2.26.1 Description

This operation converts all loaded image modalities into RTDOSE. Image contents will not change, but the intent to treat as an image or dose matrix will of course change.

2.26.2 Parameters

No registered options.

2.27 ConvertImageToMeshes

2.27.1 Description

This operation extracts surface meshes from images and pixel/voxel value thresholds. Meshes are appended to the back of the Surface_Mesh stack. There are two methods of contour generation available: a simple binary method in which voxels are either fully in or fully out of the contour, and a method based on marching cubes that will provide smoother contours. Both methods make use of marching cubes – the binary method involves pre-processing.

2.27.2 Notes

- This routine requires images to be regular (i.e., exactly abut nearest adjacent images without any overlap).

2.27.3 Parameters

- ImageSelection
- Lower
- Upper
- Channel
- Method
- MeshLabel

2.27.3.1 ImageSelection

2.27.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.27.3.1.2 Default

- `"last"`

2.27.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.27.3.2 Lower

2.27.3.2.1 Description

The lower bound (inclusive). Pixels with values < this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile].

Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile). Note that computed bounds (i.e., percentages and percentiles) consider the entire image volume.

2.27.3.2.2 Default

- `"-inf"`

2.27.3.2.3 Examples

- `"0.0"`
- `"-1E-99"`
- `"1.23"`
- `"0.2%"`
- `"23tile"`
- `"23.123 tile"`

2.27.3.3 Upper

2.27.3.3.1 Description

The upper bound (inclusive). Pixels with values > this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile). Note that computed bounds (i.e., percentages and percentiles) consider the entire image volume.

2.27.3.3.2 Default

- `"inf"`

2.27.3.3.3 Examples

- `"1.0"`
- `"1E-99"`
- `"2.34"`
- `"98.12%"`
- `"94tile"`
- `"94.123 tile"`

2.27.3.4 Channel

2.27.3.4.1 Description

The image channel to use. Zero-based.

2.27.3.4.2 Default

- `"0"`

2.27.3.4.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.27.3.5 Method

2.27.3.5.1 Description

There are currently two supported methods for generating contours: (1) a simple (and fast) binary inclusivity checker, that simply checks if a voxel is within the ROI by testing the value at the voxel centre, and (2) a robust (but slow) method based on marching cubes. The binary method is fast, but produces extremely jagged contours. It may also have problems with 'pinches' and topological consistency. The marching method is more robust and should reliably produce contours for even the most complicated topologies, but is considerably slower than the binary method.

2.27.3.5.2 Default

- `"marching"`

2.27.3.5.3 Examples

- `"binary"`
- `"marching"`

2.27.3.6 MeshLabel

2.27.3.6.1 Description

A label to attach to the surface mesh.

2.27.3.6.2 Default

- `"unspecified"`

2.27.3.6.3 Examples

- `"unspecified"`
- `"body"`
- `"air"`
- `"bone"`
- `"invalid"`
- `"above_zero"`
- `"below_5.3"`

2.28 ConvertMeshesToContours

2.28.1 Description

This operation constructs ROI contours by slicing the given meshes on a set of image planes.

2.28.2 Notes

- Surface meshes should represent polyhedra.
- This routine does **not** require images to be regular, rectilinear, or even contiguous.
- Images and meshes are unaltered. Existing contours are ignored and unaltered.
- Contour orientation is (likely) not guaranteed to be consistent in this routine.

2.28.3 Parameters

- ROILabel
- MeshSelection
- ImageSelection

2.28.3.1 ROILabel

2.28.3.1.1 Description

A label to attach to the ROI contours.

2.28.3.1.2 Default

- `"unspecified"`

2.28.3.1.3 Examples

- `"unspecified"`
- `"body"`
- `"air"`
- `"bone"`
- `"invalid"`
- `"above_zero"`
- `"below_5.3"`

2.28.3.2 MeshSelection

2.28.3.2.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.28.3.2.2 Default

- `"last"`

2.28.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`

- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.28.3.3 ImageSelection

2.28.3.3.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.28.3.3.2 Default

- `"last"`

2.28.3.3.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.29 ConvertNaNsToAir

2.29.1 Description

This operation runs the data through a per-pixel filter, converting NaN's to air in Hounsfield units (-1024).

2.29.2 Parameters

No registered options.

2.30 ConvertNaNsToZeros

2.30.1 Description

This operation runs the data through a per-pixel filter, converting NaN's to zeros.

2.30.2 Parameters

No registered options.

2.31 ConvertPixelsToPoints

2.31.1 Description

This operation extracts pixels from the selected images and converts them into a point cloud. Images are not modified.

2.31.2 Notes

- Existing point clouds are ignored and unaltered.

2.31.3 Parameters

- Label
- Lower
- Upper
- Channel
- ImageSelection

2.31.3.1 Label

2.31.3.1.1 Description

A label to attach to the point cloud.

2.31.3.1.2 Default

- `"unspecified"`

2.31.3.1.3 Examples

- `"unspecified"`
- `"POIs"`
- `"peaks"`
- `"above_zero"`
- `"below_5.3"`

2.31.3.2 Lower

2.31.3.2.1 Description

The lower bound (inclusive). Pixels with values < this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.31.3.2.2 Default

- `"-inf"`

2.31.3.2.3 Examples

- `"0.0"`
- `"-1E-99"`
- `"1.23"`
- `"0.2%"`
- `"23tile"`
- `"23.123 tile"`

2.31.3.3 Upper

2.31.3.3.1 Description

The upper bound (inclusive). Pixels with values > this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.31.3.3.2 Default

- `"inf"`

2.31.3.3.3 Examples

- `"1.0"`
- `"1E-99"`
- `"2.34"`
- `"98.12%"`
- `"94tile"`
- `"94.123 tile"`

2.31.3.4 Channel

2.31.3.4.1 Description

The image channel to use. Zero-based.

2.31.3.4.2 Default

- `"0"`

2.31.3.4.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.31.3.5 ImageSelection

2.31.3.5.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.31.3.5.2 Default

- `"last"`

2.31.3.5.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.32 ConvolveImages

2.32.1 Description

This routine convolves, correlates, or pattern-matches one rectilinear image array with another in voxel number space (i.e., the DICOM coordinate system of the convolution kernel image is entirely disregarded).

2.32.2 Notes

- Both provided image arrays must be rectilinear. In many instances they should both be regular, not just rectilinear, but rectilinearity is sufficient for constructing voxel-by-voxel adjacency relatively quickly, and some applications may require rectilinear kernels to be supported, so rectilinear inputs are permitted.
- This operation can be used to apply arbitrary convolution kernels to an image array. It can also be used to search for instances of one image array in another.
- If the magnitude of the outgoing voxels will be interpretted in absolute (i.e., thresholding based on an absolute magnitude) then the kernel should be weighted so that the sum of all kernel voxel intensities is zero. This will maintain the average voxel intensity. However, for pattern matching the kernel need not be normalized (though it may make interpretting partial matches easier.)

2.32.3 Parameters

- ImageSelection
- ReferenceImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Operation

2.32.3.1 ImageSelection

2.32.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order

specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.32.3.1.2 Default

- `"last"`

2.32.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.32.3.2 ReferenceImageSelection

2.32.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or ‘4D’ time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.32.3.2.2 Default

- `"last"`

2.32.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.32.3.3 NormalizedROILabelRegex

2.32.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.32.3.3.2 Default

- `".*"`

2.32.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.32.3.4 ROILabelRegex

2.32.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them.

All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.32.3.4.2 Default

- `".*"`

2.32.3.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.32.3.5 Channel

2.32.3.5.1 Description

The channel to operate on (zero-based). Negative values will cause all channels to be operated on.

2.32.3.5.2 Default

- `"0"`

2.32.3.5.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.32.3.6 Operation

2.32.3.6.1 Description

Controls the way the kernel is applied and the reduction is tallied. Currently, ‘convolution’, ‘correlation’, and ‘pattern-match’ are supported. For convolution, the reference image is spatially inverted along row-, column-, and image-axes. The outgoing voxel intensity is the inner (i.e., dot) product of the paired intensities of the surrounding voxel neighbourhood (i.e., the voxel at (-1,3,0) from the centre of the kernel is paired with the neighbouring voxel at (-1,3,0) from the current/outgoing voxel). For pattern-matching, the difference between the kernel and each voxel's neighbourhood voxels is compared using a 2-norm (i.e., Euclidean) cost function. With this cost function, a perfect, pixel-for-pixel match (i.e., if the kernel images appears exactly in the image being transformed) will result in the outgoing voxel having zero intensity (i.e., zero cost). For correlation, the kernel is applied as-is (just like pattern-matching), but the inner product of the paired voxel neighbourhood intensities is reported (just like convolution). In all cases the kernel is (approximately) centred.

2.32.3.6.2 Default

- `"convolution"`

2.32.3.6.3 Examples

- `"convolution"`
- `"correlation"`
- `"pattern-match"`

2.33 CopyImages

2.33.1 Description

This operation deep-copies the selected image arrays.

2.33.2 Parameters

- ImageSelection

2.33.2.1 ImageSelection

2.33.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.33.2.1.2 Default

- `"last"`

2.33.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.34 CopyMeshes

2.34.1 Description

This operation deep-copies the selected surface meshes.

2.34.2 Parameters

- MeshSelection

2.34.2.1 MeshSelection

2.34.2.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the

Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.34.2.1.2 Default

- `"last"`

2.34.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.35 CountVoxels

2.35.1 Description

This operation counts the number of voxels confined to one or more ROIs within a user-provided range.

2.35.2 Notes

- This operation is read-only.

2.35.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Inclusivity

- ContourOverlap
- Lower
- Upper
- Channel
- ResultsSummaryFileName
- UserComment

2.35.3.1 ImageSelection

2.35.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.35.3.1.2 Default

- `"last"`

2.35.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.35.3.2 NormalizedROILabelRegex

2.35.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.35.3.2.2 Default

- `".*"`

2.35.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.35.3.3 ROILabelRegex

2.35.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.35.3.3.2 Default

- `".*"`

2.35.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.35.3.4 Inclusivity

2.35.3.4.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.35.3.4.2 Default

- `"center"`

2.35.3.4.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.35.3.5 ContourOverlap

2.35.3.5.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.35.3.5.2 Default

- `"ignore"`

2.35.3.5.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.35.3.6 Lower

2.35.3.6.1 Description

The lower bound (inclusive). Pixels with values < this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.35.3.6.2 Default

- `"-inf"`

2.35.3.6.3 Examples

- `"0.0"`
- `"-1E-99"`
- `"1.23"`
- `"0.2%"`
- `"23tile"`
- `"23.123 tile"`

2.35.3.7 Upper

2.35.3.7.1 Description

The upper bound (inclusive). Pixels with values > this number are excluded from the ROI. If the number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If the number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.35.3.7.2 Default

- `"inf"`

2.35.3.7.3 Examples

- `"1.0"`
- `"1E-99"`
- `"2.34"`
- `"98.12%"`
- `"94tile"`
- `"94.123 tile"`

2.35.3.8 Channel

2.35.3.8.1 Description

The image channel to use. Zero-based.

2.35.3.8.2 Default

- `"0"`

2.35.3.8.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.35.3.9 ResultsSummaryFileName

2.35.3.9.1 Description

This file will contain a brief summary of the results. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.35.3.9.2 Default

- `""`

2.35.3.9.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.35.3.10 UserComment

2.35.3.10.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.35.3.10.2 Default

- `""`

2.35.3.10.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.36 CropImageDoseToROIs

2.36.1 Description

This operation crops image slices to the specified ROI(s), with an additional margin.

2.36.2 Parameters

- DICOMMargin
- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex

2.36.2.1 DICOMMargin

2.36.2.1.1 Description

The amount of margin (in the DICOM coordinate system) to surround the ROI(s).

2.36.2.1.2 Default

- `"0.5"`

2.36.2.1.3 Examples

- `"0.1"`
- `"2.0"`
- `"-0.5"`
- `"20.0"`

2.36.2.2 ImageSelection

2.36.2.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.36.2.2.2 Default

- `"last"`

2.36.2.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`

- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.36.2.3 NormalizedROILabelRegex

2.36.2.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.36.2.3.2 Default

- ".*"

2.36.2.3.3 Examples

- ".*"
- ".*Body.*"
- "Body"
- "Gross_Liver"
- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.36.2.4 ROILabelRegex

2.36.2.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.36.2.4.2 Default

- ".*"

2.36.2.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.37 CropImages

2.37.1 Description

This operation crops image slices in either pixel or DICOM coordinate spaces.

2.37.2 Parameters

- ImageSelection
- RowsL
- RowsH
- ColumnsL
- ColumnsH
- DICOMMargin

2.37.2.1 ImageSelection

2.37.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.37.2.1.2 Default

- `"all"`

2.37.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.37.2.2 RowsL

2.37.2.2.1 Description

The number of rows to remove, starting with the first row. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first row can be either on the top or bottom of the image.

2.37.2.2.2 Default

- `"0px"`

2.37.2.2.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.37.2.3 RowsH

2.37.2.3.1 Description

The number of rows to remove, starting with the last row. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first row can be either on the top or bottom of the image.

2.37.2.3.2 Default

- `"0px"`

2.37.2.3.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.37.2.4 ColumnsL

2.37.2.4.1 Description

The number of columns to remove, starting with the first column. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first column can be either on the top or bottom of the image.

2.37.2.4.2 Default

- `"0px"`

2.37.2.4.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.37.2.5 ColumnsH

2.37.2.5.1 Description

The number of columns to remove, starting with the last column. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first column can be either on the top or bottom of the image.

2.37.2.5.2 Default

- `"0px"`

2.37.2.5.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.37.2.6 DICOMMargin

2.37.2.6.1 Description

The amount of margin (in the DICOM coordinate system) to spare from cropping.

2.37.2.6.2 Default

- `"0.0"`

2.37.2.6.3 Examples

- `"0.1"`
- `"2.0"`
- `"-0.5"`
- `"20.0"`

2.38 CropROIDose

2.38.1 Description

This operation provides a simplified interface for overriding voxel values outside a ROI. For example, this operation can be used to modify a base plan by eliminating dose outside an OAR.

2.38.2 Notes

- This operation performs the opposite of the 'Trim' operation, which trims voxel values **inside** a ROI.
- The inclusivity of a voxel that straddles the ROI boundary can be specified in various ways. Refer to the Inclusivity parameter documentation.

2.38.3 Parameters

- Channel
- ImageSelection
- ContourOverlap
- Inclusivity
- ExteriorVal
- InteriorVal
- ExteriorOverwrite
- InteriorOverwrite
- NormalizedROILabelRegex
- ROILabelRegex
- Filename
- ParanoiaLevel

2.38.3.1 Channel

2.38.3.1.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.38.3.1.2 Default

- `"-1"`

2.38.3.1.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.38.3.2 ImageSelection

2.38.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or ‘4D’ time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.38.3.2.2 Default

- `"all"`

2.38.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.38.3.3 ContourOverlap

2.38.3.3.1 Description

Controls overlapping contours are treated. The default ‘ignore’ treats overlapping contours as a single contour, regardless of contour orientation. The option ‘honour_opposite_orientations’ makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option ‘overlapping_contours_cancel’ ignores orientation and cancels all contour overlap.

2.38.3.3.2 Default

- `"ignore"`

2.38.3.3.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.38.3.4 Inclusivity

2.38.3.4.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.38.3.4.2 Default

- `"planar_inc"`

2.38.3.4.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.38.3.5 ExteriorVal

2.38.3.5.1 Description

The value to give to voxels outside the specified ROI(s). Note that this value will be ignored if exterior overwrites are disabled.

2.38.3.5.2 Default

- `"0.0"`

2.38.3.5.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.38.3.6 InteriorVal

2.38.3.6.1 Description

The value to give to voxels within the volume of the specified ROI(s). Note that this value will be ignored if interior overwrites are disabled.

2.38.3.6.2 Default

- `"0.0"`

2.38.3.6.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.38.3.7 ExteriorOverwrite

2.38.3.7.1 Description

Whether to overwrite voxels exterior to the specified ROI(s).

2.38.3.7.2 Default

- `"true"`

2.38.3.7.3 Examples

- `"true"`
- `"false"`

2.38.3.8 InteriorOverwrite

2.38.3.8.1 Description

Whether to overwrite voxels interior to the specified ROI(s).

2.38.3.8.2 Default

- `"false"`

2.38.3.8.3 Examples

- `"true"`
- `"false"`

2.38.3.9 NormalizedROILabelRegex

2.38.3.9.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.38.3.9.2 Default

- `".*"`

2.38.3.9.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.38.3.10 ROILabelRegex

2.38.3.10.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.38.3.10.2 Default

- `".*"`

2.38.3.10.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.38.3.11 Filename

2.38.3.11.1 Description

The filename (or full path name) to which the DICOM file should be written.

2.38.3.11.2 Default

- `"/tmp/RD.dcm"`

2.38.3.11.3 Examples

- `"/tmp/RD.dcm"`
- `"./RD.dcm"`
- `"RD.dcm"`

2.38.3.12 ParanoiaLevel

2.38.3.12.1 Description

At low paranoia setting, only top-level UIDs are replaced. At medium paranoia setting, many UIDs, descriptions, and labels are replaced, but the PatientID and FrameOfReferenceUID are retained. The high paranoia setting is the same as the medium setting, but the PatientID and FrameOfReferenceUID are also replaced. (Note: this is not a full anonymization.) Use the low setting if you want to retain linkage to the originating data set. Use the medium setting if you don't. Use the high setting if your TPS goes overboard linking data sets by PatientID and/or FrameOfReferenceUID.

2.38.3.12.2 Default

- `"medium"`

2.38.3.12.3 Examples

- `"low"`
- `"medium"`
- `"high"`

2.39 DCEMRI_IAUC

2.39.1 Description

This operation will compute the Integrated Area Under the Curve (IAUC) for any images present.

2.39.2 Notes

- This operation is not optimized in any way and operates on whole images. It can be fairly slow, especially if the image volume is huge, so it is best to crop images if possible.

2.39.3 Parameters

No registered options.

2.40 DCEMRI_Nonparametric_CE

2.40.1 Description

This operation takes a single DCE-MRI scan ('measurement') and generates a "poor-mans's" contrast enhancement signal. This is accomplished by subtracting the pre-contrast injection images average ('baseline') from later images (and then possibly/optionally averaging relative to the baseline).

2.40.2 Notes

- Only a single image volume is required. It is expected to have temporal sampling beyond the contrast injection timepoint (or some default value – currently around ~30s). The resulting images retain the baseline portion, so you'll need to trim yourself if needed.
- Be aware that this method of deriving contrast enhancement is not valid! It ignores nuances due to differing T1 or T2 values due to the presence of contrast agent. It should only be used for exploratory purposes or cases where the distinction with reality is irrelevant.

2.40.3 Parameters

No registered options.

2.41 DICOMExportImagesAsDose

2.41.1 Description

This operation exports the last Image_Array to a DICOM dose file.

2.41.2 Notes

- There are various 'paranoia' levels that can be used to partially anonymize the output. In particular, most metadata and UIDs are replaced, but the files may still be recognized by a determined individual by comparing the coordinate system and pixel values. Do NOT rely on this routine to fully anonymize the data!

2.41.3 Parameters

- Filename
- ParanoiaLevel

2.41.3.1 Filename

2.41.3.1.1 Description

The filename (or full path name) to which the DICOM file should be written.

2.41.3.1.2 Default

- `"/tmp/RD.dcm"`

2.41.3.1.3 Examples

- `"/tmp/RD.dcm"`
- `"./RD.dcm"`
- `"RD.dcm"`

2.41.3.2 ParanoiaLevel

2.41.3.2.1 Description

At low paranoia setting, only top-level UIDs are replaced. At medium paranoia setting, many UIDs, descriptions, and labels are replaced, but the PatientID and FrameOfReferenceUID are retained. The high paranoia setting is the same as the medium setting, but the PatientID and FrameOfReferenceUID are also replaced. (Note: this is not a full anonymization.) Use the low setting if you want to retain linkage to the originating data set. Use the medium setting if you don't. Use the high setting if your TPS goes overboard linking data sets by PatientID and/or FrameOfReferenceUID.

2.41.3.2.2 Default

- `"medium"`

2.41.3.2.3 Examples

- `"low"`
- `"medium"`
- `"high"`

2.42 DecayDoseOverTimeHalve

2.42.1 Description

This operation transforms a dose map (assumed to be delivered some distant time in the past) to simulate 'decay' or 'evaporation' or 'forgivance' of radiation dose by simply halving the value. This model is only appropriate at long time-scales, but there is no cut-off or threshold to denote what is sufficiently 'long'. So use at your own risk. As a rule of thumb, do not use this routine if fewer than 2-3y have elapsed.

2.42.2 Notes

- This routine will combine spatially-overlapping images by summing voxel intensities. So if you have a time course it may be more sensible to aggregate images in some way (e.g., spatial averaging) prior to calling this routine.
- Since this routine is meant to be applied multiple times in succession for different ROIs (which possibly overlap), all images are imbued with a second channel that is treated as a mask. Mask channels are permanently attached so that multiple passes will not erroneously decay dose. If this will be problematic, the extra column should be trimmed immediately after calling this routine.

2.42.3 Parameters

- NormalizedROILabelRegex
- ROILabelRegex

2.42.3.1 NormalizedROILabelRegex

2.42.3.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.42.3.1.2 Default

- `".*"`

2.42.3.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.42.3.2 ROILabelRegex

2.42.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.42.3.2.2 Default

- `".*"`

2.42.3.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.43 DecayDoseOverTimeJones2014

2.43.1 Description

This operation transforms a dose map (delivered some time in the past) to account for tissue recovery (i.e., 'dose decay,' 'dose evaporation,' or 'dose forgivance') using the time-dependent model of Jones and Grant (2014; doi:10.1016/j.clon.2014.04.027). This model is specific to reirradiation of central nervous tissues. See the Jones and Grant paper or 'Nasopharyngeal Carcinoma' by Wai Tong Ng et al. (2016; doi:10.1007/174_2016_48) for more information.

2.43.2 Notes

- This routine will combine spatially-overlapping images by summing voxel intensities. So if you have a time course it may be more sensible to aggregate images in some way (e.g., spatial averaging) prior to calling this routine.
- Since this routine is meant to be applied multiple times in succession for different ROIs (which possibly overlap), all images are imbued with a second channel that is treated as a mask. Mask channels are permanently attached so that multiple passes will not erroneously decay dose. If this will be problematic, the extra column should be trimmed immediately after calling this routine.

2.43.3 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- Course1NumberOfFractions
- ToleranceTotalDose
- ToleranceNumberOfFractions
- TimeGap
- AlphaBetaRatio
- UseMoreConservativeRecovery

2.43.3.1 NormalizedROILabelRegex

2.43.3.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.43.3.1.2 Default

- `".*"`

2.43.3.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`

- "Gross_Liver"
- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.43.3.2 ROILabelRegex

2.43.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.43.3.2.2 Default

- ".*"

2.43.3.2.3 Examples

- ".*"
- ".*body.*"
- "body"
- "Gross_Liver"
- ".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"
- "left_parotid|right_parotid"

2.43.3.3 Course1NumberOfFractions

2.43.3.3.1 Description

The number of fractions delivered for the first (i.e., previous) course. If several apply, you can provide a single effective fractionation scheme's 'n'.

2.43.3.3.2 Default

- "35"

2.43.3.3.3 Examples

- "15"
- "25"
- "30.001"
- "35.3"

2.43.3.4 ToleranceTotalDose

2.43.3.4.1 Description

The dose delivered (in Gray) for a hypothetical 'lifetime dose tolerance' course. This dose corresponds to a hypothetical radiation course that nominally corresponds to the toxicity of interest. For CNS tissues, it will probably be myelopathy or necrosis at some population-level onset risk (e.g., 5% risk of myelopathy). The value provided will be converted to a BED_{a/b} so you can safely provide a 'nominal' value. Be aware that each voxel is treated independently, rather than treating OARs/ROIs as a whole. (Many dose limits reported in the literature use whole-ROI D_mean or D_max, and so may be not be directly applicable to per-voxel risk estimation!) Note that the QUANTEC 2010 reports almost all assume 2 Gy/fraction. If several fractionation schemes were used, you should provide a cumulative BED-derived dose here.

2.43.3.4.2 Default

- `"52"`

2.43.3.4.3 Examples

- `"15"`
- `"20"`
- `"25"`
- `"50"`
- `"83.2"`

2.43.3.5 ToleranceNumberOfFractions

2.43.3.5.1 Description

The number of fractions ('n') for the 'lifetime dose tolerance' toxicity you are interested in. Note that this is converted to a BED_{a/b} so you can safely provide a 'nominal' value. If several apply, you can provide a single effective fractionation scheme's 'n'.

2.43.3.5.2 Default

- `"35"`

2.43.3.5.3 Examples

- `"15"`
- `"25"`
- `"30.001"`
- `"35.3"`

2.43.3.6 TimeGap

2.43.3.6.1 Description

The number of years between radiotherapy courses. Note that this is normally estimated by (1) extracting study/series dates from the provided dose files and (2) using the current date as the second course date. Use this parameter to override the autodetected gap time. Note: if the provided value is negative, autodetection will be used. Autodetection can fail if the data has been anonymized with date-shifting.

2.43.3.6.2 Default

- `"-1"`

2.43.3.6.3 Examples

- `"0.91"`
- `"2.6"`
- `"5"`

2.43.3.7 AlphaBetaRatio

2.43.3.7.1 Description

The ratio alpha/beta (in Gray) to use when converting to a biologically-equivalent dose distribution for central nervous tissues. Jones and Grant (2014) recommend alpha/beta = 2 Gy to be conservative. It is more commonplace to use alpha/beta = 3 Gy, but this is less conservative and there is some evidence that it may be erroneous to use 3 Gy.

2.43.3.7.2 Default

- `"2"`

2.43.3.7.3 Examples

- `"2"`
- `"2.5"`
- `"3"`

2.43.3.8 UseMoreConservativeRecovery

2.43.3.8.1 Description

Jones and Grant (2014) provide two ways to estimate the function ‘r’. One is fitted to experimental data, and one is a more conservative estimate of the fitted function. This parameter controls whether or not the more conservative function is used.

2.43.3.8.2 Default

- `"true"`

2.43.3.8.3 Examples

- `"true"`
- `"false"`

2.44 DecimatePixels

2.44.1 Description

This operation spatially aggregates blocks of pixels, thereby decimating them and making the images consume far less memory. The precise size reduction and spatial aggregate can be set in the source.

2.44.2 Parameters

- OutSizeR
- OutSizeC

2.44.2.1 OutSizeR

2.44.2.1.1 Description

The number of pixels along the row unit vector to group into an outgoing pixel. Must be a multiplicative factor of the incoming image's row count. No decimation occurs if either this or 'OutSizeC' is zero or negative.

2.44.2.1.2 Default

- `"8"`

2.44.2.1.3 Examples

- `"0"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`
- `"32"`
- `"64"`
- `"128"`
- `"256"`
- `"512"`

2.44.2.2 OutSizeC

2.44.2.2.1 Description

The number of pixels along the column unit vector to group into an outgoing pixel. Must be a multiplicative factor of the incoming image's column count. No decimation occurs if either this or 'OutSizeR' is zero or negative.

2.44.2.2.2 Default

- `"8"`

2.44.2.2.3 Examples

- `"0"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`
- `"32"`
- `"64"`
- `"128"`

- "256"
- "512"

2.45 DeleteImages

2.45.1 Description

This routine deletes images from memory. It is most useful when working with positional operations in stages.

2.45.2 Parameters

- ImageSelection

2.45.2.1 ImageSelection

2.45.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.45.2.1.2 Default

- "last"

2.45.2.1.3 Examples

- "last"
- "first"
- "all"
- "none"

- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.46 DeleteMeshes

2.46.1 Description

This routine deletes surface meshes from memory. It is most useful when working with positional operations in stages.

2.46.2 Parameters

- MeshSelection

2.46.2.1 MeshSelection

2.46.2.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.46.2.1.2 Default

- `"last"`

2.46.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.47 DetectGrid3D

2.47.1 Description

This routine fits a 3D grid to a point cloud using a Procrustes analysis with point-to-model correspondence estimated via an iterative closest point approach. A RANSAC-powered loop is used to (1) randomly select a subset of the grid for coarse iterative closest point grid fitting, and then (2) use the coarse fit results as a guess for the whole point cloud in a refinement stage.

2.47.2 Notes

- Traditional Procrustes analysis requires a priori point-to-point correspondence knowledge. Because this operation fits a model (with infinite extent), point-to-point correspondence is not known and the model is effectively an infinite continuum of potential points. To overcome this problem, correspondence is estimated by projecting each point in the point cloud onto every grid line and selecting the closest projected point. The point cloud point and the project point are then treated as corresponding points. Using this phony correspondence, the Procrustes problem is solved and the grid is reoriented. This is performed iteratively. However **there is no guarantee the procedure will converge** and furthermore, even if it does converge, **there is no guarantee that the grid will be appropriately fit**. The best results will occur when the grid is already closely aligned with the point cloud (i.e., when the first guess is very close to a solution). If this cannot be guaranteed, it may be advantageous to have a nearly continuous point cloud to avoid gaps in which the iteration can get stuck in a local minimum. For this reason, RANSAC is applied to continuously reboot the fitting procedure. All but the best fit are discarded.
- A two-stage RANSAC inner-loop iterative closest point fitting procedure is used. Coarse grid fitting is first performed with a limited subset of the

whole point cloud. This is followed with a refinment stage in which the enire point cloud is fitted using an initial guess carried forward from the coarse fitting stage. This guess is expected to be reasonably close to the true grid in cases where the coarse fitting procedure was not tainted by outliers, but is only derived from a small portion of the point cloud. (Thus RANSAC is used to perform this coarse-fine iterative procedure multiple times to provide resilience to poor-quality coarse fits.) CoarseICPMaxLoops is the maximum number of iterative-closest point loop iterations performed during the coarse grid fitting stage (on a subset of the point cloud), and FineICPMaxLoops is the maximum number of iterative-closest point loop iterations performed during the refinement stage (using the whole point cloud). Note that, depending on the noise level and number of points considered (i.e., whether the RANSACDist parameter is sufficiently small to avoid spatial wrapping of corresponding points into adjacent grid cells, but sufficiently large to enclose at least one whole grid cell), the coarse phase should converge within a few iterations. However, on each loop a single point is selected as the grid's rotation centre. This means that a few extra iterations should always be used in case outliers are selected as rotation centres. Additionally, if the point cloud is dense or there are lots of outliers present, increase CoarseICPMaxLoops to ensure there is a reasonable chance of selecting legitimate rotation points. On the other hand, be aware that the coarse-fine iterative procedure is performed afresh for every RANSAC loop, and RANSAC loops are better able to ensure the point cloud is sampled ergodically. It might therefore be more productive to increase the RANSACMaxLoops parameter and reduce the number of CoarseICPMaxLoops. FineICPMaxLoops should converge quickly if the coarse fitting stage was representative of the true grid. However, as in the coarse stage a rotation centre is nominated in each loop, so it will be a good idea to keep a sufficient number of loops to ensure a legitimate and appropriate non-outlier point is nominated during this stage. Given the complicated interplay between parameters and stages, it is always best to tune using a representative sample of the point cloud you need to fit!

2.47.3 Parameters

- PointSelection
- GridSeparation
- RANSACDist
- GridSampling
- LineThickness
- RandomSeed
- RANSACMaxLoops
- CoarseICPMaxLoops
- FineICPMaxLoops
- ResultsSummaryFileName

- UserComment

2.47.3.1 PointSelection

2.47.3.1.1 Description

Select one or more point clouds. Note that point clouds can hold a variety of data with varying attributes, but each point cloud is meant to represent a single logically cohesive collection of points. Be aware that it is possible to mix logically unrelated points together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth point cloud (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last point cloud. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.47.3.1.2 Default

- `"last"`

2.47.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.47.3.2 GridSeparation

2.47.3.2.1 Description

The separation of the grid (in DICOM units; mm) being fit. This parameter describes how close adjacent grid lines are to one another. Separation is measured from one grid line centre to the nearest adjacent grid line centre.

2.47.3.2.2 Default

- `"10.0"`

2.47.3.2.3 Examples

- `"10.0"`
- `"15.5"`
- `"25.0"`
- `"1.23E4"`

2.47.3.3 RANSACDist

2.47.3.3.1 Description

Every iteration of RANSAC selects a single point from the point cloud. Only the near-vicinity of points are retained for iterative-closest-point Procrustes solving. This parameter determines the maximum radial distance from the RANSAC point within which point cloud points will be retained; all points further than this distance away will be pruned for a given round of RANSAC. This is needed because corresponding points begin to alias to incorrect cell faces when the ICP procedure begins with a poor guess. Pruning points in a spherical neighbourhood with a diameter 2-4x the GridSeparation (so a radius 1-2x GridSeparation) will help mitigate aliasing even when the initial guess is poor. However, smaller windows may increase susceptibility to noise/outliers, and RANSACDist should never be smaller than a grid voxel. If RANSACDist is not provided, a default of (1.5 * GridSeparation) is used.

2.47.3.3.2 Default

- `"nan"`

2.47.3.3.3 Examples

- `"7.0"`
- `"10.0"`
- `"2.46E4"`

2.47.3.4 GridSampling

2.47.3.4.1 Description

Specifies how the grid data has been sampled. Use value '1' if only grid cell corners (i.e., '0D' grid intersections) are sampled. Use value '2' if grid cell edges (i.e., 1D grid lines) are sampled. Use value '3' if grid cell faces (i.e., 2D planar faces) are sampled.

2.47.3.4.2 Default

- `"1"`

2.47.3.4.3 Examples

- `"1"`
- `"2"`
- `"3"`

2.47.3.5 LineThickness

2.47.3.5.1 Description

The thickness of grid lines (in DICOM units; mm). If zero, lines are treated simply as lines. If non-zero, grid lines are treated as hollow cylinders with a diameter of this thickness.

2.47.3.5.2 Default

- `"0.0"`

2.47.3.5.3 Examples

- `"1.0"`
- `"1.5"`
- `"10.0"`
- `"1.23E4"`

2.47.3.6 RandomSeed

2.47.3.6.1 Description

A whole number seed value to use for random number generation.

2.47.3.6.2 Default

- `"1317"`

2.47.3.6.3 Examples

- `"1"`
- `"2"`
- `"1113523431"`

2.47.3.7 RANSACMaxLoops

2.47.3.7.1 Description

The maximum number of iterations of RANSAC. (See operation notes for further details.)

2.47.3.7.2 Default

- `"100"`

2.47.3.7.3 Examples

- `"100"`
- `"2000"`
- `"1E4"`

2.47.3.8 CoarseICPMaxLoops

2.47.3.8.1 Description

Coarse grid fitting is performed with a limited subset of the whole point cloud. This is followed with a refinment stage in which the enire point is fitted using an initial guess from the coarse fitting stage. CoarseICPMaxLoops is the maximum number of iterative-closest point loop iterations performed during the coarse grid fitting stage. (See operation notes for further details.)

2.47.3.8.2 Default

- `"10"`

2.47.3.8.3 Examples

- `"10"`
- `"100"`
- `"1E4"`

2.47.3.9 FineICPMaxLoops

2.47.3.9.1 Description

Coarse grid fitting is performed with a limited subset of the whole point cloud. This is followed with a refinment stage in which the enire point is fitted using an initial guess from the coarse fitting stage. FineICPMaxLoops is the maximum number of iterative-closest point loop iterations performed during the refinement stage. (See operation notes for further details.)

2.47.3.9.2 Default

- `"20"`

2.47.3.9.3 Examples

- `"10"`
- `"50"`
- `"100"`

2.47.3.10 ResultsSummaryFileName

2.47.3.10.1 Description

This file will contain a brief summary of the results. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.47.3.10.2 Default

- `""`

2.47.3.10.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.47.3.11 UserComment

2.47.3.11.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.47.3.11.2 Default

- `""`

2.47.3.11.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.48 DetectShapes3D

2.48.1 Description

This operation attempts to detect shapes in image volumes.

2.48.2 Parameters

- ImageSelection

2.48.2.1 ImageSelection

2.48.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.48.2.1.2 Default

- `"all"`

2.48.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.49 DrawGeometry

2.49.1 Description

This operation draws shapes and patterns on images. Drawing is confined to one or more ROIs.

2.49.2 Parameters

- ImageSelection
- VoxelValue
- Overwrite
- Channel
- NormalizedROILabelRegex
- ROILabelRegex
- ContourOverlap
- Inclusivity
- Shapes

2.49.2.1 ImageSelection

2.49.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.49.2.1.2 Default

- `"last"`

2.49.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.49.2.2 VoxelValue

2.49.2.2.1 Description

The value to give voxels which are coincident with a point from the point cloud.

2.49.2.2.2 Default

- `"1.0"`

2.49.2.2.3 Examples

- `"-1.0"`
- `"0.0"`
- `"1.23"`
- `"nan"`
- `"inf"`

2.49.2.3 Overwrite

2.49.2.3.1 Description

Whether to overwrite voxels interior or exterior to the specified ROI(s).

2.49.2.3.2 Default

- `"interior"`

2.49.2.3.3 Examples

- `"interior"`
- `"exterior"`

2.49.2.4 Channel

2.49.2.4.1 Description

The image channel to use. Zero-based.

2.49.2.4.2 Default

- `"0"`

2.49.2.4.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.49.2.5 NormalizedROILabelRegex

2.49.2.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.49.2.5.2 Default

- `".*"`

2.49.2.5.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.49.2.6 ROILabelRegex

2.49.2.6.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.49.2.6.2 Default

- `".*"`

2.49.2.6.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.49.2.7 ContourOverlap

2.49.2.7.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.49.2.7.2 Default

- `"ignore"`

2.49.2.7.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.49.2.8 Inclusivity

2.49.2.8.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.49.2.8.2 Default

- `"center"`

2.49.2.8.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`

- "planar_inc"
- "planar_corner_exclusive"
- "planar_exc"

2.49.2.9 Shapes

2.49.2.9.1 Description

This parameter is used to specify the shapes and patterns to consider. Currently grids, wireframecubes, and solidspheres are available. Grids have four configurable parameters: two orientation unit vectors, line thickness, and line separation. A grid intersecting at the image array's centre, aligned with (1.0,0.0,0.0) and (0.0,1.0,0.0), with line thickness (i.e., diameter) 3.0 (DICOM units; mm), and separation 15.0 can be specified as 'grid(1.0,0.0,0.0, 0.0,1.0,0.0, 3.0, 15.0)'. Unit vectors will be Gram-Schmidt orthogonalized. Note that currently the grid *must* intersect the image array's centre. Cubes have the same number of configurable parameters, but only a single cube of the grid is drawn. The wireframecube is centred at the image centre, rather than intersecting it. Solid spheres have two configurable parameters: a centre vector and a radius. A solid sphere at (1.0,2.0,3.0) with radius 15.0 (all DICOM units; mm) can be specified as 'solidsphere(1.0,2.0,3.0, 15.0)'. Grid, wireframecube, and solidsphere shapes only overwrite voxels that intersect the geometry (i.e., the surface if hollow or the internal volume if solid) permitting easier composition of multiple shapes or custom backgrounds.

2.49.2.9.2 Default

- "grid(-0.0941083,0.995562,0, 0.992667,0.0938347,0.0762047, 3.0, 15.0)"

2.49.2.9.3 Examples

- "grid(1.0,0.0,0.0, 0.0,1.0,0.0, 3.0, 15.0)"
- "wireframecube(1.0,0.0,0.0, 0.0,1.0,0.0, 3.0, 15.0)"
- "solidsphere(0.0,0.0,0.0, 15.0)"

2.50 DroverDebug

2.50.1 Description

This operation reports basic information on the state of the main Drover class. It can be used to report on the state of the data, which can be useful for debugging.

2.50.2 Parameters

No registered options.

2.51 DumpAllOrderedImageMetadataToFile

2.51.1 Description

Dump exactly what order the data will be in for the following analysis.

2.51.2 Parameters

No registered options.

2.52 DumpAnEncompassedPoint

2.52.1 Description

This operation estimates the number of spatially-overlapping images. It finds an arbitrary point within an arbitrary image, and then finds all other images which encompass the point.

2.52.2 Parameters

No registered options.

2.53 DumpFilesPartitionedByTime

2.53.1 Description

This operation prints PACS filenames along with the associated time. It is more focused than the metadata dumpers above. This data can be used for many things, such as image viewers which are not DICOM-aware or deformable registration on time series data.

2.53.2 Parameters

No registered options.

2.54 DumpImageMeshes

2.54.1 Description

This operation exports images as a 3D surface mesh model (structured ASCII Wavefront OBJ) that can be manipulated in various ways (e.g., stereographic projection). Note that the mesh will be a 3D depiction of the image(s) as they naturally are – meshes will always be rectangular. A companion material library file (MTL) assigns colours to each ROI based on the voxel intensity.

2.54.2 Notes

- Each image is processed separately. Each mesh effectively produces a 2D relief map embedded into a 3D model that can be easily rendered to produce various effects (e.g., perspective, stereoscopy, extrusion, surface smoothing, etc.).

2.54.3 Parameters

- ImageSelection
- OutBase
- HistogramBins
- MagnitudeAmplification
- Normalize

2.54.3.1 ImageSelection

2.54.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.54.3.1.2 Default

- "all"

2.54.3.1.3 Examples

- "last"
- "first"
- "all"
- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.54.3.2 OutBase

2.54.3.2.1 Description

A base filename (or full path) in which to (over)write image mesh and material library files. File formats are Wavefront Object (obj) and Material Library (mtl). Every image will receive one unique and sequentially-numbered obj and mtl file using this prefix.

2.54.3.2.2 Default

- "/tmp/dicomautomaton_dumpimagemeshes_"

2.54.3.2.3 Examples

- "/tmp/image_mesh_"
- "./"
- "../model_"

2.54.3.3 HistogramBins

2.54.3.3.1 Description

The number of equal-width bins pixel intensities should be grouped into. Binning is performed in order to more easily associate material properties with pixels. If pixel intensities were continuous, each pixel would receive its own material definition. This could result in enormous MTL files and wasted disk space.

Binning solves this issue. However, if images are small or must be differentiated precisely consider using a large number of bins. Otherwise 150-1000 bins should suffice for display purposes.

2.54.3.3.2 Default

- `"255"`

2.54.3.3.3 Examples

- `"10"`
- `"50"`
- `"100"`
- `"200"`
- `"500"`

2.54.3.4 MagnitudeAmplification

2.54.3.4.1 Description

Pixel magnitudes (i.e., intensities) are scaled according to the image thickness, but a small gap is left between meshes so that abutting images do not quite intersect (this can cause non-manifold scenarios). However, if stackability is not a concern then pixel magnitudes can be magnified to exaggerate the relief effect. A value of 1.0 provides no magnification. A value of 2.0 provides 2x magnification, but note that the base of each pixel is slightly offset from the top to avoid top-bottom face intersections, even when magnification is 0.0.

2.54.3.4.2 Default

- `"1.0"`

2.54.3.4.3 Examples

- `"0.75"`
- `"1.0"`
- `"2.0"`
- `"5.0"`
- `"75.6"`

2.54.3.5 Normalize

2.54.3.5.1 Description

This parameter controls whether the model will be 'normalized,' which effectively makes the outgoing model more consistent for all images. Currently this means centring the model at (0,0,0), mapping the row and column directions to (1,0,0) and (0,1,0) respectively, and scaling the image (respecting the aspect ratio) to fit within a bounding square of size 100x100 (DICOM units; mm). If normalization is *not* used, the image mesh will inherit the spatial characteristics of the image it is derived from.

2.54.3.5.2 Default

- `"false"`

2.54.3.5.3 Examples

- `"true"`
- `"false"`

2.55 DumpImageMetadataOccurrencesToFile

2.55.1 Description

Dump all the metadata elements, but group like-items together and also print the occurence number.

2.55.2 Parameters

- ImageSelection
- FileName
- UserComment

2.55.2.1 ImageSelection

2.55.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array.

Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.55.2.1.2 Default

- `"last"`

2.55.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.55.2.2 FileName

2.55.2.2.1 Description

A filename (or full path) in which to append metadata reported by this routine. The format is tab-separated values (TSV). Leave empty to dump to generate a unique temporary file.

2.55.2.2.2 Default

- `""`

2.55.2.2.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.tsv"`
- `"derivative_data.tsv"`

2.55.2.3 UserComment

2.55.2.3.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be empty in the output.

2.55.2.3.2 Default

- `""`

2.55.2.3.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.56 DumpPerROIParams_KineticModel_1C2I_5P

2.56.1 Description

Given a perfusion model, this routine computes parameter estimates for ROIs.

2.56.2 Parameters

- ROILabelRegex
- Filename
- Separator

2.56.2.1 ROILabelRegex

2.56.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.56.2.1.2 Default

- `".*"`

2.56.2.1.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.56.2.2 Filename

2.56.2.2.1 Description

A file into which the results should be dumped. If the filename is empty, the results are dumped to the console only.

2.56.2.2.2 Default

- `""`

2.56.2.2.3 Examples

- `"/tmp/results.txt"`
- `"/dev/null"`
- `"~/output.txt"`

2.56.2.3 Separator

2.56.2.3.1 Description

The token(s) to place between adjacent columns of output. Note: because whitespace is trimmed from user parameters, whitespace separators other than the default are shortened to an empty string. So non-default whitespace are not currently supported.

2.56.2.3.2 Default

- `" "`

2.56.2.3.3 Examples

- `","`
- `";"`
- `"_a_long_separator_"`

2.57 DumpPixelValuesOverTimeForAnEncompassedPoint

2.57.1 Description

Output the pixel values over time for a generic point. Currently the point is arbitrarily taken to tbe the centre of the first image. This is useful for quickly and programmatically inspecting trends, but the SFML_Viewer operation is better for interactive exploration.

2.57.2 Parameters

No registered options.

2.58 DumpPlanSummary

2.58.1 Description

This operation dumps a summary of a radiotherapy plan. This operation can be used to gain insight into a plan from a high-level overview.

2.58.2 Parameters

- SummaryFileName
- UserComment

2.58.2.1 SummaryFileName

2.58.2.1.1 Description

A filename (or full path) in which to append summary data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.58.2.1.2 Default

- `""`

2.58.2.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.58.2.2 UserComment

2.58.2.2.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.58.2.2.2 Default

- `""`

2.58.2.2.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.59 DumpROIContours

2.59.1 Description

This operation exports contours in a standard surface mesh format (structured ASCII Wavefront OBJ) in planar polygon format. A companion material library file (MTL) assigns colours to each ROI to help differentiate them.

2.59.2 Notes

- Contours that are grouped together into a contour_collection are treated as a logical within the output. For example, all contours in a collection will share a common material property (e.g., colour). If more fine-grained grouping is required, this routine can be called once for each group which will result in a logical grouping of one ROI per file.

2.59.3 Parameters

- DumpFileName
- MTLFileName
- NormalizedROILabelRegex
- ROILabelRegex

2.59.3.1 DumpFileName

2.59.3.1.1 Description

A filename (or full path) in which to (over)write with contour data. File format is Wavefront obj. Leave empty to dump to generate a unique temporary file.

2.59.3.1.2 Default

- `""`

2.59.3.1.3 Examples

- `""`
- `"/tmp/somefile.obj"`
- `"localfile.obj"`
- `"derivative_data.obj"`

2.59.3.2 MTLFileName

2.59.3.2.1 Description

A filename (or full path) in which to (over)write a Wavefront material library file. This file is used to colour the contours to help differentiate them. Leave empty to dump to generate a unique temporary file.

2.59.3.2.2 Default

- `""`

2.59.3.2.3 Examples

- `""`
- `"/tmp/materials.mtl"`
- `"localfile.mtl"`
- `"somefile.mtl"`

2.59.3.3 NormalizedROILabelRegex

2.59.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.59.3.3.2 Default

- `".*"`

2.59.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.59.3.4 ROILabelRegex

2.59.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.59.3.4.2 Default

- `".*"`

2.59.3.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`

- "left_parotid|right_parotid"

2.60 DumpROIData

2.60.1 Description

This operation dumps ROI contour information for debugging and quick inspection purposes.

2.60.2 Parameters

No registered options.

2.61 DumpROIDoseInfo

2.61.1 Description

This operation computes mean voxel doses with the given ROIs.

2.61.2 Parameters

- ROILabelRegex

2.61.2.1 ROILabelRegex

2.61.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses grep syntax.

2.61.2.1.2 Default

- ".*"

2.61.2.1.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*parotid.*|.*sub.*mand.*"`
- `"left_parotid|right_parotid|eyes"`

2.62 DumpROISNR

2.62.1 Description

This operation computes the Signal-to-Noise ratio (SNR) for each ROI. The specific 'SNR' computed is SNR = (mean pixel) / (pixel std dev) which is the inverse of the coefficient of variation.

2.62.2 Notes

- This routine will combine spatially-overlapping images by summing voxel intensities. So if you have a time course it may be more sensible to aggregate images in some way (e.g., spatial averaging) prior to calling this routine.

2.62.3 Parameters

- SNRFileName
- NormalizedROILabelRegex
- ROILabelRegex

2.62.3.1 SNRFileName

2.62.3.1.1 Description

A filename (or full path) in which to append SNR data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.62.3.1.2 Default

- `""`

2.62.3.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.62.3.2 NormalizedROILabelRegex

2.62.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.62.3.2.2 Default

- `".*"`

2.62.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.62.3.3 ROILabelRegex

2.62.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.62.3.3.2 Default

- `".*"`

2.62.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.63 DumpROISurfaceMeshes

2.63.1 Description

This operation generates surface meshes from contour volumes. Output is written to file(s) for viewing with an external viewer (e.g., meshlab).

2.63.2 Notes

- This routine is currently limited. Many parameters can only be modified via recompilation. This will be addressed in a future version.

2.63.3 Parameters

- OutBase
- NormalizedROILabelRegex
- ROILabelRegex
- GridRows
- GridColumns
- ContourOverlap
- Inclusivity

2.63.3.1 OutBase

2.63.3.1.1 Description

The prefix of the filename that surface mesh files will be saved as. If no name is given, unique names will be chosen automatically.

2.63.3.1.2 Default

- `""`

2.63.3.1.3 Examples

- `"/tmp/dicomautomaton_dumproisurfacemesh"`
- `"../somedir/output"`
- `"/path/to/some/mesh"`

2.63.3.2 NormalizedROILabelRegex

2.63.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.63.3.2.2 Default

- `".*"`

2.63.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.63.3.3 ROILabelRegex

2.63.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses grep syntax.

2.63.3.3.2 Default

- `".*"`

2.63.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*parotid.*|.*sub.*mand.*"`
- `"left_parotid|right_parotid|eyes"`

2.63.3.4 GridRows

2.63.3.4.1 Description

Controls the spatial resolution of the grid used to approximate the ROI(s). Specifically, the number of rows. Note that the number of slices is fixed by the contour separation. A larger number will result in a more accurate mesh, but will also result longer runtimes and higher mesh complexity. Setting this parameter too high will result in excessive runtime and memory usage, so consider post-processing (i.e., subdivision) if a smooth mesh is needed.

2.63.3.4.2 Default

- `"256"`

2.63.3.4.3 Examples

- `"64"`
- `"128"`
- `"256"`
- `"512"`
- `"1024"`

2.63.3.5 GridColumns

2.63.3.5.1 Description

Controls the spatial resolution of the grid used to approximate the ROI(s). (Refer to GridRows for more information.)

2.63.3.5.2 Default

- `"256"`

2.63.3.5.3 Examples

- "64"
- "128"
- "256"
- "512"
- "1024"

2.63.3.6 ContourOverlap

2.63.3.6.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.63.3.6.2 Default

- "ignore"

2.63.3.6.3 Examples

- "ignore"
- "honour_opposite_orientations"
- "overlapping_contours_cancel"
- "honour_opps"
- "overlap_cancel"

2.63.3.7 Inclusivity

2.63.3.7.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.63.3.7.2 Default

- "center"

2.63.3.7.3 Examples

- "center"
- "centre"
- "planar_corner_inclusive"
- "planar_inc"
- "planar_corner_exclusive"
- "planar_exc"

2.64 DumpTPlanMetadataOccurrencesToFile

2.64.1 Description

Dump all the metadata elements, but group like-items together and also print the occurence number.

2.64.2 Parameters

- TPlanSelection
- FileName
- UserComment

2.64.2.1 TPlanSelection

2.64.2.1.1 Description

Select one or more treatment plans. Note that a single treatment plan may be composed of multiple beams; if delivered sequentially, they should collectively represent a single logically cohesive plan. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth treatment plan (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last treatment plan. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.64.2.1.2 Default

- "last"

2.64.2.1.3 Examples

- "last"
- "first"
- "all"
- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.64.2.2 FileName

2.64.2.2.1 Description

A filename (or full path) in which to append metadata reported by this routine. The format is tab-separated values (TSV). Leave empty to dump to generate a unique temporary file.

2.64.2.2.2 Default

- ""

2.64.2.2.3 Examples

- ""
- "/tmp/somefile"
- "localfile.tsv"
- "derivative_data.tsv"

2.64.2.3 UserComment

2.64.2.3.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be empty in the output.

2.64.2.3.2 Default

- `""`

2.64.2.3.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.65 DumpVoxelDoseInfo

2.65.1 Description

This operation locates the minimum and maximum dose voxel values. It is useful for estimating prescription doses.

2.65.2 Notes

- This implementation makes use of a primitive way of estimating dose. Please verify it works (or re-write using the new methods) before using for anything important.

2.65.3 Parameters

No registered options.

2.66 EQD2Convert

2.66.1 Description

This operation performs a BED-based conversion to a dose-equivalent that would have 2Gy fractions.

2.66.2 Notes

- This operation treats all tissue as either tumourous or not, and allows specification of a single alpha/beta for each type (i.e., one for tumourous tissues, one for normal tissues). Owing to this limitation, use of this operation is generally limited to single-OAR or PTV-only EQD2 conversions.

- This operation requires NumberOfFractions and cannot use DosePerFraction. The reasoning is that the DosePerFraction would need to be specified for each individual voxel; the prescription DosePerFraction is NOT the same as voxels outside the PTV.

2.66.3 Parameters

- ImageSelection
- AlphaBetaRatioNormal
- AlphaBetaRatioTumour
- NumberOfFractions
- PrescriptionDose
- NormalizedROILabelRegex
- ROILabelRegex

2.66.3.1 ImageSelection

2.66.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.66.3.1.2 Default

- `"last"`

2.66.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`

- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.66.3.2 AlphaBetaRatioNormal

2.66.3.2.1 Description

The value to use for alpha/beta in normal (non-cancerous) tissues. Generally a value of 3.0 Gy is used. Tissues that are sensitive to fractionation may warrant smaller ratios, such as 1.5-3 Gy for cervical central nervous tissues and 2.3-4.9 for lumbar central nervous tissues (consult table 8.1, page 107 in: Joiner et al., ‘Fractionation: the linear-quadratic approach’, 4th Ed., 2009, in the book ‘Basic Clinical Radiobiology’, ISBN: 0340929669). Note that the selected ROIs denote which tissues are diseased. The remaining tissues are considered to be normal.

2.66.3.2.2 Default

- `"3.0"`

2.66.3.2.3 Examples

- `"2.0"`
- `"3.0"`

2.66.3.3 AlphaBetaRatioTumour

2.66.3.3.1 Description

The value to use for alpha/beta in diseased (tumourous) tissues. Generally a value of 10.0 is used. Note that the selected ROIs denote which tissues are diseased. The remaining tissues are considered to be normal.

2.66.3.3.2 Default

- `"10.0"`

2.66.3.3.3 Examples

- `"10.0"`

2.66.3.4 NumberOfFractions

2.66.3.4.1 Description

The number of fractions in which a plan was (or will be) delivered. Decimal fractions are supported to accommodate previous BED conversions.

2.66.3.4.2 Default

- `"35"`

2.66.3.4.3 Examples

- `"10"`
- `"20.5"`
- `"35"`
- `"40.123"`

2.66.3.5 PrescriptionDose

2.66.3.5.1 Description

The prescription dose that was (or will be) delivered to the PTV. Note that this is a theoretical dose since the PTV or CTV will only nominally receive this dose. Also note that the specified dose need not exist somewhere in the image. It can be purely theoretical to accommodate previous BED conversions.

2.66.3.5.2 Default

- `"70"`

2.66.3.5.3 Examples

- `"15"`
- `"22.5"`
- `"45.0"`
- `"66"`
- `"70.001"`

2.66.3.6 NormalizedROILabelRegex

2.66.3.6.1 Description

A regex matching ROI labels/names to consider as bounding tumourous tissues. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.66.3.6.2 Default

- `".*"`

2.66.3.6.3 Examples

- `".*"`
- `".*GTV.*"`
- `"PTV66"`
- `".*PTV.*|.*GTV.**"`

2.66.3.7 ROILabelRegex

2.66.3.7.1 Description

A regex matching ROI labels/names to consider as bounding tumourous tissues. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.66.3.7.2 Default

- `".*"`

2.66.3.7.3 Examples

- `".*"`
- `".*GTV.*"`
- `"PTV66"`
- `".*PTV.*|.*GTV.**"`

2.67 EvaluateDoseVolumeHistograms

2.67.1 Description

This operation evaluates dose-volume histograms for the selected ROI(s).

2.67.2 Notes

- This routine generates cumulative DVHs with absolute dose on the x-axis and both absolute and fractional volume on the y-axis. Dose is reported in DICOM units (nominally Gy), absolute volume is reported in volumetric DICOM units (mm^3), and relative volume is reported as a fraction of the given ROI's total volume.
- This routine will correctly handle logically-related contours that are scattered amongst many contour collections, re-partitioning them based on ROIName. While this is often the desired behaviour, beware that any user-specified partitions will be overridden.
- This routine will correctly handle voxels of different volumes. It will not correctly handle overlapping voxels (i.e., each overlapping voxel will be counted without regard for overlap). If necessary, resample image arrays to be rectilinear.
- This routine will combine spatially-overlapping images by summing voxel intensities. It will not combine separate image_arrays. If needed, you'll have to perform a meld on them beforehand.

2.67.3 Parameters

- ImageSelection
- Channel
- NormalizedROILabelRegex
- ROILabelRegex
- ContourOverlap
- Inclusivity
- dDose
- UserComment
- OutFileName

2.67.3.1 ImageSelection

2.67.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.67.3.1.2 Default

- `"last"`

2.67.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.67.3.2 Channel

2.67.3.2.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.67.3.2.2 Default

- `"-1"`

2.67.3.2.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.67.3.3 NormalizedROILabelRegex

2.67.3.3.1 Description

A regex matching the ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.67.3.3.2 Default

- `".*"`

2.67.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.67.3.4 ROILabelRegex

2.67.3.4.1 Description

A regex matching the ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.67.3.4.2 Default

- `".*"`

2.67.3.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.67.3.5 ContourOverlap

2.67.3.5.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.67.3.5.2 Default

- `"ignore"`

2.67.3.5.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.67.3.6 Inclusivity

2.67.3.6.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.67.3.6.2 Default

- `"center"`

2.67.3.6.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.67.3.7 dDose

2.67.3.7.1 Description

The (fixed) bin width, in units of dose (DICOM units; nominally Gy).

2.67.3.7.2 Default

- `"1.0"`

2.67.3.7.3 Examples

- `"0.1"`
- `"0.5"`
- `"2.0"`
- `"5.0"`
- `"10"`
- `"50"`

2.67.3.8 UserComment

2.67.3.8.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.67.3.8.2 Default

- `""`

2.67.3.8.3 Examples

- `"Using XYZ"`
- `"Patient treatment plan C"`

2.67.3.9 OutFileName

2.67.3.9.1 Description

A filename (or full path) in which to append the histogram data generated by this routine. The format is a three-column data file suitable for plotting consisting of dose (absolute, in DICOM units of dose; nominally Gy), cumulative volume (absolute, in DICOM units of volume; mm^3), and cumulative volume (relative to the ROI's total volume, [0,1]). Existing files will be appended to; a short header will separate entries. Each distinct ROI name will have a distinct DVH entry, which will need to be delineated. (Alternatively, select a single ROI and write to a unique file.) Leave this parameter empty to generate a unique temporary file.

2.67.3.9.2 Default

- `""`

2.67.3.9.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.dat"`
- `"derivative_data.dat"`

2.68 EvaluateDoseVolumeStats

2.68.1 Description

This operation evaluates a variety of Dose-Volume statistics. It is geared toward PTV ROIs. Currently the following are implemented: (1) Dose Homogeneity Index: H = (D_{2%} - D_{98%})/D_{median} | over one or more PTVs, where D_{2%} is the maximum dose that covers 2% of the volume of the PTV, and D_{98%} is the minimum dose that covers 98% of the volume of the PTV. (2) Conformity Number: C = V_{T,pres}^{2} / (V_{T} * V_{pres}) where V_{T,pres} is the PTV volume receiving at least 95% of the PTV prescription dose, V_{T} is the volume of the PTV, and V_{pres} is volume of all (tissue) voxels receiving at least 95% of the PTV prescription dose.

2.68.2 Notes

- This routine will combine spatially-overlapping images by summing voxel intensities. It will not combine separate image_arrays though. If needed, you'll have to perform a meld on them beforehand.

2.68.3 Parameters

- OutFileName
- PTVPrescriptionDose
- PTVNormalizedROILabelRegex
- PTVROILabelRegex
- BodyNormalizedROILabelRegex
- BodyROILabelRegex
- UserComment

2.68.3.1 OutFileName

2.68.3.1.1 Description

A filename (or full path) in which to append dose statistic data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.68.3.1.2 Default

- `""`

2.68.3.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.68.3.2 PTVPrescriptionDose

2.68.3.2.1 Description

The dose prescribed to the PTV of interest (in Gy).

2.68.3.2.2 Default

- `"70"`

2.68.3.2.3 Examples

- `"50"`
- `"66"`
- `"70"`
- `"82.5"`

2.68.3.3 PTVNormalizedROILabelRegex

2.68.3.3.1 Description

A regex matching PTV ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.68.3.3.2 Default

- `".*"`

2.68.3.3.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.68.3.4 PTVROILabelRegex

2.68.3.4.1 Description

A regex matching PTV ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.68.3.4.2 Default

- `".*PTV.*"`

2.68.3.4.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.68.3.5 BodyNormalizedROILabelRegex

2.68.3.5.1 Description

A regex matching body ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.68.3.5.2 Default

- `".*"`

2.68.3.5.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.68.3.6 BodyROILabelRegex

2.68.3.6.1 Description

A regex matching PTV ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them.

All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.68.3.6.2 Default

- `".*body.*"`

2.68.3.6.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.68.3.7 UserComment

2.68.3.7.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.68.3.7.2 Default

- `""`

2.68.3.7.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.69 EvaluateNTCPModels

2.69.1 Description

This operation evaluates a variety of NTCP models for each provided ROI. The selected ROI should be OARs. Currently the following are implemented: (1) The LKB model. (2) The ‘Fenwick’ model for solid tumours (in the lung; for a whole-lung OAR).

2.69.2 Notes

- Generally these models require dose in 2Gy/fractions equivalents (‘EQD2’). You must pre-convert the data if the RT plan is not already 2Gy/fraction. There is no easy way to ensure this conversion has taken place or was unnecessary.
- This routine will combine spatially-overlapping images by summing voxel intensities. So if you have a time course it may be more sensible to aggregate images in some way (e.g., spatial averaging) prior to calling this routine.
- The LKB and mEUD both have their own gEUD ‘alpha’ parameter, but they are not necessarily shared. Huang et al. 2015 (doi:10.1038/srep18010) used alpha=1 for the LKB model and alpha=5 for the mEUD model.

2.69.3 Parameters

- NTCPFileName
- NormalizedROILabelRegex
- ROILabelRegex
- LKB_TD50
- LKB_M
- LKB_Alpha
- UserComment

2.69.3.1 NTCPFileName

2.69.3.1.1 Description

A filename (or full path) in which to append NTCP data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.69.3.1.2 Default

- `""`

2.69.3.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.69.3.2 NormalizedROILabelRegex

2.69.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.69.3.2.2 Default

- `".*"`

2.69.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.69.3.3 ROILabelRegex

2.69.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.69.3.3.2 Default

- `".*"`

2.69.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`

- "left_parotid|right_parotid"

2.69.3.4 LKB_TD50

2.69.3.4.1 Description

The dose (in Gray) needed to deliver to the selected OAR that will induce the effect in 50% of cases.

2.69.3.4.2 Default

- "26.8"

2.69.3.4.3 Examples

- "26.8"

2.69.3.5 LKB_M

2.69.3.5.1 Description

No description given…

2.69.3.5.2 Default

- "0.45"

2.69.3.5.3 Examples

- "0.45"

2.69.3.6 LKB_Alpha

2.69.3.6.1 Description

The weighting factor α that controls the relative weighting of volume and dose in the generalized Equivalent Uniform Dose (gEUD) model. When $\alpha = 1$, the gEUD is equivalent to the mean; when $\alpha = 0$, the gEUD is equivalent to the geometric mean. Wu et al. (doi:10.1016/S0360-3016(01)02585-8) claim that for normal tissues, α can be related to the Lyman-Kutcher-Burman (LKB) model volume parameter 'n' via $\alpha = 1/n$. Sovik et al. (doi:10.1016/j.ejmp.2007.09.001) found that gEUD is not strongly impacted by errors in α. Niemierko et al. ('A generalized concept of equivalent uniform dose. Med Phys 26:1100, 1999) generated maximum likelihood estimates for 'several tumors and normal structures' which ranged from -13.1 for local control of chordoma tumors to +17.7 for perforation of esophagus. Gay et al. (doi:10.1016/j.ejmp.2007.07.001) table 2 lists estimates based on the work of Emami (doi:10.1016/0360-3016(91)90171-Y) for normal tissues ranging from 1-31. Brenner et al. (doi:10.1016/0360-3016(93)90189-3) recommend -7.2 for breast cancer, -10 for melanoma, and -13 for squamous cell carcinomas. A 2017 presentation by Ontida Apinorasethkul claims the tumour range spans [-40:-1] and the organs at risk range spans [1:40]. AAPM TG report 166 also provides a listing of recommended values, suggesting -10 for PTV and GTV, +1 for parotid, 20 for spinal cord, and 8-16 for rectum, bladder, brainstem, chiasm, eye, and optic nerve. Burman (1991) and QUANTEC (2010) also provide estimates.

2.69.3.6.2 Default

- `"1.0"`

2.69.3.6.3 Examples

- `"1"`
- `"3"`
- `"4"`
- `"20"`
- `"31"`

2.69.3.7 UserComment

2.69.3.7.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.69.3.7.2 Default

- `""`

2.69.3.7.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.70 EvaluateTCPModels

2.70.1 Description

This operation evaluates a variety of TCP models for each provided ROI. The selected ROI should be the GTV (according to the Fenwick model). Currently the following are implemented: (1) The 'Martel' model. (2) Equivalent Uniform Dose (EUD) TCP. (3) The 'Fenwick' model for solid tumours.

2.70.2 Notes

- Generally these models require dose in 2Gy/fractions equivalents ('EQD2'). You must pre-convert the data if the RT plan is not already 2Gy/fraction. There is no easy way to ensure this conversion has taken place or was unnecessary.
- This routine will combine spatially-overlapping images by summing voxel intensities. So if you have a time course it may be more sensible to aggregate images in some way (e.g., spatial averaging) prior to calling this routine.
- The Fenwick and Martel models share the value of D_{50}. There may be a slight difference in some cases. Huang et al. 2015 (doi:10.1038/srep18010) used both models and used 84.5 Gy for the Martel model while using 84.6 Gy for the Fenwick model. (The paper also reported using a Fenwick 'm' of 0.329 whereas the original report by Fenwick reported 0.392, so I don't think this should be taken as strong evidence of the equality of D_{50}. However, the difference seems relatively insignificant.)

2.70.3 Parameters

- TCPFileName
- NormalizedROILabelRegex
- ROILabelRegex
- Gamma50
- Dose50
- EUD_Gamma50
- EUD_TCD50
- EUD_Alpha
- Fenwick_C
- Fenwick_M
- Fenwick_Vref
- UserComment

2.70.3.1 TCPFileName

2.70.3.1.1 Description

A filename (or full path) in which to append TCP data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.70.3.1.2 Default

- `""`

2.70.3.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.70.3.2 NormalizedROILabelRegex

2.70.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.70.3.2.2 Default

- `".*"`

2.70.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.70.3.3 ROILabelRegex

2.70.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.70.3.3.2 Default

- `".*"`

2.70.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.70.3.4 Gamma50

2.70.3.4.1 Description

The unitless 'normalized dose-response gradient' or normalized slope of the logistic dose-response model at the half-maximum point (e.g., D_50). Informally, this parameter controls the steepness of the dose-response curve. (For more specific information, consult a standard reference such as 'Basic Clinical Radiobiology' 4th Edition by Joiner et al., sections 5.3-5.5.) This parameter is empirically fit and not universal. Late endpoints for normal tissues have gamma_50 around 2-6 whereas gamma_50 nominally varies around 1.5-2.5 for local control of squamous cell carcinomas of the head and neck.

2.70.3.4.2 Default

- `"2.3"`

2.70.3.4.3 Examples

- `"1.5"`
- `"2"`
- `"2.5"`
- `"6"`

2.70.3.5 Dose50

2.70.3.5.1 Description

The dose (in Gray) needed to achieve 50% probability of local tumour control according to an empirical logistic dose-response model (e.g., D_50). Informally, this parameter 'shifts' the model along the dose axis. (For more specific information, consult a standard reference such as 'Basic Clinical Radiobiology' 4th Edition by Joiner et al., sections 5.1-5.3.) This parameter is empirically fit and not universal. In 'Quantifying the position and steepness of radiation dose-response curves' by Bentzen and Tucker in 1994, D_50 of around 60-65 Gy are reported for local control of head and neck cancers (pyriform sinus carcinoma and neck nodes with max diameter <= 3cm). Martel et al. report 84.5 Gy in lung.

2.70.3.5.2 Default

- `"65"`

2.70.3.5.3 Examples

- `"37.9"`
- `"52"`
- `"60"`
- `"65"`
- `"84.5"`

2.70.3.6 EUD_Gamma50

2.70.3.6.1 Description

The unitless 'normalized dose-response gradient' or normalized slope of the gEUD TCP model. It is defined only for the generalized Equivalent Uniform Dose (gEUD) model. This is sometimes referred to as the change in TCP for a unit change in dose straddled at the TCD_50 dose. It is a counterpart to the Martel model's 'Gamma_50' parameter, but is not quite the same. Okunieff et al. (doi:10.1016/0360-3016(94)00475-Z) computed Gamma50 for tumours in human subjects across multiple institutions; they found a median of 0.8 for gross disease and a median of 1.5 for microscopic disease. The inter-quartile range was [0.7:1.8] and [0.7:2.2] respectively. (Refer to table 3 for site-specific values.) Additionally, Gay et al. (doi:10.1016/j.ejmp.2007.07.001) claim that a value of 4.0 for late effects a value of 2.0 for tumors in 'are reasonable initial estimates in [our] experience.' Their table 2 lists (NTCP) estimates based on the work of Emami (doi:10.1016/0360-3016(91)90171-Y).

2.70.3.6.2 Default

- `"0.8"`

2.70.3.6.3 Examples

- `"0.8"`
- `"1.5"`

2.70.3.7 EUD_TCD50

2.70.3.7.1 Description

The uniform dose (in Gray) needed to deliver to the tumour to achieve 50% probability of local control. It is defined only for the generalized Equivalent Uniform Dose (gEUD) model. It is a counterpart to the Martel model's 'Dose_50' parameter, but is not quite the same (n.b., TCD_50 is a uniform dose whereas D_50 is more like a per voxel TCP-weighted mean.) Okunieff et al. (doi:10.1016/0360-3016(94)00475-Z) computed TCD50 for tumours in human subjects across multiple institutions; they found a median of 51.9 Gy for gross disease and a median of 37.9 Gy for microscopic disease. The inter-quartile range was [38.4:62.8] and [27.0:49.1] respectively. (Refer to table 3 for site-specific values.) Gay et al. (doi:10.1016/j.ejmp.2007.07.001) table 2 lists (NTCP) estimates based on the work of Emami (doi:10.1016/0360-3016(91)90171-Y) ranging from 18-68 Gy.

2.70.3.7.2 Default

- `"51.9"`

2.70.3.7.3 Examples

- `"51.9"`
- `"37.9"`

2.70.3.8 EUD_Alpha

2.70.3.8.1 Description

The weighting factor α that controls the relative weighting of volume and dose in the generalized Equivalent Uniform Dose (gEUD) model. When $\alpha = 1$, the gEUD is equivalent to the mean; when $\alpha = 0$, the gEUD is equivalent to the geometric mean. Wu et al. (doi:10.1016/S0360-3016(01)02585-8) claim that for normal tissues, α can be related to the Lyman-Kutcher-Burman (LKB) model volume parameter 'n' via $\alpha = 1/n$. Sovik et al. (doi:10.1016/j.ejmp.2007.09.001) found that gEUD is not strongly impacted by error in α. Niemierko et al. ('A generalized concept of equivalent uniform dose. Med Phys 26:1100, 1999) generated maximum likelihood estimates for 'several tumors and normal structures' which ranged from -13.1 for local control of chordoma tumors to +17.7 for perforation of esophagus. Gay et al. (doi:10.1016/j.ejmp.2007.07.001) table 2 lists estimates based on the work of Emami (doi:10.1016/0360-3016(91)90171-Y) for normal tissues ranging from 1-31. Brenner et al. (doi:10.1016/0360-3016(93)90189-3) recommend -7.2 for breast cancer, -10 for melanoma, and -13 for squamous cell carcinomas. A 2017 presentation by Ontida Apinorasethkul claims the tumour range spans [-40:-1] and the organs at risk range spans [1:40]. AAPM TG report 166 also provides a listing of recommended values, suggesting -10 for PTV and GTV, +1 for parotid, 20 for spinal cord, and 8-16 for rectum, bladder, brainstem, chiasm, eye, and optic nerve. Burman (1991) and QUANTEC (2010) also provide estimates.

2.70.3.8.2 Default

- `"-13.0"`

2.70.3.8.3 Examples

- `"-40"`
- `"-13.0"`
- `"-10"`
- `"-7.2"`
- `"0.3"`
- `"1"`
- `"3"`
- `"4"`
- `"20"`
- `"40"`

2.70.3.9 Fenwick_C

2.70.3.9.1 Description

This parameter describes the degree that superlinear doses are required to control large tumours. In other words, as tumour volume grows, a disproportionate amount of additional dose is required to maintain the same level of control. The Fenwick model is semi-empirical, so this number must be fitted or used from values reported in the literature. Fenwick et al. 2008 (doi:10.1016/j.clon.2008.12.011) provide values: 9.58 for local progression free survival at 30 months for NSCLC tumours and 5.00 for head-and-neck tumours.

2.70.3.9.2 Default

- "9.58"

2.70.3.9.3 Examples

- "9.58"
- "5.00"

2.70.3.10 Fenwick_M

2.70.3.10.1 Description

This parameter describes the dose-response steepness in the Fenwick model. Fenwick et al. 2008 (doi:10.1016/j.clon.2008.12.011) provide values: 0.392 for local progression free survival at 30 months for NSCLC tumours and 0.280 for head-and-neck tumours.

2.70.3.10.2 Default

- "0.392"

2.70.3.10.3 Examples

- "0.392"
- "0.280"

2.70.3.11 Fenwick_Vref

2.70.3.11.1 Description

This parameter is the volume (in DICOM units; usually mm^3) of a reference tumour (i.e., GTV; primary tumour and involved nodes) which the D_{50} are estimated using. In other words, this is a 'nominal' tumour volume. Fenwick et al. 2008 (doi:10.1016/j.clon.2008.12.011) recommend 148'410 mm^3 (i.e., a sphere of diameter 6.6 cm). However, an appropriate value depends on the nature of the tumour.

2.70.3.11.2 Default

- "148410.0"

2.70.3.11.3 Examples

- `"148410.0"`

2.70.3.12 UserComment

2.70.3.12.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data. If left empty, the column will be omitted from the output.

2.70.3.12.2 Default

- `""`

2.70.3.12.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.71 ExportFITSImages

2.71.1 Description

This operation writes image arrays to FITS-formatted image files.

2.71.2 Notes

- Only pixel information and basic image positioning metadata are exported. In particular, contours and arbitrary metadata are **not** exported by this routine. (If a rendering of the image with contours drawn is needed, consult the PresentationImage operation.)

2.71.3 Parameters

- ImageSelection
- FilenameBase

2.71.3.1 ImageSelection

2.71.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.71.3.1.2 Default

- `"last"`

2.71.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.71.3.2 FilenameBase

2.71.3.2.1 Description

The base filename that images will be written to. A sequentially-increasing number and file suffix are appended after the base filename. Note that the file type is FITS.

2.71.3.2.2 Default

- `"/tmp/dcma_exportfitsimages"`

2.71.3.2.3 Examples

- `"../somedir/out"`
- `"/path/to/some/dir/file_prefix"`

2.72 ExportSurfaceMeshes

2.72.1 Description

This operation writes a surface mesh to a file.

2.72.2 Parameters

- MeshSelection
- Filename

2.72.2.1 MeshSelection

2.72.2.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.72.2.1.2 Default

- `"last"`

2.72.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.72.2.2 Filename

2.72.2.2.1 Description

The filename (or full path name) to which the surface mesh data should be written. The file format is an ASCII OFF model. If no name is given, unique names will be chosen automatically.

2.72.2.2.2 Default

- `""`

2.72.2.2.3 Examples

- `"smesh.off"`
- `"../somedir/mesh.off"`
- `"/path/to/some/surface_mesh.off"`

2.73 ExtractRadiomicFeatures

2.73.1 Description

This operation extracts radiomic features from the selected images. Features are implemented as per specification in the Image Biomarker Standardisation Initiative (IBSI) or pyradiomics documentation if the IBSI specification is unclear or ambiguous.

2.73.2 Notes

- This routine is meant to be processed by an external analysis.

- If this routine is slow, simplifying ROI contours may help speed surface-mesh-based feature extraction. Often removing the highest-frequency components of the contour will help, such as edges that conform tightly to individual voxels.

2.73.3 Parameters

- UserComment
- FeaturesFileName
- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex

2.73.3.1 UserComment

2.73.3.1.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.73.3.1.2 Default

- `""`

2.73.3.1.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.73.3.2 FeaturesFileName

2.73.3.2.1 Description

Features will be appended to this file. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.73.3.2.2 Default

- `""`

2.73.3.2.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.73.3.3 ImageSelection

2.73.3.3.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.73.3.3.2 Default

- `"last"`

2.73.3.3.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.73.3.4 NormalizedROILabelRegex

2.73.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.73.3.4.2 Default

- `".*"`

2.73.3.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.73.3.5 ROILabelRegex

2.73.3.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.73.3.5.2 Default

- `".*"`

2.73.3.5.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.74 FVPicketFence

2.74.1 Description

This operation performs a picket fence QA test using an RTIMAGE file.

2.74.2 Notes

- This is a 'simplified' version of the full picket fence analysis program that uses defaults that are expected to be reasonable across a wide range of scenarios.

2.74.3 Parameters

- ROILabel
- ImageSelection
- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Replacement
- Replace
- NeighbourCount
- AgreementCount
- MaxDistance
- ImageSelection
- DICOMMargin
- RTIMAGE
- ImageSelection
- RowsL
- RowsH
- ColumnsL
- ColumnsH
- DICOMMargin
- ImageSelection
- MLCModel
- MLCROILabel
- JunctionROILabel
- PeakROILabel
- MinimumJunctionSeparation
- ThresholdDistance
- LeafGapsFileName
- ResultsSummaryFileName
- UserComment

- InteractivePlots
- ScaleFactor
- ImageFileName
- ColourMapRegex
- WindowLow
- WindowHigh

2.74.3.1 ROILabel

2.74.3.1.1 Description

A label to attach to the ROI contours.

2.74.3.1.2 Default

- `"entire_image"`

2.74.3.1.3 Examples

- `"everything"`
- `"whole_images"`
- `"unspecified"`

2.74.3.2 ImageSelection

2.74.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.74.3.2.2 Default

- "last"

2.74.3.2.3 Examples

- "last"
- "first"
- "all"
- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.74.3.3 ImageSelection

2.74.3.3.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.74.3.3.2 Default

- "last"

2.74.3.3.3 Examples

- "last"
- "first"
- "all"

- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.74.3.4 NormalizedROILabelRegex

2.74.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.74.3.4.2 Default

- ".*"

2.74.3.4.3 Examples

- ".*"
- ".*Body.*"
- "Body"
- "Gross_Liver"
- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.74.3.5 ROILabelRegex

2.74.3.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.74.3.5.2 Default

- "entire_image"

2.74.3.5.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.74.3.6 Channel

2.74.3.6.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.74.3.6.2 Default

- `"0"`

2.74.3.6.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.74.3.7 Replacement

2.74.3.7.1 Description

Controls how replacements are generated. ‘Mean’ and ‘median’ replacement strategies replace the voxel value with the mean and median, respectively, from the surrounding neighbourhood. ‘Conservative’ refers to the so-called conservative filter that suppresses isolated peaks; for every voxel considered, the voxel intensity is clamped to the local neighbourhood’s extrema. This filter works best for removing spurious peak and trough voxels and performs no averaging. A numeric value can also be supplied, which will replace all isolated or well-connected voxels.

2.74.3.7.2 Default

- `"conservative"`

2.74.3.7.3 Examples

- `"mean"`
- `"median"`
- `"conservative"`
- `"0.0"`
- `"-1.23"`
- `"1E6"`
- `"nan"`

2.74.3.8 Replace

2.74.3.8.1 Description

Controls whether isolated or well-connected voxels are retained.

2.74.3.8.2 Default

- `"isolated"`

2.74.3.8.3 Examples

- `"isolated"`
- `"well-connected"`

2.74.3.9 NeighbourCount

2.74.3.9.1 Description

Controls the number of neighbours being considered. For purposes of speed, this option is limited to specific levels of neighbour adjacency.

2.74.3.9.2 Default

- `"isolated"`

2.74.3.9.3 Examples

- `"1"`
- `"2"`
- `"3"`

2.74.3.10 AgreementCount

2.74.3.10.1 Description

Controls the number of neighbours that must be in agreement for a voxel to be considered ‘well-connected.’

2.74.3.10.2 Default

- `"6"`

2.74.3.10.3 Examples

- `"1"`
- `"2"`
- `"25"`

2.74.3.11 MaxDistance

2.74.3.11.1 Description

The maximum distance (inclusive, in DICOM units: mm) within which neighbouring voxels will be evaluated. For spherical neighbourhoods, this distance refers to the radius. For cubic neighbourhoods, this distance refers to ‘box radius’ or the distance from the cube centre to the nearest point on each bounding face. Voxels separated by more than this distance will not be evaluated together.

2.74.3.11.2 Default

- `"2.0"`

2.74.3.11.3 Examples

- `"0.5"`
- `"2.0"`
- `"15.0"`

2.74.3.12 ImageSelection

2.74.3.12.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or ‘4D’ time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals.

Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.74.3.12.2 Default

- `"last"`

2.74.3.12.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.74.3.13 DICOMMargin

2.74.3.13.1 Description

The amount of margin (in the DICOM coordinate system) to spare from cropping.

2.74.3.13.2 Default

- `"0.0"`

2.74.3.13.3 Examples

- `"0.1"`
- `"2.0"`
- `"-0.5"`
- `"20.0"`

2.74.3.14 RTIMAGE

2.74.3.14.1 Description

If true, attempt to crop the image using information embedded in an RTIMAGE. This option cannot be used with the other options.

2.74.3.14.2 Default

- `"true"`

2.74.3.14.3 Examples

- `"true"`
- `"false"`

2.74.3.15 ImageSelection

2.74.3.15.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.74.3.15.2 Default

- `"last"`

2.74.3.15.3 Examples

- `"last"`
- `"first"`
- `"all"`

- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.74.3.16 RowsL

2.74.3.16.1 Description

The number of rows to remove, starting with the first row. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first row can be either on the top or bottom of the image.

2.74.3.16.2 Default

- "5px"

2.74.3.16.3 Examples

- "0px"
- "10px"
- "100px"
- "15%"
- "15.75%"
- "123.45"

2.74.3.17 RowsH

2.74.3.17.1 Description

The number of rows to remove, starting with the last row. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first row can be either on the top or bottom of the image.

2.74.3.17.2 Default

- "5px"

2.74.3.17.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.74.3.18 ColumnsL

2.74.3.18.1 Description

The number of columns to remove, starting with the first column. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first column can be either on the top or bottom of the image.

2.74.3.18.2 Default

- `"5px"`

2.74.3.18.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.74.3.19 ColumnsH

2.74.3.19.1 Description

The number of columns to remove, starting with the last column. Can be absolute (px), percentage (%), or distance in terms of the DICOM coordinate system. Note the DICOM coordinate system can be flipped, so the first column can be either on the top or bottom of the image.

2.74.3.19.2 Default

- `"5px"`

2.74.3.19.3 Examples

- `"0px"`
- `"10px"`
- `"100px"`
- `"15%"`
- `"15.75%"`
- `"123.45"`

2.74.3.20 DICOMMargin

2.74.3.20.1 Description

The amount of margin (in the DICOM coordinate system) to spare from cropping.

2.74.3.20.2 Default

- `"0.0"`

2.74.3.20.3 Examples

- `"0.1"`
- `"2.0"`
- `"-0.5"`
- `"20.0"`

2.74.3.21 ImageSelection

2.74.3.21.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.74.3.21.2 Default

- `"last"`

2.74.3.21.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.74.3.22 MLCModel

2.74.3.22.1 Description

The MLC design geometry to use. ‘VarianMillenniumMLC80’ has 40 leafs in each bank; leaves are 10mm wide at isocentre; and the maximum static field size is 40cm x 40cm. ‘VarianMillenniumMLC120’ has 60 leafs in each bank; the 40 central leaves are 5mm wide at isocentre; the 20 peripheral leaves are 10mm wide; and the maximum static field size is 40cm x 40cm. ‘VarianHD120’ has 60 leafs in each bank; the 32 central leaves are 2.5mm wide at isocentre; the 28 peripheral leaves are 5mm wide; and the maximum static field size is 40cm x 22cm.

2.74.3.22.2 Default

- `"VarianMillenniumMLC120"`

2.74.3.22.3 Examples

- `"VarianMillenniumMLC80"`
- `"VarianMillenniumMLC120"`
- `"VarianHD120"`

2.74.3.23 MLCROILabel

2.74.3.23.1 Description

An ROI imitating the MLC axes of leaf pairs is created. This is the label to apply to it. Note that the leaves are modeled with thin contour rectangles of virtually zero area. Also note that the outline colour is significant and denotes leaf pair pass/fail.

2.74.3.23.2 Default

- `"Leaves"`

2.74.3.23.3 Examples

- `"MLC_leaves"`
- `"MLC"`
- `"approx_leaf_axes"`

2.74.3.24 JunctionROILabel

2.74.3.24.1 Description

An ROI imitating the junction is created. This is the label to apply to it. Note that the junctions are modeled with thin contour rectangles of virtually zero area.

2.74.3.24.2 Default

- `"Junction"`

2.74.3.24.3 Examples

- `"Junction"`
- `"Picket_Fence_Junction"`

2.74.3.25 PeakROILabel

2.74.3.25.1 Description

ROIs encircling the leaf profile peaks are created. This is the label to apply to it. Note that the peaks are modeled with small squares.

2.74.3.25.2 Default

- `"Peak"`

2.74.3.25.3 Examples

- `"Peak"`
- `"Picket_Fence_Peak"`

2.74.3.26 MinimumJunctionSeparation

2.74.3.26.1 Description

The minimum distance between junctions on the SAD isoplane in DICOM units (mm). This number is used to de-duplicate automatically detected junctions. Analysis results should not be sensitive to the specific value.

2.74.3.26.2 Default

- `"10.0"`

2.74.3.26.3 Examples

- `"5.0"`
- `"10.0"`
- `"15.0"`
- `"25.0"`

2.74.3.27 ThresholdDistance

2.74.3.27.1 Description

The threshold distance in DICOM units (mm) above which MLC separations are considered to 'fail'. Each leaf pair is evaluated separately. Pass/fail status is also indicated by setting the leaf axis contour colour (blue for pass, red for fail).

2.74.3.27.2 Default

- `"0.5"`

2.74.3.27.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`

2.74.3.28 LeafGapsFileName

2.74.3.28.1 Description

This file will contain gap and nominal-vs-actual offset distances for each leaf pair. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.74.3.28.2 Default

- `""`

2.74.3.28.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.74.3.29 ResultsSummaryFileName

2.74.3.29.1 Description

This file will contain a brief summary of the results. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.74.3.29.2 Default

- `""`

2.74.3.29.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.74.3.30 UserComment

2.74.3.30.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.74.3.30.2 Default

- `""`

2.74.3.30.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.74.3.31 InteractivePlots

2.74.3.31.1 Description

Whether to interactively show plots showing detected edges.

2.74.3.31.2 Default

- `"false"`

2.74.3.31.3 Examples

- `"true"`
- `"false"`

2.74.3.32 ScaleFactor

2.74.3.32.1 Description

This factor is applied to the image width and height to magnify (larger than 1) or shrink (less than 1) the image. This factor only affects the output image size. Note that aspect ratio is retained, but rounding for non-integer factors may lead to small (1-2 pixel) discrepancies.

2.74.3.32.2 Default

- `"1.5"`

2.74.3.32.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`
- `"5.23"`

2.74.3.33 ImageFileName

2.74.3.33.1 Description

The file name to use for the image. If blank, a filename will be generated sequentially.

2.74.3.33.2 Default

- `""`

2.74.3.33.3 Examples

- `""`
- `"/tmp/an_image.png"`
- `"afile.png"`

2.74.3.34 ColourMapRegex

2.74.3.34.1 Description

The colour mapping to apply to the image if there is a single channel. The default will match the first available, and if there is no matching map found, the first available will be selected.

2.74.3.34.2 Default

- `".*"`

2.74.3.34.3 Examples

- `"Viridis"`
- `"Magma"`
- `"Plasma"`
- `"Inferno"`
- `"Jet"`
- `"MorelandBlueRed"`
- `"MorelandBlackBody"`
- `"MorelandExtendedBlackBody"`
- `"KRC"`
- `"ExtendedKRC"`
- `"Kovesi_LinKRYW_5-100_c64"`
- `"Kovesi_LinKRYW_0-100_c71"`
- `"Kovesi_Cyclic_cet-c2"`

- `"LANLOliveGreentoBlue"`
- `"YgorIncandescent"`
- `"LinearRamp"`

2.74.3.35 WindowLow

2.74.3.35.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or lower will be assigned the lowest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.74.3.35.2 Default

- `""`

2.74.3.35.3 Examples

- `""`
- `"-1.23"`
- `"0"`
- `"1E4"`

2.74.3.36 WindowHigh

2.74.3.36.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or higher will be assigned the highest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.74.3.36.2 Default

- `""`

2.74.3.36.3 Examples

- `""`
- `"1.23"`
- `"0"`
- `"10.3E4"`

2.75 GenerateCalibrationCurve

2.75.1 Description

This operation uses two overlapping images volumes to generate a calibration curve mapping from the first image volume to the second. Only the region within the specified ROI(s) is considered.

2.75.2 Notes

- ROI(s) are interpretted relative to the mapped-to ('reference' or 'fixed') image. The reason for this is that typically the reference images are associated with contours (e.g., planning data) and the mapped-from images do not (e.g., CBCTs that have been registered).
- This routine can handle overlapping or duplicate contours.

2.75.3 Parameters

- Channel
- ImageSelection
- RefImageSelection
- ContourOverlap
- Inclusivity
- CalibCurveFileName
- NormalizedROILabelRegex
- ROILabelRegex

2.75.3.1 Channel

2.75.3.1.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.75.3.1.2 Default

- `"-1"`

2.75.3.1.3 Examples

- `"-1"`
- `"0"`

- "1"
- "2"

2.75.3.2 ImageSelection

2.75.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax. Note that these images are the 'mapped-from' or 'moving' images.

2.75.3.2.2 Default

- "last"

2.75.3.2.3 Examples

- "last"
- "first"
- "all"
- "none"
- "#0"
- "#-0"
- "!last"
- "!#-3"
- "key@.*value.*"
- "key1@.*value1.*;key2@^value2$;first"

2.75.3.3 RefImageSelection

2.75.3.3.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax. Note that these images are the 'mapped-to' or 'fixed' images.

2.75.3.3.2 Default

- `"last"`

2.75.3.3.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.75.3.4 ContourOverlap

2.75.3.4.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes).

The option ‘overlapping_contours_cancel’ ignores orientation and cancels all contour overlap.

2.75.3.4.2 Default

- `"ignore"`

2.75.3.4.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.75.3.5 Inclusivity

2.75.3.5.1 Description

Controls how voxels are deemed to be ‘within’ the interior of the selected ROI(s). The default ‘center’ considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, ‘planar_corner_inclusive’, considers a voxel interior if ANY corner is interior. The second, ‘planar_corner_exclusive’, considers a voxel interior if ALL (four) corners are interior.

2.75.3.5.2 Default

- `"center"`

2.75.3.5.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.75.3.6 CalibCurveFileName

2.75.3.6.1 Description

The file to which a calibration curve will be written to. The format is line-based with 4 numbers per line: (original pixel value) (uncertainty) (new pixel value) (uncertainty). Uncertainties refer to the prior number and may be uniformly zero if unknown. Lines beginning with '#' are comments. The curve is meant to be interpolated. (Later attempts to extrapolate may result in failure.)

2.75.3.6.2 Default

- `""`

2.75.3.6.3 Examples

- `"./calib.dat"`

2.75.3.7 NormalizedROILabelRegex

2.75.3.7.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.75.3.7.2 Default

- `".*"`

2.75.3.7.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.75.3.8 ROILabelRegex

2.75.3.8.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.75.3.8.2 Default

- `".*"`

2.75.3.8.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.76 GenerateSurfaceMask

2.76.1 Description

This operation generates a surface image mask, which contains information about whether each voxel is within, on, or outside the selected ROI(s).

2.76.2 Parameters

- BackgroundVal
- InteriorVal
- SurfaceVal
- NormalizedROILabelRegex
- ROILabelRegex

2.76.2.1 BackgroundVal

2.76.2.1.1 Description

The value to give to voxels neither inside nor on the surface of the ROI(s).

2.76.2.1.2 Default

- "0.0"

2.76.2.1.3 Examples

- "0.0"
- "-1.0"
- "1.23"
- "2.34E26"

2.76.2.2 InteriorVal

2.76.2.2.1 Description

The value to give to voxels within the volume of the ROI(s) but not on the surface.

2.76.2.2.2 Default

- "1.0"

2.76.2.2.3 Examples

- "0.0"
- "-1.0"
- "1.23"
- "2.34E26"

2.76.2.3 SurfaceVal

2.76.2.3.1 Description

The value to give to voxels on the surface/boundary of ROI(s).

2.76.2.3.2 Default

- "2.0"

2.76.2.3.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.76.2.4 NormalizedROILabelRegex

2.76.2.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.76.2.4.2 Default

- `".*"`

2.76.2.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.76.2.5 ROILabelRegex

2.76.2.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.76.2.5.2 Default

- `".*"`

2.76.2.5.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.77 GenerateSyntheticImages

2.77.1 Description

This operation generates a synthetic, regular bitmap image array. It can be used for testing how images are quantified or transformed.

2.77.2 Parameters

- NumberOfImages
- NumberOfRows
- NumberOfColumns
- NumberOfChannels
- SliceThickness
- SpacingBetweenSlices
- VoxelWidth
- VoxelHeight
- ImageAnchor
- ImagePosition
- ImageOrientationColumn
- ImageOrientationRow
- InstanceNumber
- AcquisitionNumber
- VoxelValue
- StipleValue
- Metadata

2.77.2.1 NumberOfImages

2.77.2.1.1 Description

The number of images to create.

2.77.2.1.2 Default

- `"100"`

2.77.2.1.3 Examples

- `"1"`
- `"100"`
- `"1000"`

2.77.2.2 NumberOfRows

2.77.2.2.1 Description

The number of rows each image should contain.

2.77.2.2.2 Default

- `"256"`

2.77.2.2.3 Examples

- `"1"`
- `"100"`
- `"1000"`

2.77.2.3 NumberOfColumns

2.77.2.3.1 Description

The number of columns each image should contain.

2.77.2.3.2 Default

- `"256"`

2.77.2.3.3 Examples

- `"1"`
- `"100"`
- `"1000"`

2.77.2.4 NumberOfChannels

2.77.2.4.1 Description

The number of channels each image should contain.

2.77.2.4.2 Default

- `"1"`

2.77.2.4.3 Examples

- `"1"`
- `"10"`
- `"100"`

2.77.2.5 SliceThickness

2.77.2.5.1 Description

Image slices will be have this thickness (in DICOM units: mm). For most purposes, SliceThickness should be equal to SpacingBetweenSlices. If SpacingBetweenSlices is smaller than SliceThickness, images will overlap. If SpacingBetweenSlices is larger than SliceThickness, there will be a gap between images.

2.77.2.5.2 Default

- `"1.0"`

2.77.2.5.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.77.2.6 SpacingBetweenSlices

2.77.2.6.1 Description

Image slice centres will be separated by this distance (in DICOM units: mm). For most purposes, SpacingBetweenSlices should be equal to SliceThickness. If SpacingBetweenSlices is smaller than SliceThickness, images will overlap. If SpacingBetweenSlices is larger than SliceThickness, there will be a gap between images.

2.77.2.6.2 Default

- `"1.0"`

2.77.2.6.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.77.2.7 VoxelWidth

2.77.2.7.1 Description

Voxels will have this (in-plane) width (in DICOM units: mm). This means that row-adjacent voxels centres will be separated by VoxelWidth). Each voxel will have dimensions: VoxelWidth x VoxelHeight x SliceThickness.

2.77.2.7.2 Default

- `"1.0"`

2.77.2.7.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.77.2.8 VoxelHeight

2.77.2.8.1 Description

Voxels will have this (in-plane) height (in DICOM units: mm). This means that column-adjacent voxels centres will be separated by VoxelHeight). Each voxel will have dimensions: VoxelWidth x VoxelHeight x SliceThickness.

2.77.2.8.2 Default

- `"1.0"`

2.77.2.8.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.77.2.9 ImageAnchor

2.77.2.9.1 Description

A point in 3D space which denotes the origin (in DICOM units: mm). All other vectors are taken to be relative to this point. Under most circumstance the anchor should be (0,0,0). Specify coordinates separated by commas.

2.77.2.9.2 Default

- `"0.0, 0.0, 0.0"`

2.77.2.9.3 Examples

- `"0.0, 0.0, 0.0"`
- `"0.0,0.0,0.0"`
- `"1.0, -2.3, 45.6"`

2.77.2.10 ImagePosition

2.77.2.10.1 Description

The centre of the row=0, column=0 voxel in the first image (in DICOM units: mm). Specify coordinates separated by commas.

2.77.2.10.2 Default

- `"0.0, 0.0, 0.0"`

2.77.2.10.3 Examples

- `"0.0, 0.0, 0.0"`
- `"100.0,100.0,100.0"`
- `"1.0, -2.3, 45.6"`

2.77.2.11 ImageOrientationColumn

2.77.2.11.1 Description

The orientation unit vector that is aligned with image columns. Care should be taken to ensure ImageOrientationRow and ImageOrientationColumn are orthogonal. (A Gram-Schmidt orthogonalization procedure ensures they are, but the image orientation may not match the expected orientation.) Note that the magnitude will also be scaled to unit length for convenience. Specify coordinates separated by commas.

2.77.2.11.2 Default

- `"1.0, 0.0, 0.0"`

2.77.2.11.3 Examples

- `"1.0, 0.0, 0.0"`
- `"1.0, 1.0, 0.0"`
- `"0.0, 0.0, -1.0"`

2.77.2.12 ImageOrientationRow

2.77.2.12.1 Description

The orientation unit vector that is aligned with image rows. Care should be taken to ensure ImageOrientationRow and ImageOrientationColumn are orthogonal. (A Gram-Schmidt orthogonalization procedure ensures they are, but the image orientation may not match the expected orientation.) Note that the magnitude will also be scaled to unit length for convenience. Specify coordinates separated by commas.

2.77.2.12.2 Default

- `"0.0, 1.0, 0.0"`

2.77.2.12.3 Examples

- `"0.0, 1.0, 0.0"`
- `"0.0, 1.0, 1.0"`
- `"-1.0, 0.0, 0.0"`

2.77.2.13 InstanceNumber

2.77.2.13.1 Description

A number affixed to the first image, and then incremented and affixed for each subsequent image.

2.77.2.13.2 Default

- `"1"`

2.77.2.13.3 Examples

- `"1"`
- `"100"`
- `"1234"`

2.77.2.14 AcquisitionNumber

2.77.2.14.1 Description

A number affixed to all images, meant to indicate membership in a single acquisition.

2.77.2.14.2 Default

- `"1"`

2.77.2.14.3 Examples

- `"1"`
- `"100"`
- `"1234"`

2.77.2.15 VoxelValue

2.77.2.15.1 Description

The value that is assigned to all voxels, or possibly every other voxel. Note that if StipleValue is given a finite value, only half the voxels will be assigned a value of VoxelValue and the other half will be assigned a value of StipleValue. This produces a checkerboard pattern.

2.77.2.15.2 Default

- `"0.0"`

2.77.2.15.3 Examples

- `"0.0"`
- `"1.0E4"`
- `"-1234"`
- `"nan"`

2.77.2.16 StipleValue

2.77.2.16.1 Description

The value that is assigned to every other voxel. If StipleValue is given a finite value, half of all voxels will be assigned a value of VoxelValue and the other half will be assigned a value of StipleValue. This produces a checkerboard pattern.

2.77.2.16.2 Default

- `"nan"`

2.77.2.16.3 Examples

- `"1.0"`
- `"-1.0E4"`
- `"1234"`

2.77.2.17 Metadata

2.77.2.17.1 Description

A semicolon-separated list of ‘key@value’ metadata to imbue into each image. This metadata will overwrite any existing keys with the provided values.

2.77.2.17.2 Default

- `""`

2.77.2.17.3 Examples

- `"keyA@valueA;keyB@valueB"`

2.78 GenerateVirtualDataContourViaThresholdTestV1

2.78.1 Description

This operation generates data suitable for testing the ContourViaThreshold operation.

2.78.2 Parameters

No registered options.

2.79 GenerateVirtualDataDoseStairsV1

2.79.1 Description

This operation generates a dosimetric stairway. It can be used for testing how dosimetric data is transformed.

2.79.2 Parameters

No registered options.

2.80 GenerateVirtualDataImageSphereV1

2.80.1 Description

This operation generates a bitmap image of a sphere. It can be used for testing how images are quantified or transformed.

2.80.2 Parameters

No registered options.

2.81 GenerateVirtualDataPerfusionV1

2.81.1 Description

This operation generates data suitable for testing perfusion modeling operations. There are no specific checks in this code. Another operation performs the actual validation. You might be able to manually verify if the perfusion model admits a simple solution.

2.81.2 Parameters

No registered options.

2.82 GiveWholeImageArrayABoneWindowLevel

2.82.1 Description

This operation runs the images in an image array through a uniform window-and-leveler instead of per-slice window-and-level or no window-and-level at all. Data is modified and no copy is made!

2.82.2 Parameters

No registered options.

2.83 GiveWholeImageArrayAHeadAndNeckWindowLevel

2.83.1 Description

This operation runs the images in an image array through a uniform window-and-leveler instead of per-slice window-and-level or no window-and-level at all. Data is modified and no copy is made!

2.83.2 Parameters

No registered options.

2.84 GiveWholeImageArrayAThoraxWindowLevel

2.84.1 Description

This operation runs the images in an image array through a uniform window-and-leveler instead of per-slice window-and-level or no window-and-level at all. Data is modified and no copy is made!

2.84.2 Parameters

No registered options.

2.85 GiveWholeImageArrayAnAbdominalWindowLevel

2.85.1 Description

This operation runs the images in an image array through a uniform window-and-leveler instead of per-slice window-and-level or no window-and-level at all. Data is modified and no copy is made!

2.85.2 Parameters

No registered options.

2.86 GridBasedRayCastDoseAccumulate

2.86.1 Description

This operation performs a ray casting to estimate the surface dose of an ROI.

2.86.2 Parameters

- DoseMapFileName
- DoseLengthMapFileName
- LengthMapFileName
- NormalizedReferenceROILabelRegex
- NormalizedROILabelRegex
- ReferenceROILabelRegex
- ROILabelRegex
- SmallestFeature
- RaydL
- GridRows
- GridColumns
- SourceDetectorRows
- SourceDetectorColumns
- NumberOfImages

2.86.2.1 DoseMapFileName

2.86.2.1.1 Description

A filename (or full path) for the dose image map. Note that this file is approximate, and may not be accurate. There is more information available when you use the length and dose*length maps instead. However, this file is useful for viewing and eyeballing tuning settings. The format is FITS. Leave empty to dump to generate a unique temporary file.

2.86.2.1.2 Default

- `""`

2.86.2.1.3 Examples

- `""`
- `"/tmp/dose.fits"`
- `"localfile.fits"`
- `"derivative_data.fits"`

2.86.2.2 DoseLengthMapFileName

2.86.2.2.1 Description

A filename (or full path) for the (dose)*(length traveled through the ROI peel) image map. The format is FITS. Leave empty to dump to generate a unique temporary file.

2.86.2.2.2 Default

- `""`

2.86.2.2.3 Examples

- `""`
- `"/tmp/doselength.fits"`
- `"localfile.fits"`
- `"derivative_data.fits"`

2.86.2.3 LengthMapFileName

2.86.2.3.1 Description

A filename (or full path) for the (length traveled through the ROI peel) image map. The format is FITS. Leave empty to dump to generate a unique temporary file.

2.86.2.3.2 Default

- `""`

2.86.2.3.3 Examples

- `""`
- `"/tmp/surfacelength.fits"`
- `"localfile.fits"`
- `"derivative_data.fits"`

2.86.2.4 NormalizedReferenceROILabelRegex

2.86.2.4.1 Description

A regex matching reference ROI labels/names to consider. The default will match all available ROIs, which is non-sensical. The reference ROI is used to orient the cleaving plane to trim the grid surface mask.

2.86.2.4.2 Default

- `".*"`

2.86.2.4.3 Examples

- `".*"`
- `".*Prostate.*"`
- `"Left Kidney"`
- `"Gross Liver"`

2.86.2.5 NormalizedROILabelRegex

2.86.2.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.86.2.5.2 Default

- `".*"`

2.86.2.5.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.86.2.6 ReferenceROILabelRegex

2.86.2.6.1 Description

A regex matching reference ROI labels/names to consider. The default will match all available ROIs, which is non-sensical. The reference ROI is used to orient the cleaving plane to trim the grid surface mask.

2.86.2.6.2 Default

- `".*"`

2.86.2.6.3 Examples

- `".*"`
- `".*[pP]rostate.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.86.2.7 ROILabelRegex

2.86.2.7.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.86.2.7.2 Default

- `".*"`

2.86.2.7.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.86.2.8 SmallestFeature

2.86.2.8.1 Description

A length giving an estimate of the smallest feature you want to resolve. Quantity is in the DICOM coordinate system.

2.86.2.8.2 Default

- `"0.5"`

2.86.2.8.3 Examples

- `"1.0"`
- `"2.0"`
- `"0.5"`
- `"5.0"`

2.86.2.9 RaydL

2.86.2.9.1 Description

The distance to move a ray each iteration. Should be << img_thickness and << cylinder_radius. Making too large will invalidate results, causing rays to pass through the surface without registering any dose accumulation. Making too small will cause the run-time to grow and may eventually lead to truncation or round-off errors. Quantity is in the DICOM coordinate system.

2.86.2.9.2 Default

- `"0.1"`

2.86.2.9.3 Examples

- `"0.1"`
- `"0.05"`
- `"0.01"`
- `"0.005"`

2.86.2.10 GridRows

2.86.2.10.1 Description

The number of rows in the surface mask grid images.

2.86.2.10.2 Default

- `"512"`

2.86.2.10.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.86.2.11 GridColumns

2.86.2.11.1 Description

The number of columns in the surface mask grid images.

2.86.2.11.2 Default

- `"512"`

2.86.2.11.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.86.2.12 SourceDetectorRows

2.86.2.12.1 Description

The number of rows in the resulting images. Setting too fine relative to the surface mask grid or dose grid is futile.

2.86.2.12.2 Default

- `"1024"`

2.86.2.12.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.86.2.13 SourceDetectorColumns

2.86.2.13.1 Description

The number of columns in the resulting images. Setting too fine relative to the surface mask grid or dose grid is futile.

2.86.2.13.2 Default

- `"1024"`

2.86.2.13.3 Examples

- `"10"`
- `"50"`
- `"128"`
- `"1024"`

2.86.2.14 NumberOfImages

2.86.2.14.1 Description

The number of images used for grid-based surface detection. Leave negative for computation of a reasonable value; set to something specific to force an override.

2.86.2.14.2 Default

- `"-1"`

2.86.2.14.3 Examples

- `"-1"`
- `"10"`
- `"50"`
- `"100"`

2.87 GroupImages

2.87.1 Description

This operation will group individual image slices into partitions (Image_Arrays) based on the values of the specified metadata tags. DICOMautomaton operations are usually performed on containers rather than individual images, and grouping can express connections between images. For example a group could contain the scans belonging to a whole study, one of the series in a study, individual image volumes within a single series (e.g., a 3D volume in a temporal perfusion scan), or individual slices. A group could also contain all the slices that intersect a given plane, or were taken on a specified StationName.

2.87.2 Notes

- Images are moved, not copied.
- Image order within a group is retained (i.e., stable grouping), but groups are appended to the back of the Image_Array list according to the default sort for the group's key-value value.
- Images that do not contain the specified metadata will be grouped into a special N/A group at the end.

2.87.3 Parameters

- ImageSelection
- KeysCommon
- Enforce

2.87.3.1 ImageSelection

2.87.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.87.3.1.2 Default

- `"all"`

2.87.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.87.3.2 KeysCommon

2.87.3.2.1 Description

Image metadata keys to use for exact-match groupings. For each group that is produced, every image will share the same key-value pair. This is generally useful for non-numeric (or integer, date, etc.) key-values. A ';'-delimited list can be specified to group on multiple criteria simultaneously. An empty string disables metadata-based grouping.

2.87.3.2.2 Default

- `""`

2.87.3.2.3 Examples

- `"SeriesNumber"`
- `"BodyPartExamined;StudyDate"`
- `"SeriesInstanceUID"`
- `"StationName"`

2.87.3.3 Enforce

2.87.3.3.1 Description

Other specialized grouping operations that involve custom logic. Currently, only 'no-overlap' is available, but it has two variants. Both partition based on the spatial extent of images; in each non-overlapping partition, no two images will spatially overlap. 'No-overlap-as-is' will effectively insert partitions without altering the order. A partition is inserted when an image is found to overlap with an image already within the partition. For this grouping to be useful, images must be sorted so that partitions can be inserted without any necessary reordering. An example of when this grouping is useful is CT shuttling in which the ordering of images alternate between increasing and decreasing SliceNumber. Note that, depending on the ordering, some partitions may therefore be incomplete. 'No-overlap-adjust' will rearrange images so that the first partition is always complete. This is achieved by building a queue of spatially-overlapping images and greedily stealing one of each kind when constructing partitions. An example of when this grouping is useful are 4DCTs which have been acquired for all phases while the couch remains at a single SliceNumber; the images are ordered on disk in the acquisition order (i.e., alike SliceNumbers are bunched together) but the logical analysis order is that each contiguous volume should represent a single phase. An empty string disables logic-based grouping.

2.87.3.3.2 Default

- `""`

2.87.3.3.3 Examples

- `"no-overlap-as-is"`
- `"no-overlap-adjust"`

2.88 GrowContours

2.88.1 Description

This routine will grow (or shrink) 2D contours in their plane by the specified amount. Growth is accomplish by translating vertices away from the interior by the specified amount. The direction is chosen to be the direction opposite of the in-plane normal produced by averaging the line segments connecting the contours.

2.88.2 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- Distance

2.88.2.1 NormalizedROILabelRegex

2.88.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.88.2.1.2 Default

- `".*"`

2.88.2.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.88.2.2 ROILabelRegex

2.88.2.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.88.2.2.2 Default

- `".*"`

2.88.2.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.88.2.3 Distance

2.88.2.3.1 Description

The distance to translate contour vertices. (The direction is outward.)

2.88.2.3.2 Default

- `"0.00354165798657632"`

2.88.2.3.3 Examples

- `"1E-5"`
- `"0.321"`
- `"1.1"`
- `"15.3"`

2.89 HighlightROIs

2.89.1 Description

This operation overwrites voxel data inside and/or outside of ROI(s) to 'highlight' them. It can handle overlapping or duplicate contours.

2.89.2 Parameters

- Channel
- ImageSelection
- ContourOverlap
- Inclusivity
- ExteriorVal
- InteriorVal
- ExteriorOverwrite
- InteriorOverwrite
- NormalizedROILabelRegex
- ROILabelRegex

2.89.2.1 Channel

2.89.2.1.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.89.2.1.2 Default

- `"-1"`

2.89.2.1.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.89.2.2 ImageSelection

2.89.2.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.89.2.2.2 Default

- `"last"`

2.89.2.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.89.2.3 ContourOverlap

2.89.2.3.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.89.2.3.2 Default

- `"ignore"`

2.89.2.3.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`

- "honour_opps"
- "overlap_cancel"

2.89.2.4 Inclusivity

2.89.2.4.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.89.2.4.2 Default

- "center"

2.89.2.4.3 Examples

- "center"
- "centre"
- "planar_corner_inclusive"
- "planar_inc"
- "planar_corner_exclusive"
- "planar_exc"

2.89.2.5 ExteriorVal

2.89.2.5.1 Description

The value to give to voxels outside the specified ROI(s). Note that this value will be ignored if exterior overwrites are disabled.

2.89.2.5.2 Default

- "0.0"

2.89.2.5.3 Examples

- "0.0"
- "-1.0"
- "1.23"
- "2.34E26"

2.89.2.6 InteriorVal

2.89.2.6.1 Description

The value to give to voxels within the volume of the specified ROI(s). Note that this value will be ignored if interior overwrites are disabled.

2.89.2.6.2 Default

- `"1.0"`

2.89.2.6.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.89.2.7 ExteriorOverwrite

2.89.2.7.1 Description

Whether to overwrite voxels exterior to the specified ROI(s).

2.89.2.7.2 Default

- `"true"`

2.89.2.7.3 Examples

- `"true"`
- `"false"`

2.89.2.8 InteriorOverwrite

2.89.2.8.1 Description

Whether to overwrite voxels interior to the specified ROI(s).

2.89.2.8.2 Default

- `"true"`

2.89.2.8.3 Examples

- `"true"`
- `"false"`

2.89.2.9 NormalizedROILabelRegex

2.89.2.9.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.89.2.9.2 Default

- `".*"`

2.89.2.9.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.89.2.10 ROILabelRegex

2.89.2.10.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.89.2.10.2 Default

- `".*"`

2.89.2.10.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.90 ImageRoutineTests

2.90.1 Description

This operation performs a series of sub-operations that are generally useful when inspecting an image.

2.90.2 Parameters

No registered options.

2.91 ImprintImages

2.91.1 Description

This operation creates imprints of point clouds on the selected images. Images are modified where the points are coindicident.

2.91.2 Parameters

- ImageSelection
- PointSelection
- VoxelValue
- Channel

2.91.2.1 ImageSelection

2.91.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.91.2.1.2 Default

- `"last"`

2.91.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.91.2.2 PointSelection

2.91.2.2.1 Description

Select one or more point clouds. Note that point clouds can hold a variety of data with varying attributes, but each point cloud is meant to represent a single logically cohesive collection of points. Be aware that it is possible to mix logically unrelated points together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth point cloud (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last point cloud. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.91.2.2.2 Default

- `"last"`

2.91.2.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.91.2.3 VoxelValue

2.91.2.3.1 Description

The value to give voxels which are coincident with a point from the point cloud. Note that point cloud attributes, if present, may override this value.

2.91.2.3.2 Default

- `"1.0"`

2.91.2.3.3 Examples

- `"-1.0"`
- `"0.0"`
- `"1.23"`
- `"nan"`
- `"inf"`

2.91.2.4 Channel

2.91.2.4.1 Description

The image channel to use. Zero-based.

2.91.2.4.2 Default

- `"0"`

2.91.2.4.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.92 InterpolateSlices

2.92.1 Description

This operation interpolates the slices of an image array using a reference image array, effectively performing trilinear interpolation. This operation is meant to prepare image arrays to be compared or operated on in a per-voxel manner.

2.92.2 Notes

- No images are overwritten by this operation. The outgoing images will inherit (interpolated) voxel values from the selected images and image geometry from the reference images.
- If all images (selected and reference, altogether) are detected to be rectilinear, this operation will avoid in-plane interpolation and will thus be much faster. There is no **need** for rectilinearity, however without it sections of the image that cannot reasonably be interpolated (via plane-orthogonal projection onto the reference images) will be invalid and marked with NaNs. Non-rectilearity which amounts to a differing number of rows or columns will merely be slower to interpolate.

2.92.3 Parameters

- ImageSelection
- ReferenceImageSelection
- Channel

2.92.3.1 ImageSelection

2.92.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.92.3.1.2 Default

- `"all"`

2.92.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.92.3.2 ReferenceImageSelection

2.92.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array.

Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.92.3.2.2 Default

- `"all"`

2.92.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.92.3.3 Channel

2.92.3.3.1 Description

The channel to compare (zero-based). A negative value will result in all channels being interpolated, otherwise unspecified channels are merely default initialized. Note that both test images and reference images will share this specifier.

2.92.3.3.2 Default

- `"-1"`

2.92.3.3.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.93 IsolatedVoxelFilter

2.93.1 Description

This routine applies a filter that discriminates between well-connected and isolated voxels. Isolated voxels can either be filtered out or retained. This operation considers the 3D neighbourhood surrounding a voxel.

2.93.2 Notes

- The provided image collection must be rectilinear.
- If the neighbourhood involves voxels that do not exist, they are treated as NaNs in the same way that voxels with the NaN value are treated.

2.93.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Replacement
- Replace
- NeighbourCount
- AgreementCount
- MaxDistance

2.93.3.1 ImageSelection

2.93.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order

specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.93.3.1.2 Default

- `"last"`

2.93.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.93.3.2 NormalizedROILabelRegex

2.93.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.93.3.2.2 Default

- `".*"`

2.93.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.93.3.3 ROILabelRegex

2.93.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.93.3.3.2 Default

- `".*"`

2.93.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.93.3.4 Channel

2.93.3.4.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.93.3.4.2 Default

- `"0"`

2.93.3.4.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.93.3.5 Replacement

2.93.3.5.1 Description

Controls how replacements are generated. 'Mean' and 'median' replacement strategies replace the voxel value with the mean and median, respectively, from the surrounding neighbourhood. 'Conservative' refers to the so-called conservative filter that suppresses isolated peaks; for every voxel considered, the voxel intensity is clamped to the local neighbourhood's extrema. This filter works best for removing spurious peak and trough voxels and performs no averaging. A numeric value can also be supplied, which will replace all isolated or well-connected voxels.

2.93.3.5.2 Default

- `"mean"`

2.93.3.5.3 Examples

- `"mean"`
- `"median"`
- `"conservative"`
- `"0.0"`
- `"-1.23"`
- `"1E6"`
- `"nan"`

2.93.3.6 Replace

2.93.3.6.1 Description

Controls whether isolated or well-connected voxels are retained.

2.93.3.6.2 Default

- `"isolated"`

2.93.3.6.3 Examples

- `"isolated"`
- `"well-connected"`

2.93.3.7 NeighbourCount

2.93.3.7.1 Description

Controls the number of neighbours being considered. For purposes of speed, this option is limited to specific levels of neighbour adjacency.

2.93.3.7.2 Default

- `"isolated"`

2.93.3.7.3 Examples

- `"1"`
- `"2"`
- `"3"`

2.93.3.8 AgreementCount

2.93.3.8.1 Description

Controls the number of neighbours that must be in agreement for a voxel to be considered ‘well-connected.’

2.93.3.8.2 Default

- `"6"`

2.93.3.8.3 Examples

- `"1"`
- `"2"`
- `"25"`

2.93.3.9 MaxDistance

2.93.3.9.1 Description

The maximum distance (inclusive, in DICOM units: mm) within which neighbouring voxels will be evaluated. For spherical neighbourhoods, this distance refers to the radius. For cubic neighbourhoods, this distance refers to ‘box radius’ or the distance from the cube centre to the nearest point on each bounding face. Voxels separated by more than this distance will not be evaluated together.

2.93.3.9.2 Default

- `"2.0"`

2.93.3.9.3 Examples

- `"0.5"`
- `"2.0"`
- `"15.0"`

2.94 LoadFiles

2.94.1 Description

This operation loads files on-the-fly.

2.94.2 Notes

- This operation requires all files provided to it to exist and be accessible. Inaccessible files are not silently ignored and will cause this operation to fail.

2.94.3 Parameters

- FileName

2.94.3.1 FileName

2.94.3.1.1 Description

This file will be parsed and loaded. All file types supported by the DICOMautomaton system can be loaded in this way. Currently this includes serialized Drover class files, DICOM files, FITS image files, and XYZ point cloud files.

2.94.3.1.2 Default

- `""`

2.94.3.1.3 Examples

- `"/tmp/image.dcm"`
- `"rois.dcm"`
- `"dose.dcm"`
- `"image.fits"`
- `"point_cloud.xyz"`

2.95 LogScale

2.95.1 Description

This operation log-scales pixels for all available image arrays. This functionality is often desired for viewing purposes, to make the pixel level changes appear more linear. Be weary of using for anything quantitative!

2.95.2 Parameters

- ImageSelection

2.95.2.1 ImageSelection

2.95.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.95.2.1.2 Default

- `"last"`

2.95.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.96 MaxMinPixels

2.96.1 Description

This operation replaces pixels with the pixel-wise difference (max)-(min).

2.96.2 Parameters

No registered options.

2.97 MeldDose

2.97.1 Description

This operation melds all available dose image data. At a high level, dose melding sums overlapping pixel values for multi-part dose arrays. For more information about what this specifically entails, refer to the appropriate subroutine.

2.97.2 Parameters

No registered options.

2.98 MinkowskiSum3D

2.98.1 Description

This operation computes a Minkowski sum or symmetric difference of a 3D surface mesh generated from the selected ROIs with a sphere. The effect is that a margin is added or subtracted to the ROIs, causing them to 'grow' outward or 'shrink' inward. Exact and inexact routines can be used.

2.98.2 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- ImageSelection
- Operation
- Distance

2.98.2.1 NormalizedROILabelRegex

2.98.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.98.2.1.2 Default

- `".*"`

2.98.2.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.98.2.2 ROILabelRegex

2.98.2.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses grep syntax.

2.98.2.2.2 Default

- `".*"`

2.98.2.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*parotid.*|.*sub.*mand.*"`
- `"left_parotid|right_parotid|eyes"`

2.98.2.3 ImageSelection

2.98.2.3.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax. Note that the selected images are used to sample the new contours on. Image planes need not match the original since a full 3D mesh surface is generated.

2.98.2.3.2 Default

- `"last"`

2.98.2.3.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`

- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.98.2.4 Operation

2.98.2.4.1 Description

The specific operation to perform. Available options are: 'dilate_exact_surface', 'dilate_exact_vertex', 'dilate_inexact_isotropic', 'erode_inexact_isotropic', and 'shell_inexact_isotropic'.

2.98.2.4.2 Default

- `"dilate_inexact_isotropic"`

2.98.2.4.3 Examples

- `"dilate_exact_surface"`
- `"dilate_exact_vertex"`
- `"dilate_inexact_isotropic"`
- `"erode_inexact_isotropic"`
- `"shell_inexact_isotropic"`

2.98.2.5 Distance

2.98.2.5.1 Description

For dilation and erosion operations, this parameter controls the distance the surface should travel. For shell operations, this parameter controls the resultant thickness of the shell. In all cases DICOM units are assumed.

2.98.2.5.2 Default

- `"1.0"`

2.98.2.5.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`

- "3.0"
- "5.0"

2.99 ModifyContourMetadata

2.99.1 Description

This operation injects metadata into contours.

2.99.2 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- KeyValues

2.99.2.1 NormalizedROILabelRegex

2.99.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.99.2.1.2 Default

- ".*"

2.99.2.1.3 Examples

- ".*"
- ".*Body.*"
- "Body"
- "Gross_Liver"
- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.99.2.2 ROILabelRegex

2.99.2.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.99.2.2.2 Default

- `".*"`

2.99.2.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.99.2.3 KeyValues

2.99.2.3.1 Description

Key-value pairs in the form of 'key1@value1;key2@value2' that will be injected into the selected images. Existing metadata will be overwritten. Both keys and values are case-sensitive. Note that a semi-colon separates key-value pairs, not a colon. Note that quotation marks are not stripped internally, but may have to be provided for the shell to properly interpret the argument.

2.99.2.3.2 Default

- `""`

2.99.2.3.3 Examples

- `"Description@'some description'"`
- `"'Description@some description'"`
- `"MinimumSeparation@1.23"`
- `"'Description@some description;MinimumSeparation@1.23'"`

2.100 ModifyImageMetadata

2.100.1 Description

This operation injects metadata into images. It can also modify image spatial characteristics, which are distinct from metadata.

2.100.2 Parameters

- ImageSelection
- KeyValues
- SliceThickness
- VoxelWidth
- VoxelHeight
- ImageAnchor
- ImagePosition
- ImageOrientationColumn
- ImageOrientationRow

2.100.2.1 ImageSelection

2.100.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.100.2.1.2 Default

- `"last"`

2.100.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.100.2.2 KeyValues

2.100.2.2.1 Description

Key-value pairs in the form of ‘key1@value1;key2@value2’ that will be injected into the selected images. Existing metadata will be overwritten. Both keys and values are case-sensitive. Note that a semi-colon separates key-value pairs, not a colon. Note that quotation marks are not stripped internally, but may have to be provided for the shell to properly interpret the argument. Also note that updating spatial metadata will not result in the image characteristics being altered – use the specific parameters provided to update spatial characteristics.

2.100.2.2.2 Default

- `""`

2.100.2.2.3 Examples

- `"Description@'some description'"`
- `"'Description@some description'"`
- `"MinimumSeparation@1.23"`
- `"'Description@some description;MinimumSeparation@1.23'"`

2.100.2.3 SliceThickness

2.100.2.3.1 Description

Image slices will be have this thickness (in DICOM units: mm). For most purposes, SliceThickness should be equal to SpacingBetweenSlices. If SpacingBetweenSlices is smaller than SliceThickness, images will overlap. If SpacingBetweenSlices is larger than SliceThickness, there will be a gap between images.

Updating the SliceThickness or image positioning using this operation will alter the image, but will not update SpacingBetweenSlices. This gives the user freedom to alter all image planes individually, allowing construction of non-rectilinear image volumes. If SpacingBetweenSlices is known and consistent, it should be reflected in the image metadata (by the user).

2.100.2.3.2 Default

- `"1.0"`

2.100.2.3.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.100.2.4 VoxelWidth

2.100.2.4.1 Description

Voxels will have this (in-plane) width (in DICOM units: mm). This means that row-adjacent voxels centres will be separated by VoxelWidth). Each voxel will have dimensions: VoxelWidth x VoxelHeight x SliceThickness.

2.100.2.4.2 Default

- `"1.0"`

2.100.2.4.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.100.2.5 VoxelHeight

2.100.2.5.1 Description

Voxels will have this (in-plane) height (in DICOM units: mm). This means that column-adjacent voxels centres will be separated by VoxelHeight). Each voxel will have dimensions: VoxelWidth x VoxelHeight x SliceThickness.

2.100.2.5.2 Default

- `"1.0"`

2.100.2.5.3 Examples

- `"0.1"`
- `"0.5"`
- `"1.0"`
- `"10.0"`

2.100.2.6 ImageAnchor

2.100.2.6.1 Description

A point in 3D space which denotes the origin (in DICOM units: mm). All other vectors are taken to be relative to this point. Under most circumstance the anchor should be (0,0,0). Specify coordinates separated by commas.

2.100.2.6.2 Default

- `"0.0, 0.0, 0.0"`

2.100.2.6.3 Examples

- `"0.0, 0.0, 0.0"`
- `"0.0,0.0,0.0"`
- `"1.0, -2.3, 45.6"`

2.100.2.7 ImagePosition

2.100.2.7.1 Description

The centre of the row=0, column=0 voxel in the first image (in DICOM units: mm). Specify coordinates separated by commas.

2.100.2.7.2 Default

- `"0.0, 0.0, 0.0"`

2.100.2.7.3 Examples

- `"0.0, 0.0, 0.0"`
- `"100.0,100.0,100.0"`
- `"1.0, -2.3, 45.6"`

2.100.2.8 ImageOrientationColumn

2.100.2.8.1 Description

The orientation unit vector that is aligned with image columns. Care should be taken to ensure ImageOrientationRow and ImageOrientationColumn are orthogonal. (A Gram-Schmidt orthogonalization procedure ensures they are, but the image orientation may not match the expected orientation.) Note that the magnitude will also be scaled to unit length for convenience. Specify coordinates separated by commas.

2.100.2.8.2 Default

- `"1.0, 0.0, 0.0"`

2.100.2.8.3 Examples

- `"1.0, 0.0, 0.0"`
- `"1.0, 1.0, 0.0"`
- `"0.0, 0.0, -1.0"`

2.100.2.9 ImageOrientationRow

2.100.2.9.1 Description

The orientation unit vector that is aligned with image rows. Care should be taken to ensure ImageOrientationRow and ImageOrientationColumn are orthogonal. (A Gram-Schmidt orthogonalization procedure ensures they are, but the image orientation may not match the expected orientation.) Note that the magnitude will also be scaled to unit length for convenience. Specify coordinates separated by commas.

2.100.2.9.2 Default

- `"0.0, 1.0, 0.0"`

2.100.2.9.3 Examples

- `"0.0, 1.0, 0.0"`
- `"0.0, 1.0, 1.0"`
- `"-1.0, 0.0, 0.0"`

2.101 NegatePixels

2.101.1 Description

This operation negates pixels for the selected image arrays. This functionality is often desired for processing MR images.

2.101.2 Parameters

- ImageSelection

2.101.2.1 ImageSelection

2.101.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.101.2.1.2 Default

- `"last"`

2.101.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.102 NormalizePixels

2.102.1 Description

This routine normalizes voxel intensities by adjusting them so they satisfy a 'normalization' criteria. This operation is useful as a pre-processing step when performing convolution or thresholding with absolute magnitudes.

2.102.2 Notes

- This operation considers entire image arrays, not just single images.

2.102.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Inclusivity
- ContourOverlap
- Channel
- Method

2.102.3.1 ImageSelection

2.102.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.102.3.1.2 Default

- `"last"`

2.102.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.102.3.2 NormalizedROILabelRegex

2.102.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.102.3.2.2 Default

- `".*"`

2.102.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.102.3.3 ROILabelRegex

2.102.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.102.3.3.2 Default

- `".*"`

2.102.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.102.3.4 Inclusivity

2.102.3.4.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.102.3.4.2 Default

- "center"

2.102.3.4.3 Examples

- "center"
- "centre"
- "planar_corner_inclusive"
- "planar_inc"
- "planar_corner_exclusive"
- "planar_exc"

2.102.3.5 ContourOverlap

2.102.3.5.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.102.3.5.2 Default

- "ignore"

2.102.3.5.3 Examples

- "ignore"
- "honour_opposite_orientations"
- "overlapping_contours_cancel"
- "honour_opps"
- "overlap_cancel"

2.102.3.6 Channel

2.102.3.6.1 Description

The channel to operate on (zero-based). Negative values will cause all channels to be operated on.

2.102.3.6.2 Default

- "0"

2.102.3.6.3 Examples

- "-1"
- "0"
- "1"

2.102.3.7 Method

2.102.3.7.1 Description

Controls the specific type of normalization that will be applied. ‘Stretch01’ will rescale the voxel values so the minima are 0 and the maxima are 1. Likewise, ‘stretch11’ will rescale such that the minima are -1 and the maxima are 1. Clamp will ensure all voxel intensities are within [0:1] by setting those lower than 0 to 0 and those higher than 1 to 1. (Voxels already within [0:1] will not be altered.) ‘Sum-to-zero’ will shift all voxels so that the sum of all voxel intensities is zero. (This is useful for convolution kernels.)

2.102.3.7.2 Default

- "stretch11"

2.102.3.7.3 Examples

- "clamp"
- "stretch01"
- "stretch11"
- "sum-to-zero"

2.103 OptimizeStaticBeams

2.103.1 Description

This operation takes dose matrices corresponding to single, static RT beams and attempts to optimize beam weighting to create an optimal plan subject to various criteria.

2.103.2 Notes

- This routine is a simplisitic routine that attempts to estimate the optimal beam weighting. It should NOT be used for clinical purposes, except maybe as a secondary check or a means to guess reasonable beam weights prior to optimization within the clinical TPS.
- Because beam weights are (generally) not specified in DICOM RTDOSE files, the beam weights are assumed to all be 1.0. If they are not all 1.0, the weights reported here will be relative to whatever the existing weights are.
- If no PTV ROI is available, the BODY contour may suffice. If this is not available, dose outside the body should somehow be set to zero to avoid confusing D_{max} metrics. For example, bolus D_{max} can be high, but is ultimately irrelevant.
- By default, this routine uses all available images. This may be fixed in a future release. Patches are welcome.

2.103.3 Parameters

- ImageSelection
- ResultsSummaryFileName
- UserComment
- NormalizedROILabelRegex
- ROILabelRegex
- MaxVoxelSamples
- NormalizationD
- NormalizationV
- RxDose

2.103.3.1 ImageSelection

2.103.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.103.3.1.2 Default

- `"all"`

2.103.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.103.3.2 ResultsSummaryFileName

2.103.3.2.1 Description

This file will contain a brief summary of the results. The format is CSV. Leave empty to dump to generate a unique temporary file. If an existing file is present, rows will be appended without writing a header.

2.103.3.2.2 Default

- `""`

2.103.3.2.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.103.3.3 UserComment

2.103.3.3.1 Description

A string that will be inserted into the output file which will simplify merging output with differing parameters, from different sources, or using sub-selections of the data.

2.103.3.3.2 Default

- `""`

2.103.3.3.3 Examples

- `""`
- `"Using XYZ"`
- `"Patient treatment plan C"`

2.103.3.4 NormalizedROILabelRegex

2.103.3.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.103.3.4.2 Default

- `".*"`

2.103.3.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.103.3.5 ROILabelRegex

2.103.3.5.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.103.3.5.2 Default

- `".*"`

2.103.3.5.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.103.3.6 MaxVoxelSamples

2.103.3.6.1 Description

The maximum number of voxels to randomly sample (deterministically) within the PTV. Setting lower will result in faster calculation, but lower precision. A reasonable setting depends on the size of the target structure; small targets may suffice with a few hundred voxels, but larger targets probably require several thousand.

2.103.3.6.2 Default

- `"1000"`

2.103.3.6.3 Examples

- `"200"`
- `"500"`
- `"1000"`
- `"2000"`
- `"5000"`

2.103.3.7 NormalizationD

2.103.3.7.1 Description

The isodose value that should envelop a given volume in the PTV ROI. In other words, this parameter is the ‘D’ parameter in a DVH constraint of the form $V_D \geq V_{min}$. It should be given as a fraction within [0:1] relative to the prescription dose. For example, 95% isodose should be provided as ‘0.95’.

2.103.3.7.2 Default

- `"0.95"`

2.103.3.7.3 Examples

- `"0.90"`
- `"0.95"`
- `"0.98"`
- `"0.99"`
- `"1.0"`

2.103.3.8 NormalizationV

2.103.3.8.1 Description

The minimal fractional volume of ROI that should be enclosed within one or more surfaces that demarcate the given isodose value. In other words, this parameter is the ‘Vmin’ parameter in a DVH constraint of the form $V_D \geq V_{min}$. It should be given as a fraction within [0:1] relative to the volume of the ROI (typically discretized to the number of voxels in the ROI). For example, if Vmin = 99%, provide the value ‘0.99’.

2.103.3.8.2 Default

- `"0.99"`

2.103.3.8.3 Examples

- `"0.90"`
- `"0.95"`
- `"0.98"`
- `"0.99"`
- `"1.0"`

2.103.3.9 RxDose

2.103.3.9.1 Description

The dose prescribed to the ROI that will be optimized. The units depend on the DICOM file, but will likely be Gy.

2.103.3.9.2 Default

- `"70.0"`

2.103.3.9.3 Examples

- `"48.0"`
- `"60.0"`
- `"63.3"`
- `"70.0"`
- `"100.0"`

2.104 OrderImages

2.104.1 Description

This operation will order individual image slices within collections (Image_Arrays) based on the values of the specified metadata tags.

2.104.2 Notes

- Images are moved, not copied.
- Image groupings are retained, and the order of groupings is not altered.
- Images that do not contain the specified metadata will be sorted after the end.

2.104.3 Parameters

- ImageSelection
- Key

2.104.3.1 ImageSelection

2.104.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.104.3.1.2 Default

- `"all"`

2.104.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.104.3.2 Key

2.104.3.2.1 Description

Image metadata key to use for ordering. Images will be sorted according to the key's value 'natural' sorting order, which compares sub-strings of numbers and characters separately. Note this ordering is expected to be stable, but may not always be on some systems.

2.104.3.2.2 Default

- `""`

2.104.3.2.3 Examples

- `"AcquisitionTime"`
- `"ContentTime"`
- `"SeriesNumber"`
- `"SeriesDescription"`

2.105 PlotPerROITimeCourses

2.105.1 Description

Interactively plot time courses for the specified ROI(s).

2.105.2 Parameters

- ROILabelRegex

2.105.2.1 ROILabelRegex

2.105.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.105.2.1.2 Default

- `".*"`

2.105.2.1.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.106 PreFilterEnormousCTValues

2.106.1 Description

This operation runs the data through a per-pixel filter, censoring pixels which are too high to legitimately show up in a clinical CT. Censored pixels are set to NaN. Data is modified and no copy is made!

2.106.2 Parameters

No registered options.

2.107 PresentationImage

2.107.1 Description

This operation renders an image with any contours in-place and colour mapping using an SFML backend.

2.107.2 Notes

- By default this operation displays the last available image. This makes it easier to produce a sequence of images by inserting this operation into a sequence of operations.

2.107.3 Parameters

- ScaleFactor
- ImageFileName
- ColourMapRegex
- WindowLow
- WindowHigh

2.107.3.1 ScaleFactor

2.107.3.1.1 Description

This factor is applied to the image width and height to magnify (larger than 1) or shrink (less than 1) the image. This factor only affects the output image size. Note that aspect ratio is retained, but rounding for non-integer factors may lead to small (1-2 pixel) discrepancies.

2.107.3.1.2 Default

- `"1.0"`

2.107.3.1.3 Examples

- `"0.5"`
- `"1.0"`
- `"2.0"`
- `"5.23"`

2.107.3.2 ImageFileName

2.107.3.2.1 Description

The file name to use for the image. If blank, a filename will be generated sequentially.

2.107.3.2.2 Default

- `""`

2.107.3.2.3 Examples

- `""`
- `"/tmp/an_image.png"`
- `"afile.png"`

2.107.3.3 ColourMapRegex

2.107.3.3.1 Description

The colour mapping to apply to the image if there is a single channel. The default will match the first available, and if there is no matching map found, the first available will be selected.

2.107.3.3.2 Default

- `".*"`

2.107.3.3.3 Examples

- `"Viridis"`
- `"Magma"`
- `"Plasma"`
- `"Inferno"`
- `"Jet"`
- `"MorelandBlueRed"`
- `"MorelandBlackBody"`
- `"MorelandExtendedBlackBody"`
- `"KRC"`
- `"ExtendedKRC"`
- `"Kovesi_LinKRYW_5-100_c64"`
- `"Kovesi_LinKRYW_0-100_c71"`
- `"Kovesi_Cyclic_cet-c2"`
- `"LANLOliveGreentoBlue"`
- `"YgorIncandescent"`
- `"LinearRamp"`

2.107.3.4 WindowLow

2.107.3.4.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or lower will be assigned the lowest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.107.3.4.2 Default

- `""`

2.107.3.4.3 Examples

- `""`
- `"-1.23"`
- `"0"`
- `"1E4"`

2.107.3.5 WindowHigh

2.107.3.5.1 Description

If provided, this parameter will override any existing window and level. All pixels with the intensity value or higher will be assigned the highest possible colour according to the colour map. Not providing a valid number will disable window overrides.

2.107.3.5.2 Default

- `""`

2.107.3.5.3 Examples

- `""`
- `"1.23"`
- `"0"`
- `"10.3E4"`

2.108 PruneEmptyImageDoseArrays

2.108.1 Description

This operation deletes Image_Arrays that do not contain any images.

2.108.2 Parameters

No registered options.

2.109 PurgeContours

2.109.1 Description

This routine purges contours if they satisfy various criteria.

2.109.2 Notes

- This operation considers individual contours only at the moment. It could be extended to operate on whole ROIs (i.e., contour_collections), or to perform a separate vote within each ROI. The individual contour approach was taken since filtering out small contour 'islands' is the primary use-case.

2.109.3 Parameters

- ROILabelRegex
- NormalizedROILabelRegex
- Invert
- AreaAbove
- AreaBelow
- PerimeterAbove
- PerimeterBelow

2.109.3.1 ROILabelRegex

2.109.3.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.109.3.1.2 Default

- `".*"`

2.109.3.1.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.109.3.2 NormalizedROILabelRegex

2.109.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.109.3.2.2 Default

- `".*"`

2.109.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.109.3.3 Invert

2.109.3.3.1 Description

If false, matching contours will be purged. If true, matching contours will be retained and non-matching contours will be purged. Enabling this option is equivalent to a 'Keep if and only if' operation.

2.109.3.3.2 Default

- `"false"`

2.109.3.3.3 Examples

- `"true"`
- `"false"`

2.109.3.4 AreaAbove

2.109.3.4.1 Description

If this option is provided with a valid positive number, contour(s) with an area greater than the specified value are purged. Note that the DICOM coordinate space is used. (Supplying the default, inf, will disable this option.)

2.109.3.4.2 Default

- `"inf"`

2.109.3.4.3 Examples

- "inf"
- "100.0"
- "1000"
- "10.23E8"

2.109.3.5 AreaBelow

2.109.3.5.1 Description

If this option is provided with a valid positive number, contour(s) with an area less than the specified value are purged. Note that the DICOM coordinate space is used. (Supplying the default, -inf, will disable this option.)

2.109.3.5.2 Default

- "-inf"

2.109.3.5.3 Examples

- "-inf"
- "100.0"
- "1000"
- "10.23E8"

2.109.3.6 PerimeterAbove

2.109.3.6.1 Description

If this option is provided with a valid positive number, contour(s) with a perimeter greater than the specified value are purged. Note that the DICOM coordinate space is used. (Supplying the default, inf, will disable this option.)

2.109.3.6.2 Default

- "inf"

2.109.3.6.3 Examples

- "inf"
- "10.0"
- "100"
- "10.23E4"

2.109.3.7 PerimeterBelow

2.109.3.7.1 Description

If this option is provided with a valid positive number, contour(s) with a perimeter less than the specified value are purged. Note that the DICOM coordinate space is used. (Supplying the default, -inf, will disable this option.)

2.109.3.7.2 Default

- `"-inf"`

2.109.3.7.3 Examples

- `"-inf"`
- `"10.0"`
- `"100"`
- `"10.23E4"`

2.110 RankPixels

2.110.1 Description

This operation ranks pixels throughout an image array.

2.110.2 Notes

- This routine operates on all images in an image array, so pixel value ranks are valid throughout the array. However, the window and level of each window is separately determined. You will need to set a uniform window and level manually if desired.

- This routine operates on all images in an image array. If images need to be processed individually, arrays will have to be exploded prior to calling this routine. Note that if this functionality is required, it can be implemented as an operation option easily. Likewise, if multiple image arrays must be considered simultaneously, they will need to be combined before invoking this operation.

2.110.3 Parameters

- ImageSelection
- Method

- LowerThreshold
- UpperThreshold

2.110.3.1 ImageSelection

2.110.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.110.3.1.2 Default

- `"last"`

2.110.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.110.3.2 Method

2.110.3.2.1 Description

Pixels participating in the ranking process will have their pixel values replaced. They can be replaced with either a rank or the corresponding percentile. Ranks start at zero and percentiles are centre-weighted (i.e., rank-averaged).

2.110.3.2.2 Default

- `"Percentile"`

2.110.3.2.3 Examples

- `"Rank"`
- `"Percentile"`

2.110.3.3 LowerThreshold

2.110.3.3.1 Description

The (inclusive) threshold above which pixel values must be in order to participate in the rank.

2.110.3.3.2 Default

- `"-inf"`

2.110.3.3.3 Examples

- `"-inf"`
- `"0.0"`
- `"-900"`

2.110.3.4 UpperThreshold

2.110.3.4.1 Description

The (inclusive) threshold below which pixel values must be in order to participate in the rank.

2.110.3.4.2 Default

- `"inf"`

2.110.3.4.3 Examples

- `"inf"`
- `"0.0"`
- `"1500"`

2.111 ReduceNeighbourhood

2.111.1 Description

This routine walks the voxels of a 3D rectilinear image collection, reducing the distribution of voxels within the local volumetric neighbourhood to a scalar value, and updating the voxel value with this scalar. This routine can be used to implement mean and median filters (amongst others) that operate over a variety of 3D neighbourhoods. Besides purely statistical reductions, logical reductions can be applied.

2.111.2 Notes

- The provided image collection must be rectilinear.
- This operation can be used to compute core 3D morphology operations (erosion and dilation) as well as composite operations like opening (i.e., erosion followed by dilation), closing (i.e., dilation followed by erosion), 'gradient' (i.e., the difference between dilation and erosion, which produces an outline), and various other combinations of core and composite operations.

2.111.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Neighbourhood
- Reduction
- MaxDistance

2.111.3.1 ImageSelection

2.111.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or ‘4D’ time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be ‘first’, ‘last’, ‘none’, or ‘all’ literals. Additionally ‘#N’ for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, ‘#-N’ selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a ‘!’. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a ‘!’). Multiple criteria can be specified by separating them with a ‘;’ and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.111.3.1.2 Default

- `"last"`

2.111.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.111.3.2 NormalizedROILabelRegex

2.111.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.111.3.2.2 Default

- `".*"`

2.111.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.111.3.3 ROILabelRegex

2.111.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.111.3.3.2 Default

- `".*"`

2.111.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.111.3.4 Channel

2.111.3.4.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.111.3.4.2 Default

- `"0"`

2.111.3.4.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.111.3.5 Neighbourhood

2.111.3.5.1 Description

Controls how the neighbourhood surrounding a voxel is defined. Variable-size neighbourhoods ‘spherical’ and ‘cubic’ are defined. An appropriate isotropic extent must be provided for these neighbourhoods. (See below; extents must be provided in DICOM units, i.e., mm.) Fixed-size neighbourhoods specify a fixed number of adjacent voxels. Fixed rectagular neighbourhoods are specified like ‘RxCxI’ for row, column, and image slice extents (as integer number of rows, columns, and slices). Fixed spherical neighbourhoods are specified like ‘Wsphere’ where W is the width (i.e., the number of voxels wide). In morphological terminology, the neighbourhood is referred to as a ‘structuring element.’ A similar concept is the convolutional ‘kernel.’

2.111.3.5.2 Default

- `"spherical"`

2.111.3.5.3 Examples

- `"spherical"`
- `"cubic"`
- `"3x3x3"`
- `"5x5x5"`
- `"3sphere"`
- `"5sphere"`
- `"7sphere"`
- `"9sphere"`
- `"11sphere"`
- `"13sphere"`
- `"15sphere"`

2.111.3.6 Reduction

2.111.3.6.1 Description

Controls how the distribution of voxel values from neighbouring voxels is reduced. Statistical distribution reducers ‘min’, ‘mean’, ‘median’, and ‘max’ are defined. ‘Min’ is also known as the ‘erosion’ operation. Likewise, ‘max’ is also known as the ‘dilation’ operation. Note that the morphological ‘opening’ operation can be accomplished by sequentially performing an erosion and then a dilation using the same neighbourhood. Logical reducers ‘is_min’ and ‘is_max’ are also available – is_min (is_max) replace the voxel value with 1.0 if it was the min (max) in the neighbourhood and 0.0 otherwise. Logical reducers ‘is_min_nan’ and ‘is_max_nan’ are variants that replace the voxel with a NaN instead of 1.0 and otherwise do not overwrite the original voxel value.

2.111.3.6.2 Default

- `"median"`

2.111.3.6.3 Examples

- `"min"`
- `"erode"`
- `"mean"`
- `"median"`
- `"max"`
- `"dilate"`
- `"is_min"`
- `"is_max"`
- `"is_min_nan"`
- `"is_max_nan"`

2.111.3.7 MaxDistance

2.111.3.7.1 Description

The maximum distance (inclusive, in DICOM units: mm) within which neighbouring voxels will be evaluated for variable-size neighbourhoods. Note that this parameter will be ignored if a fixed-size neighbourhood has been specified. For spherical neighbourhoods, this distance refers to the radius. For cubic neighbourhoods, this distance refers to ‘box radius’ or the distance from the cube centre to the nearest point on each bounding face. Voxels separated by more than this distance will not be evaluated together.

2.111.3.7.2 Default

- `"2.0"`

2.111.3.7.3 Examples

- `"0.5"`
- `"2.0"`
- `"15.0"`

2.112 RemeshSurfaceMeshes

2.112.1 Description

This operation re-meshes existing surface meshes according to the specified criteria, replacing the original meshes with remeshed copies.

2.112.2 Notes

- Selected surface meshes should represent polyhedra.

2.112.3 Parameters

- MeshSelection
- Iterations
- TargetEdgeLength

2.112.3.1 MeshSelection

2.112.3.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.112.3.1.2 Default

- `"last"`

2.112.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.112.3.2 Iterations

2.112.3.2.1 Description

The number of remeshing iterations to perform.

2.112.3.2.2 Default

- `"5"`

2.112.3.2.3 Examples

- `"1"`
- `"3"`
- `"5"`
- `"10"`

2.112.3.3 TargetEdgeLength

2.112.3.3.1 Description

The desired length of all edges in the remeshed mesh in DICOM units (mm).

2.112.3.3.2 Default

- `"1.5"`

2.112.3.3.3 Examples

- "0.2"
- "0.75"
- "1.0"
- "1.5"
- "2.015"

2.113 SFML_Viewer

2.113.1 Description

Launch an interactive viewer based on SFML. Using this viewer, it is possible to contour ROIs, generate plots of pixel intensity along profiles or through time, inspect and compare metadata, and various other things.

2.113.2 Parameters

- SingleScreenshot
- SingleScreenshotFileName

2.113.2.1 SingleScreenshot

2.113.2.1.1 Description

If 'true', a single screenshot is taken and then the viewer is exited. This option works best for quick visual inspections, and should not be used for later processing or analysis.

2.113.2.1.2 Default

- "false"

2.113.2.1.3 Examples

- "true"
- "false"

2.113.2.2 SingleScreenshotFileName

2.113.2.2.1 Description

Iff invoking the ‘SingleScreenshot’ argument, use this string as the screenshot filename. If blank, a filename will be generated sequentially.

2.113.2.2.2 Default

- `""`

2.113.2.2.3 Examples

- `""`
- `"/tmp/a_screenshot.png"`
- `"afile.png"`

2.114 ScalePixels

2.114.1 Description

This operation scales pixel (voxel) values confined to one or more ROIs.

2.114.2 Notes

- This routine could be used to derive, for example, per-fraction dose from a total dose image array.

2.114.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Inclusivity
- ContourOverlap
- ScaleFactor
- Channel

2.114.3.1 ImageSelection

2.114.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.114.3.1.2 Default

- `"last"`

2.114.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.114.3.2 NormalizedROILabelRegex

2.114.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.114.3.2.2 Default

- ".*"

2.114.3.2.3 Examples

- ".*"
- ".*Body.*"
- "Body"
- "Gross_Liver"
- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.114.3.3 ROILabelRegex

2.114.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.114.3.3.2 Default

- ".*"

2.114.3.3.3 Examples

- ".*"
- ".*body.*"
- "body"
- "Gross_Liver"
- ".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"
- "left_parotid|right_parotid"

2.114.3.4 Inclusivity

2.114.3.4.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if

ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.114.3.4.2 Default

- `"center"`

2.114.3.4.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.114.3.5 ContourOverlap

2.114.3.5.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.114.3.5.2 Default

- `"ignore"`

2.114.3.5.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.114.3.6 ScaleFactor

2.114.3.6.1 Description

The numeric factor to multiply all pixel (voxel) values with.

2.114.3.6.2 Default

- `"1.0"`

2.114.3.6.3 Examples

- `"-1.0"`
- `"0.0"`
- `"1.23E-5"`

2.114.3.7 Channel

2.114.3.7.1 Description

The image channel to use. Zero-based.

2.114.3.7.2 Default

- `"0"`

2.114.3.7.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.115 SeamContours

2.115.1 Description

This routine converts contours that represent 'outer' and 'inner' via contour orientation into contours that are uniformly outer but have a zero-area seam connecting the inner and outer portions.

2.115.2 Notes

- This routine currently operates on all available ROIs.

- This routine operates on one contour_collection at a time. It will combine contours that are in the same contour_collection and overlap, even if they have different ROINames. Consider making a complementary routine that partitions contours into ROIs based on ROIName (or other metadata) if more rigorous enforcement is needed.
- This routine actually computes the XOR Boolean of contours that overlap. So if contours partially overlap, this routine will treat the overlapping parts as if they are holes, and the non-overlapping parts as if they represent the ROI. This behaviour may be surprising in some cases.
- This routine will also treat overlapping contours with like orientation as if the smaller contour were a hole of the larger contour.
- This routine will ignore contour orientation if there is only a single contour. More specifically, for a given ROI label, planes with a single contour will be unaltered.
- Only the common metadata between outer and inner contours is propagated to the seamed contours.
- This routine will NOT combine disconnected contours with a seam. Disconnected contours will remain disconnected.

2.115.3 Parameters

No registered options.

2.116 SelectSlicesIntersectingROI

2.116.1 Description

This operation applies a whitelist to the most-recently loaded images. Images must 'slice' through one of the described ROIs in order to make the whitelist. This operation is typically used to reduce long computations by trimming the field of view of extraneous image slices.

2.116.2 Parameters

- NormalizedROILabelRegex
- ROILabelRegex

2.116.2.1 NormalizedROILabelRegex

2.116.2.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.116.2.1.2 Default

- `".*"`

2.116.2.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.116.2.2 ROILabelRegex

2.116.2.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.116.2.2.2 Default

- `".*"`

2.116.2.2.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.117 SimplifyContours

2.117.1 Description

This operation performs simplification on contours by removing or moving vertices. This operation is mostly used to reduce the computational complexity of other operations.

2.117.2 Notes

- Contours are currently processed individually, not as a volume.
- Simplification is generally performed most eagerly on regions with relatively low curvature. Regions of high curvature are generally simplified only as necessary.

2.117.3 Parameters

- NormalizedROILabelRegex
- ROILabelRegex
- FractionalAreaTolerance
- SimplificationMethod

2.117.3.1 NormalizedROILabelRegex

2.117.3.1.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.117.3.1.2 Default

- `".*"`

2.117.3.1.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`

- ".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"
- "Left Parotid|Right Parotid"

2.117.3.2 ROILabelRegex

2.117.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.117.3.2.2 Default

- ".*"

2.117.3.2.3 Examples

- ".*"
- ".*body.*"
- "body"
- "Gross_Liver"
- ".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"
- "left_parotid|right_parotid"

2.117.3.3 FractionalAreaTolerance

2.117.3.3.1 Description

The fraction of area each contour will tolerate during simplified. This is a measure of how much the contour area can change due to simplification.

2.117.3.3.2 Default

- "0.01"

2.117.3.3.3 Examples

- "0.001"
- "0.01"
- "0.02"
- "0.05"
- "0.10"

2.117.3.4 SimplificationMethod

2.117.3.4.1 Description

The specific algorithm used to perform contour simplification. 'Vertex removal' is a simple algorithm that removes vertices one-by-one without replacement. It iteratively ranks vertices and removes the single vertex that has the least impact on contour area. It is best suited to removing redundant vertices or whenever new vertices should not be added. 'Vertex collapse' combines two adjacent vertices into a single vertex at their midpoint. It iteratively ranks vertex pairs and removes the single vertex that has the least total impact on contour area. Note that small sharp features that alternate inward and outward will have a small total area cost, so will be pruned early. Thus this technique acts as a low-pass filter and will defer simplification of high-curvature regions until necessary. It is more economical compared to vertex removal in that it will usually simplify contours more for a given tolerance (or, equivalently, can retain contour fidelity better than vertex removal for the same number of vertices). However, vertex collapse performs an averaging that may result in numerical imprecision.

2.117.3.4.2 Default

- `"vert-collapse"`

2.117.3.4.3 Examples

- `"vertex-collapse"`
- `"vertex-removal"`

2.118 SimplifySurfaceMeshes

2.118.1 Description

This operation performs mesh simplification on existing surface meshes according to the specified criteria, replacing the original meshes with simplified copies.

2.118.2 Notes

- Selected surface meshes should represent polyhedra.

2.118.3 Parameters

- MeshSelection
- EdgeCountLimit

2.118.3.1 MeshSelection

2.118.3.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.118.3.1.2 Default

- `"last"`

2.118.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.118.3.2 EdgeCountLimit

2.118.3.2.1 Description

The maximum number of edges simplified meshes should contain.

2.118.3.2.2 Default

- `"250000"`

2.118.3.2.3 Examples

- `"20000"`
- `"100000"`
- `"500000"`
- `"5000000"`

2.119 SimulateRadiograph

2.119.1 Description

This routine uses ray marching an volumteric sampling to simulate radiographs from a CT image array. Voxels are assumed to have intensities in HU. A simplisitic conversion from CT number (in HU) to relative electron density (see note below) is performed for marched rays.

2.119.2 Notes

- Images must be rectilinear.
- This operation currently takes a simplistic approach and should only be used for purposes where the simulated radiograph contrast can be tuned and validated (e.g., in a relative way).
- This operation assumes mass density (in g/cm^3) and relative electron density (dimensionless; relative to electron density of water, which is $3.343E23$ cm^3) are numerically equivalent. This assumption appears to be reasonable for bulk human tissue (arXiv:1508.00226v1).

2.119.3 Parameters

- ImageSelection
- Filename
- MarchingDistance
- SourcePosition
- AttenuationScale

- Rows
- Columns

2.119.3.1 ImageSelection

2.119.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.119.3.1.2 Default

- `"last"`

2.119.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.119.3.2 Filename

2.119.3.2.1 Description

The filename (or full path) to which the simulated image will be saved to. The format is FITS. Leaving empty will result in a unique name being generated.

2.119.3.2.2 Default

- `""`

2.119.3.2.3 Examples

- `""`
- `"./img.fits"`
- `"sim_radiograph.fits"`
- `"/tmp/out.fits"`

2.119.3.3 MarchingDistance

2.119.3.3.1 Description

The distance (in DICOM units; mm) that rays will incrementally be marched at each iteration. This value should be on the order of the smallest image voxel size to give the best image quality. Conversely, if a course radiograph is needed then larger values can be used. Trilinear interpolation is used to sample the CT number at arbitrary points in 3D. Note that the CT numbers between sample points are ignored, so tissue heterogeneities and features smaller than the marching distance are not likely to show up in the image.

2.119.3.3.2 Default

- `"0.5"`

2.119.3.3.3 Examples

- `"0.25"`
- `"0.5"`
- `"1.0"`

2.119.3.4 SourcePosition

2.119.3.4.1 Description

This parameter controls where the virtual point source is. Both absolute and relative positioning are available. A source located at point (1.0, -2.3, 4.5) in the DICOM coordinate system of a given image can be specified as 'absolute(1.0, -2.3, 4.5)'. A source located relative to the image centre by offset (10.0, -23.4, 45.6) in the DICOM coordinate system of a given image can be specified as 'relative(10.0, -23.4, 45.6)'. Relative offsets must be specified relative to the image centre. Note that DICOM units (i.e., mm) are used for all coordinates.

2.119.3.4.2 Default

- `"relative(0.0, 1000.0, 20.0)"`

2.119.3.4.3 Examples

- `"relative(0.0, 1610.0, 20.0)"`
- `"absolute(-123.0, 123.0, 1.23)"`

2.119.3.5 AttenuationScale

2.119.3.5.1 Description

This parameter globally scales all attenuation factors derived via ray marching. Adjusting this parameter will alter the radiograph image contrast; numbers within (0:1) will result in less attenuation, whereas numbers within (1:inf] will result in more attenuation. Thin or low-mass subjects might require artifically increased attenuation, whereas thick or high-mass subjects might require artifically decreased attenuation. Setting this number to 1 will result in no scaling.

2.119.3.5.2 Default

- `"1.0"`

2.119.3.5.3 Examples

- `"0.01"`
- `"0.1"`
- `"1.0"`
- `"10.0"`
- `"1E2"`

2.119.3.6 Rows

2.119.3.6.1 Description

The number of rows that the simulated radiograph will contain. Note that the field of view is determined separately from the number of rows and columns, so increasing the row count will only result in increased spatial resolution.

2.119.3.6.2 Default

- `"512"`

2.119.3.6.3 Examples

- `"100"`
- `"500"`
- `"2000"`

2.119.3.7 Columns

2.119.3.7.1 Description

The number of columns that the simulated radiograph will contain. Note that the field of view is determined separately from the number of rows and columns, so increasing the column count will only result in increased spatial resolution.

2.119.3.7.2 Default

- `"512"`

2.119.3.7.3 Examples

- `"100"`
- `"500"`
- `"2000"`

2.120 SpatialBlur

2.120.1 Description

This operation blurs pixels (within the plane of the image only) using the specified estimator.

2.120.2 Parameters

- ImageSelection
- Estimator
- GaussianOpenSigma

2.120.2.1 ImageSelection

2.120.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.120.2.1.2 Default

- `"all"`

2.120.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.120.2.2 Estimator

2.120.2.2.1 Description

Controls the (in-plane) blur estimator to use. Options are currently: box_3x3, box_5x5, gaussian_3x3, gaussian_5x5, and gaussian_open. The latter (gaussian_open) is adaptive and requires a supplementary parameter that controls the number of adjacent pixels to consider. The former ('. . . 3x3' and '. . . 5x5') are 'fixed' estimators that use a convolution kernel with a fixed size (3x3 or 5x5 pixel neighbourhoods). All estimators operate in 'pixel-space' and are ignorant about the image spatial extent. All estimators are normalized, and thus won't significantly affect the pixel magnitude scale.

2.120.2.2.2 Default

- `"gaussian_open"`

2.120.2.2.3 Examples

- `"box_3x3"`
- `"box_5x5"`
- `"gaussian_3x3"`
- `"gaussian_5x5"`
- `"gaussian_open"`

2.120.2.3 GaussianOpenSigma

2.120.2.3.1 Description

Controls the number of neighbours to consider (only) when using the gaussian_open estimator. The number of pixels is computed automatically to accommodate the specified sigma (currently ignored pixels have 3*sigma or less weighting). Be aware this operation can take an enormous amount of time, since the pixel neighbourhoods quickly grow large.

2.120.2.3.2 Default

- `"1.5"`

2.120.2.3.3 Examples

- `"0.5"`
- `"1.0"`
- `"1.5"`
- `"2.5"`
- `"5.0"`

2.121 SpatialDerivative

2.121.1 Description

This operation estimates various partial derivatives (of pixel values) within 2D images.

2.121.2 Parameters

- ImageSelection
- Estimator
- Method

2.121.2.1 ImageSelection

2.121.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.121.2.1.2 Default

- `"last"`

2.121.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`

- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.121.2.2 Estimator

2.121.2.2.1 Description

Controls the finite-difference partial derivative order or estimator used. All estimators are centred and use mirror boundary conditions. First-order estimators include the basic nearest-neighbour first derivative, and Roberts' cross, Prewitt, Sobel, Scharr estimators. 'XxY' denotes the size of the convolution kernel (i.e., the number of adjacent pixels considered). The only second-order estimator is the basic nearest-neighbour second derivative.

2.121.2.2.2 Default

- `"Scharr-3x3"`

2.121.2.2.3 Examples

- `"first"`
- `"Roberts-cross-3x3"`
- `"Prewitt-3x3"`
- `"Sobel-3x3"`
- `"Sobel-5x5"`
- `"Scharr-3x3"`
- `"Scharr-5x5"`
- `"second"`

2.121.2.3 Method

2.121.2.3.1 Description

Controls partial derivative method. First-order derivatives can be row- or column-aligned, Roberts' cross can be (+row,+col)-aligned or (-row,+col)-aligned. Second-order derivatives can be row-aligned, column-aligned, or 'cross' –meaning the compound partial derivative. All methods support non-maximum-suppression for edge thinning, but currently only the magnitude is output. All methods support magnitude (addition of orthogonal components in quadrature) and orientation (in radians; [0,2pi)).

2.121.2.3.2 Default

- `"magnitude"`

2.121.2.3.3 Examples

- `"row-aligned"`
- `"column-aligned"`
- `"prow-pcol-aligned"`
- `"nrow-pcol-aligned"`
- `"magnitude"`
- `"orientation"`
- `"non-maximum-suppression"`
- `"cross"`

2.122 SpatialSharpen

2.122.1 Description

This operation 'sharpens' pixels (within the plane of the image only) using the specified estimator.

2.122.2 Parameters

- ImageSelection
- Estimator

2.122.2.1 ImageSelection

2.122.2.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order

specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.122.2.1.2 Default

- `"all"`

2.122.2.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.122.2.2 Estimator

2.122.2.2.1 Description

Controls the (in-plane) sharpening estimator to use. Options are currently: sharpen_3x3 and unsharp_mask_5x5. The latter is based on a 5x5 Gaussian blur estimator.

2.122.2.2.2 Default

- `"unsharp_mask_5x5"`

2.122.2.2.3 Examples

- `"sharpen_3x3"`
- `"unsharp_mask_5x5"`

2.123 SubdivideSurfaceMeshes

2.123.1 Description

This operation subdivides existing surface meshes according to the specified criteria, replacing the original meshes with subdivided copies.

2.123.2 Notes

- Selected surface meshes should represent polyhedra.

2.123.3 Parameters

- MeshSelection
- Iterations

2.123.3.1 MeshSelection

2.123.3.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.123.3.1.2 Default

- `"last"`

2.123.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`

- `"key1@.*value1.*;key2@^value2$;first"`

2.123.3.2 Iterations

2.123.3.2.1 Description

The number of times subdivision should be performed.

2.123.3.2.2 Default

- `"2"`

2.123.3.2.3 Examples

- `"1"`
- `"2"`
- `"5"`

2.124 Subsegment_ComputeDose_VanLuijk

2.124.1 Description

This operation sub-segments the selected ROI(s) and computes dose within the resulting sub-segments.

2.124.2 Parameters

- AreaDataFileName
- DerivativeDataFileName
- DistributionDataFileName
- NormalizedROILabelRegex
- PlanarOrientation
- ReplaceAllWithSubsegment
- RetainSubsegment
- ROILabelRegex
- SubsegMethod
- XSelection
- YSelection
- ZSelection
- FractionalTolerance
- MaxBisects

2.124.2.1 AreaDataFileName

2.124.2.1.1 Description

A filename (or full path) in which to append sub-segment areaa data generated by this routine. The format is CSV. Note that if a sub-segment has zero area or does not exist, no area will be printed. You'll have to manually add sub-segments with zero area as needed if this info is relevant to you (e.g., if you are deriving a population average). Leave empty to NOT dump anything.

2.124.2.1.2 Default

- `""`

2.124.2.1.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"area_data.csv"`

2.124.2.2 DerivativeDataFileName

2.124.2.2.1 Description

A filename (or full path) in which to append derivative data generated by this routine. The format is CSV. Leave empty to dump to generate a unique temporary file.

2.124.2.2.2 Default

- `""`

2.124.2.2.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"derivative_data.csv"`

2.124.2.3 DistributionDataFileName

2.124.2.3.1 Description

A filename (or full path) in which to append raw distribution data generated by this routine. The format is one line of description followed by one line for the distribution; pixel intensities are listed with a single space between elements; the descriptions contain the patient ID, ROIName, and subsegment description (guaranteed) and possibly various other data afterward. Leave empty to NOT dump anything.

2.124.2.3.2 Default

- `""`

2.124.2.3.3 Examples

- `""`
- `"/tmp/somefile"`
- `"localfile.csv"`
- `"distributions.data"`

2.124.2.4 NormalizedROILabelRegex

2.124.2.4.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.124.2.4.2 Default

- `".*"`

2.124.2.4.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.124.2.5 PlanarOrientation

2.124.2.5.1 Description

A string instructing how to orient the cleaving planes. Currently only 'AxisAligned' (i.e., align with the image/dose grid row and column unit vectors) and 'StaticOblique' (i.e., same as AxisAligned but rotated 22.5 degrees to reduce colinearity, which sometimes improves sub-segment area consistency).

2.124.2.5.2 Default

- `"AxisAligned"`

2.124.2.5.3 Examples

- `"AxisAligned"`
- `"StaticOblique"`

2.124.2.6 ReplaceAllWithSubsegment

2.124.2.6.1 Description

Keep the sub-segment and remove any existing contours from the original ROIs. This is most useful for further processing, such as nested sub-segmentation. Note that sub-segment contours currently have identical metadata to their parent contours.

2.124.2.6.2 Default

- `"false"`

2.124.2.6.3 Examples

- `"true"`
- `"false"`

2.124.2.7 RetainSubsegment

2.124.2.7.1 Description

Keep the sub-segment as part of the original ROIs. The contours are appended to the original ROIs, but the contour ROIName and NormalizedROIName are set to the argument provided. (If no argument is provided, sub-segments are not retained.) This is most useful for inspection of sub-segments. Note that sub-segment contours currently have identical metadata to their parent contours, except they are renamed accordingly.

2.124.2.7.2 Default

- `""`

2.124.2.7.3 Examples

- `"subsegment_01"`
- `"subsegment_02"`
- `"selected_subsegment"`

2.124.2.8 ROILabelRegex

2.124.2.8.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.124.2.8.2 Default

- `".*"`

2.124.2.8.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.124.2.9 SubsegMethod

2.124.2.9.1 Description

The method to use for sub-segmentation. Nested sub-segmentation should almost always be preferred unless you know what you're doing. It should be faster too. The compound method was used in the van Luijk paper, but it is known to have serious problems.

2.124.2.9.2 Default

- `"nested"`

2.124.2.9.3 Examples

- `"nested"`
- `"compound"`

2.124.2.10 XSelection

2.124.2.10.1 Description

(See ZSelection description.) The “X” direction is defined in terms of movement on an image when the row number increases. This is generally VERTICAL and DOWNWARD. All selections are defined in terms of the original ROIs.

2.124.2.10.2 Default

- `"1.0;0.0"`

2.124.2.10.3 Examples

- `"0.50;0.50"`
- `"0.50;0.0"`
- `"0.30;0.0"`
- `"0.30;0.70"`

2.124.2.11 YSelection

2.124.2.11.1 Description

(See ZSelection description.) The “Y” direction is defined in terms of movement on an image when the column number increases. This is generally HORIZONTAL and RIGHTWARD. All selections are defined in terms of the original ROIs.

2.124.2.11.2 Default

- `"1.0;0.0"`

2.124.2.11.3 Examples

- `"0.50;0.50"`
- `"0.50;0.0"`
- `"0.30;0.0"`
- `"0.30;0.70"`

2.124.2.12 ZSelection

2.124.2.12.1 Description

The thickness and offset defining the single, continuous extent of the sub-segmentation in terms of the fractional area remaining above a plane. The planes define the portion extracted and are determined such that sub-segmentation will give the desired fractional planar areas. The numbers specify the thickness and offset from the bottom of the ROI volume to the bottom of the extent. The 'upper' direction is take from the contour plane orientation and assumed to be positive if pointing toward the positive-z direction. Only a single 3D selection can be made per operation invocation. Sub-segmentation can be performed in transverse ("Z"), row_unit ("X"), and column_unit ("Y") directions (in that order). All selections are defined in terms of the original ROIs. Note that it is possible to perform nested sub-segmentation (including passing along the original contours) by opting to replace the original ROI contours with this sub-segmentation and invoking this operation again with the desired sub-segmentation. If you want the middle 50% of an ROI, specify '0.50;0.25'. If you want the upper 50% then specify '0.50;0.50'. If you want the lower 50% then specify '0.50;0.0'. If you want the upper 30% then specify '0.30;0.70'. If you want the lower 30% then specify '0.30;0.70'.

2.124.2.12.2 Default

- `"1.0;0.0"`

2.124.2.12.3 Examples

- `"0.50;0.50"`
- `"0.50;0.0"`
- `"0.30;0.0"`
- `"0.30;0.70"`

2.124.2.13 FractionalTolerance

2.124.2.13.1 Description

The tolerance of X, Y, and Z fractional area bisection criteria (see ZSelection description). This parameter specifies a stopping condition for the bisection procedure. If it is set too high, sub-segments may be inadequatly rough. If it is set too low, bisection below the machine precision floor may be attempted, which will result in instabilities. Note that the number of permitted iterations will control whether this tolerance can possibly be reached; if strict adherence is required, set the maximum number of iterations to be excessively large.

2.124.2.13.2 Default

- `"0.001"`

2.124.2.13.3 Examples

- `"1E-2"`
- `"1E-3"`
- `"1E-4"`
- `"1E-5"`

2.124.2.14 MaxBisects

2.124.2.14.1 Description

The maximum number of iterations the bisection procedure can perform. This parameter specifies a stopping condition for the bisection procedure. If it is set too low, sub-segments may be inadequatly rough. If it is set too high, bisection below the machine precision floor may be attempted, which will result in instabilities. Note that the fractional tolerance will control whether this tolerance can possibly be reached; if an exact number of iterations is required, set the fractional tolerance to be excessively small.

2.124.2.14.2 Default

- `"20"`

2.124.2.14.3 Examples

- `"10"`
- `"20"`
- `"30"`

2.125 SubtractImages

2.125.1 Description

This routine subtracts images that spatially overlap.

2.125.2 Notes

- The ReferenceImageSelection is subtracted from the ImageSelection and the result is stored in ImageSelection. So this operation implements $A = A - B$ where A is ImageSelection and B is ReferenceImageSelection. The ReferenceImageSelection images are not altered.
- Multiple image volumes can be selected by both ImageSelection and ReferenceImageSelection. For each ImageSelection volume, each of the ReferenceImageSelection volumes are subtracted sequentially.

2.125.3 Parameters

- ImageSelection
- ReferenceImageSelection

2.125.3.1 ImageSelection

2.125.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.125.3.1.2 Default

- `"last"`

2.125.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.125.3.2 ReferenceImageSelection

2.125.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.125.3.2.2 Default

- `"!last"`

2.125.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`

- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.126 SupersampleImageGrid

2.126.1 Description

This operation scales supersamples images so they have more rows and/or columns, but the whole image keeps its shape and spatial extent. This operation is typically used for zooming into images or trying to ensure a sufficient number of voxels are within small contours.

2.126.2 Notes

- Be aware that specifying large multipliers (or even small multipliers on large images) will consume much memory. It is best to pre-crop images to a region of interest if possible.

2.126.3 Parameters

- ImageSelection
- RowScaleFactor
- ColumnScaleFactor
- SliceScaleFactor
- SamplingMethod

2.126.3.1 ImageSelection

2.126.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.126.3.1.2 Default

- `"last"`

2.126.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.126.3.2 RowScaleFactor

2.126.3.2.1 Description

A positive integer specifying how many rows will be in the new images. The number is relative to the incoming image row count. Specifying '1' will result in nothing happening. Specifying '8' will result in 8x as many rows.

2.126.3.2.2 Default

- `"2"`

2.126.3.2.3 Examples

- `"1"`
- `"2"`
- `"3"`
- `"8"`

2.126.3.3 ColumnScaleFactor

2.126.3.3.1 Description

A positive integer specifying how many columns will be in the new images. The number is relative to the incoming image column count. Specifying '1' will result in nothing happening. Specifying '8' will result in 8x as many columns.

2.126.3.3.2 Default

- `"2"`

2.126.3.3.3 Examples

- `"1"`
- `"2"`
- `"3"`
- `"8"`

2.126.3.4 SliceScaleFactor

2.126.3.4.1 Description

A positive integer specifying how many image slices will be in the new images. The number is relative to the incoming image slice count. Specifying '1' will result in nothing happening. Specifying '8' will result in 8x as many slices. Note that slice supersampling always happens *after* in-plane supersampling. Also note that merely setting this factor will not enable 3D supersampling; you also need to specify a 3D-aware SamplingMethod.

2.126.3.4.2 Default

- `"2"`

2.126.3.4.3 Examples

- `"1"`
- `"2"`
- `"3"`
- `"8"`

2.126.3.5 SamplingMethod

2.126.3.5.1 Description

The supersampling method to use. Note: 'inplane-' methods only consider neighbours in the plane of a single image – neighbours in adjacent images are not considered and the supersampled image will contain the same number of image slices as the inputs.

2.126.3.5.2 Default

- `"inplane-bilinear"`

2.126.3.5.3 Examples

- `"inplane-bicubic"`
- `"inplane-bilinear"`
- `"trilinear"`

2.127 SurfaceBasedRayCastDoseAccumulate

2.127.1 Description

This routine uses rays (actually: line segments) to estimate point-dose on the surface of an ROI. The ROI is approximated by surface mesh and rays are passed through. Dose is interpolated at the intersection points and intersecting lines (i.e., where the ray 'glances' the surface) are discarded. The surface reconstruction can be tweaked, but appear to reasonably approximate the ROI contours; both can be output to compare visually. Though it is not required by the implementation, only the ray-surface intersection nearest to the detector is considered. All other intersections (i.e., on the far side of the surface mesh) are ignored. This routine is fairly fast compared to the slow grid-based counterpart previously implemented. The speedup comes from use of an AABB-tree to accelerate intersection queries and avoid having to 'walk' rays step-by-step through over/through the geometry.

2.127.2 Parameters

- TotalDoseMapFileName
- RefCroppedTotalDoseMapFileName
- IntersectionCountMapFileName
- DepthMapFileName
- RadialDistMapFileName
- RefIntersectionCountMapFileName
- ROISurfaceMeshFileName
- SubdividedROISurfaceMeshFileName
- RefSurfaceMeshFileName

- SubdividedRefSurfaceMeshFileName
- ROICOMCOMLineFileName
- NormalizedReferenceROILabelRegex
- NormalizedROILabelRegex
- ReferenceROILabelRegex
- ROILabelRegex
- SourceDetectorRows
- SourceDetectorColumns
- MeshingSubdivisionIterations
- MaxRaySurfaceIntersections
- OnlyGenerateSurface

2.127.2.1 TotalDoseMapFileName

2.127.2.1.1 Description

A filename (or full path) for the total dose image map (at all ray-surface intersection points). The dose for each ray is summed over all ray-surface point intersections. The format is FITS. This file is always generated. Leave the argument empty to generate a unique filename.

2.127.2.1.2 Default

- `""`

2.127.2.1.3 Examples

- `""`
- `"total_dose_map.fits"`
- `"/tmp/out.fits"`

2.127.2.2 RefCroppedTotalDoseMapFileName

2.127.2.2.1 Description

A filename (or full path) for the total dose image map (at all ray-surface intersection points). The dose for each ray is summed over all ray-surface point intersections. Doses in this map are only registered when the ray intersects the reference ROI mesh. The format is FITS. This file is always generated. Leave the argument empty to generate a unique filename.

2.127.2.2.2 Default

- `""`

2.127.2.2.3 Examples

- `""`
- `"total_dose_map.fits"`
- `"/tmp/out.fits"`

2.127.2.3 IntersectionCountMapFileName

2.127.2.3.1 Description

A filename (or full path) for the (number of ray-surface intersections) image map. Each pixel in this map (and the total dose map) represents a single ray; the number of times the ray intersects the surface can be useful for various purposes, but most often it will simply be a sanity check for the cross-sectional shape or that a specific number of intersections were recorded in regions with geometrical folds. Pixels will all be within [0,MaxRaySurfaceIntersections]. The format is FITS. Leave empty to dump to generate a unique filename.

2.127.2.3.2 Default

- `""`

2.127.2.3.3 Examples

- `""`
- `"intersection_count_map.fits"`
- `"/tmp/out.fits"`

2.127.2.4 DepthMapFileName

2.127.2.4.1 Description

A filename (or full path) for the distance (depth) of each ray-surface intersection point from the detector. Has DICOM coordinate system units. This image is potentially multi-channel with MaxRaySurfaceIntersections channels (when MaxRaySurfaceIntersections = 1 there is 1 channel). The format is FITS. Leaving empty will result in no file being written.

2.127.2.4.2 Default

- `""`

2.127.2.4.3 Examples

- `""`
- `"depth_map.fits"`
- `"/tmp/out.fits"`

2.127.2.5 RadialDistMapFileName

2.127.2.5.1 Description

A filename (or full path) for the distance of each ray-surface intersection point from the line joining reference and target ROI centre-of-masses. This helps quantify position in 3D. Has DICOM coordinate system units. This image is potentially multi-channel with MaxRaySurfaceIntersections channels (when MaxRaySurfaceIntersections = 1 there is 1 channel). The format is FITS. Leaving empty will result in no file being written.

2.127.2.5.2 Default

- `""`

2.127.2.5.3 Examples

- `""`
- `"radial_dist_map.fits"`
- `"/tmp/out.fits"`

2.127.2.6 RefIntersectionCountMapFileName

2.127.2.6.1 Description

A filename (or full path) for the (number of ray-surface intersections) for the reference ROIs. Each pixel in this map (and the total dose map) represents a single ray; the number of times the ray intersects the surface can be useful for various purposes, but most often it will simply be a sanity check for the cross-sectional shape or that a specific number of intersections were recorded in regions with geometrical folds. Note: currently, the number of intersections is limited to 0 or 1! The format is FITS. Leave empty to dump to generate a unique filename.

2.127.2.6.2 Default

- `""`

2.127.2.6.3 Examples

- `""`
- `"ref_roi_intersection_count_map.fits"`
- `"/tmp/out.fits"`

2.127.2.7 ROISurfaceMeshFileName

2.127.2.7.1 Description

A filename (or full path) for the (pre-subdivided) surface mesh that is contructed from the ROI contours. The format is OFF. This file is mostly useful for inspection of the surface or comparison with contours. Leaving empty will result in no file being written.

2.127.2.7.2 Default

- `""`

2.127.2.7.3 Examples

- `""`
- `"/tmp/roi_surface_mesh.off"`
- `"roi_surface_mesh.off"`

2.127.2.8 SubdividedROISurfaceMeshFileName

2.127.2.8.1 Description

A filename (or full path) for the Loop-subdivided surface mesh that is contructed from the ROI contours. The format is OFF. This file is mostly useful for inspection of the surface or comparison with contours. Leaving empty will result in no file being written.

2.127.2.8.2 Default

- `""`

2.127.2.8.3 Examples

- `""`
- `"/tmp/subdivided_roi_surface_mesh.off"`
- `"subdivided_roi_surface_mesh.off"`

2.127.2.9 RefSurfaceMeshFileName

2.127.2.9.1 Description

A filename (or full path) for the (pre-subdivided) surface mesh that is contructed from the reference ROI contours. The format is OFF. This file is mostly useful for inspection of the surface or comparison with contours. Leaving empty will result in no file being written.

2.127.2.9.2 Default

- `""`

2.127.2.9.3 Examples

- `""`
- `"/tmp/roi_surface_mesh.off"`
- `"roi_surface_mesh.off"`

2.127.2.10 SubdividedRefSurfaceMeshFileName

2.127.2.10.1 Description

A filename (or full path) for the Loop-subdivided surface mesh that is contructed from the reference ROI contours. The format is OFF. This file is mostly useful for inspection of the surface or comparison with contours. Leaving empty will result in no file being written.

2.127.2.10.2 Default

- `""`

2.127.2.10.3 Examples

- `""`
- `"/tmp/subdivided_roi_surface_mesh.off"`
- `"subdivided_roi_surface_mesh.off"`

2.127.2.11 ROICOMCOMLineFileName

2.127.2.11.1 Description

A filename (or full path) for the line segment that connected the centre-of-mass (COM) of reference and target ROI. The format is OFF. This file is mostly useful for inspection of the surface or comparison with contours. Leaving empty will result in no file being written.

2.127.2.11.2 Default

- `""`

2.127.2.11.3 Examples

- `""`
- `"/tmp/roi_com_com_line.off"`
- `"roi_com_com_line.off"`

2.127.2.12 NormalizedReferenceROILabelRegex

2.127.2.12.1 Description

A regex matching reference ROI labels/names to consider. The default will match all available ROIs, which is non-sensical. The reference ROI is used to orient the cleaving plane to trim the grid surface mask.

2.127.2.12.2 Default

- `".*"`

2.127.2.12.3 Examples

- `".*"`
- `".*Prostate.*"`
- `"Left Kidney"`
- `"Gross Liver"`

2.127.2.13 NormalizedROILabelRegex

2.127.2.13.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them.

All ROIs have to match the single regex, so use the ‘or’ token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.127.2.13.2 Default

- `".*"`

2.127.2.13.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.127.2.14 ReferenceROILabelRegex

2.127.2.14.1 Description

A regex matching reference ROI labels/names to consider. The default will match all available ROIs, which is non-sensical. The reference ROI is used to orient the cleaving plane to trim the grid surface mask.

2.127.2.14.2 Default

- `".*"`

2.127.2.14.3 Examples

- `".*"`
- `".*[pP]rostate.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.127.2.15 ROILabelRegex

2.127.2.15.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.127.2.15.2 Default

- `".*"`

2.127.2.15.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.127.2.16 SourceDetectorRows

2.127.2.16.1 Description

The number of rows in the resulting images, which also defines how many rays are used. (Each pixel in the source image represents a single ray.) Setting too fine relative to the surface mask grid or dose grid is futile.

2.127.2.16.2 Default

- `"1024"`

2.127.2.16.3 Examples

- `"100"`
- `"128"`
- `"1024"`
- `"4096"`

2.127.2.17 SourceDetectorColumns

2.127.2.17.1 Description

The number of columns in the resulting images. (Each pixel in the source image represents a single ray.) Setting too fine relative to the surface mask grid or dose grid is futile.

2.127.2.17.2 Default

- `"1024"`

2.127.2.17.3 Examples

- `"100"`
- `"128"`
- `"1024"`
- `"4096"`

2.127.2.18 MeshingSubdivisionIterations

2.127.2.18.1 Description

The number of iterations of Loop's subdivision to apply to the surface mesh. The aim of subdivision in this context is to have a smooth surface to work with, but too many applications will create too many facets. More facets will not lead to more precise results beyond a certain (modest) amount of smoothing. If the geometry is relatively spherical already, and meshing bounds produce reasonably smooth (but 'blocky') surface meshes, then 2-3 iterations should suffice. More than 3-4 iterations will almost always be inappropriate.

2.127.2.18.2 Default

- `"2"`

2.127.2.18.3 Examples

- `"0"`
- `"1"`
- `"2"`
- `"3"`

2.127.2.19 MaxRaySurfaceIntersections

2.127.2.19.1 Description

The maximum number of ray-surface intersections to accumulate before retiring each ray. Note that intersections are sorted spatially by their distance to the detector, and those closest to the detector are considered first. If the ROI surface is opaque, setting this value to 1 will emulate visibility. Setting to 2 will permit rays continue through the ROI and pass through the other side; dose will be the accumulation of dose at each ray-surface intersection. This value should most often be 1 or some very high number (e.g., 1000) to make the surface either completely opaque or completely transparent. (A transparent surface may help to visualize geometrical 'folds' or other surface details of interest.)

2.127.2.19.2 Default

- `"1"`

2.127.2.19.3 Examples

- `"1"`
- `"4"`
- `"1000"`

2.127.2.20 OnlyGenerateSurface

2.127.2.20.1 Description

Stop processing after writing the surface and subdivided surface meshes. This option is primarily used for debugging and visualization.

2.127.2.20.2 Default

- `"false"`

2.127.2.20.3 Examples

- `"true"`
- `"false"`

2.128 ThresholdImages

2.128.1 Description

This operation applies thresholds to images. Both upper and lower thresholds can be specified.

2.128.2 Notes

- This routine operates on individual images. When thresholds are specified on a percentile basis, each image is considered separately and therefore each image may be thresholded with different values.
- Both thresholds are inclusive. To binarize an image, use the same threshold for both upper and lower threshold parameters. Voxels that fall on the threshold will currently be treated as if they exclusively satisfy the upper threshold, but this behaviour is not guaranteed.

2.128.3 Parameters

- Lower
- Low
- Upper
- High
- Channel
- ImageSelection

2.128.3.1 Lower

2.128.3.1.1 Description

The lower bound (inclusive). Pixels with values < this number are replaced with the 'low' value. If this number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If this number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.128.3.1.2 Default

- `"-inf"`

2.128.3.1.3 Examples

- `"0.0"`
- `"-1E-99"`
- `"1.23"`
- `"0.2%"`
- `"23tile"`
- `"23.123 tile"`

2.128.3.2 Low

2.128.3.2.1 Description

The value a pixel will take when below the lower threshold.

2.128.3.2.2 Default

- `"-inf"`

2.128.3.2.3 Examples

- `"0.0"`
- `"-1000.0"`
- `"-inf"`
- `"nan"`

2.128.3.3 Upper

2.128.3.3.1 Description

The upper bound (inclusive). Pixels with values > this number are replaced with the 'high' value. If this number is followed by a '%', the bound will be scaled between the min and max pixel values [0-100%]. If this number is followed by 'tile', the bound will be replaced with the corresponding percentile [0-100tile]. Note that upper and lower bounds can be specified separately (e.g., lower bound is a percentage, but upper bound is a percentile).

2.128.3.3.2 Default

- `"inf"`

2.128.3.3.3 Examples

- `"1.0"`
- `"1E-99"`
- `"2.34"`
- `"98.12%"`
- `"94tile"`
- `"94.123 tile"`

2.128.3.4 High

2.128.3.4.1 Description

The value a pixel will take when above the upper threshold.

2.128.3.4.2 Default

- `"inf"`

2.128.3.4.3 Examples

- `"0.0"`
- `"1000.0"`
- `"inf"`
- `"nan"`

2.128.3.5 Channel

2.128.3.5.1 Description

The image channel to use. Zero-based.

2.128.3.5.2 Default

- `"0"`

2.128.3.5.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.128.3.6 ImageSelection

2.128.3.6.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.128.3.6.2 Default

- `"last"`

2.128.3.6.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.129 ThresholdOtsu

2.129.1 Description

This routine performs Otsu thresholding (i.e., 'binarization') on an image volume. The thresholding is limited within ROI(s). Otsu thresholding works best on images with a well-defined bimodal voxel intensity histogram. It works by finding the threshold that partitions the voxel intensity histogram into two parts, essentially so that the sum of each partition's variance is minimal. The number of histogram bins (i.e., number of distinct voxel magnitude levels) is configurable. Voxels are binarized; the replacement values are also configurable.

2.129.2 Notes

- The Otsu method will not necessarily cleanly separate bimodal peaks in the voxel intensity histogram.

2.129.3 Parameters

- ImageSelection

- HistogramBins
- ReplacementLow
- ReplacementHigh
- OverwriteVoxels
- Channel
- NormalizedROILabelRegex
- ROILabelRegex
- ContourOverlap
- Inclusivity

2.129.3.1 ImageSelection

2.129.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.129.3.1.2 Default

- `"last"`

2.129.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`

- `"key1@.*value1.*;key2@^value2$;first"`

2.129.3.2 HistogramBins

2.129.3.2.1 Description

The number of equal-width bins the histogram should have. Classically, images were 8-bit integer-valued and thus 255 bins were commonly used. However, because floating-point numbers are used practically any number of bins are supported. What is optimal (or acceptable) depends on the analytical requirements. If the threshold does not have to be exact, try use the smallest number of bins you can get away with; 50-150 should suffice. This will speed up computation. If the threshold is being used for analytical purposes, use as many bins as the data can support – if the voxel values span only 8-bit integers, having more than 255 bins will not improve the analysis. Likewise if voxels are discretized or sparse. Experiment by gradually increasing the number of bins until the threshold value converges to a reasonable number, and then use that number of bins for future analysis.

2.129.3.2.2 Default

- `"255"`

2.129.3.2.3 Examples

- `"10"`
- `"50"`
- `"100"`
- `"200"`
- `"500"`

2.129.3.3 ReplacementLow

2.129.3.3.1 Description

The value to give voxels which are below (exclusive) the Otsu threshold value.

2.129.3.3.2 Default

- `"0.0"`

2.129.3.3.3 Examples

- `"-1.0"`
- `"0.0"`
- `"1.23"`
- `"nan"`
- `"inf"`

2.129.3.4 ReplacementHigh

2.129.3.4.1 Description

The value to give voxels which are above (inclusive) the Otsu threshold value.

2.129.3.4.2 Default

- `"1.0"`

2.129.3.4.3 Examples

- `"-1.0"`
- `"0.0"`
- `"1.23"`
- `"nan"`
- `"inf"`

2.129.3.5 OverwriteVoxels

2.129.3.5.1 Description

Controls whether voxels should actually be binarized or not. Whether or not voxel intensities are overwritten, the Otsu threshold value is written into the image metadata as ‘OtsuThreshold’ in case further processing is needed.

2.129.3.5.2 Default

- `"true"`

2.129.3.5.3 Examples

- `"true"`
- `"false"`

2.129.3.6 Channel

2.129.3.6.1 Description

The image channel to use. Zero-based.

2.129.3.6.2 Default

- `"0"`

2.129.3.6.3 Examples

- `"0"`
- `"1"`
- `"2"`

2.129.3.7 NormalizedROILabelRegex

2.129.3.7.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.129.3.7.2 Default

- `".*"`

2.129.3.7.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.129.3.8 ROILabelRegex

2.129.3.8.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.129.3.8.2 Default

- `".*"`

2.129.3.8.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.129.3.9 ContourOverlap

2.129.3.9.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.129.3.9.2 Default

- `"ignore"`

2.129.3.9.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.129.3.10 Inclusivity

2.129.3.10.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.129.3.10.2 Default

- `"center"`

2.129.3.10.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.130 TransformMeshes

2.130.1 Description

This operation transforms meshes by translating, scaling, and rotating vertices.

2.130.2 Notes

- A single transformation can be specified at a time. Perform this operation sequentially to enforce order.

2.130.3 Parameters

- MeshSelection
- Transform

2.130.3.1 MeshSelection

2.130.3.1.1 Description

Select one or more surface meshes. Note that a single surface mesh may hold many disconnected mesh components; they should collectively represent a single logically cohesive object. Be aware that it is possible to mix logically unrelated sub-meshes together in a single mesh. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth surface mesh (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last surface mesh. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.130.3.1.2 Default

- `"last"`

2.130.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.130.3.2 Transform

2.130.3.2.1 Description

This parameter is used to specify the transformation that should be performed. A single transformation can be specified for each invocation of this operation. Currently translation, scaling, and rotation are available. Translations have three configurable scalar parameters denoting the translation along x, y, and z in the DICOM coordinate system. Translating $x = 1.0$, $y = -2.0$, and $z = 0.3$ can be specified as 'translate(1.0, -2.0, 0.3)'. The scale transformation has four configurable scalar parameters denoting the scale centre 3-vector and the magnification factor. Note that the magnification factor can be negative, which will cause the mesh to be inverted along x, y, and z axes and magnified. Take note that face orientations will also become inverted. Magnifying by 2.7x about $(1.23, -2.34, 3.45)$ can be specified as 'scale(1.23, -2.34, 3.45, 2.7)'. Rotations around an arbitrary axis line can be accomplished. The rotation transformation has seven configurable scalar parameters denoting the rotation centre 3-vector, the rotation axis 3-vector, and the rotation angle in radians. A rotation of pi radians around the axis line parallel to vector $(1.0, 0.0, 0.0)$ that intersects the point $(4.0, 5.0, 6.0)$ can be specified as 'rotate(4.0, 5.0, 6.0, 1.0, 0.0, 0.0, 3.141592653)'.

2.130.3.2.2 Default

- `"translate(0.0, 0.0, 0.0)"`

2.130.3.2.3 Examples

- `"translate(1.0, -2.0, 0.3)"`
- `"scale(1.23, -2.34, 3.45, 2.7)"`
- `"rotate(4.0, 5.0, 6.0,  1.0, 0.0, 0.0,  3.141592653)"`

2.131 TrimROIDose

2.131.1 Description

This operation provides a simplified interface for overriding the dose within a ROI. For example, this operation can be used to modify a base plan by eliminating dose that coincides with a PTV/CTV/GTV/ROI etc.

2.131.2 Notes

- This operation performs the opposite of the 'Crop' operation, which trims the dose outside a ROI.
- The inclusivity of a dose voxel that straddles the ROI boundary can be specified in various ways. Refer to the Inclusivity parameter documentation.
- By default this operation only overrides dose within a ROI. The opposite, overriding dose outside of a ROI, can be accomplished using the expert interface.

2.131.3 Parameters

- Channel
- ImageSelection
- ContourOverlap
- Inclusivity
- ExteriorVal
- InteriorVal
- ExteriorOverwrite
- InteriorOverwrite
- NormalizedROILabelRegex
- ROILabelRegex
- Filename
- ParanoiaLevel

2.131.3.1 Channel

2.131.3.1.1 Description

The image channel to use. Zero-based. Use '-1' to operate on all available channels.

2.131.3.1.2 Default

- `"-1"`

2.131.3.1.3 Examples

- `"-1"`
- `"0"`
- `"1"`
- `"2"`

2.131.3.2 ImageSelection

2.131.3.2.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array.

Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.131.3.2.2 Default

- `"all"`

2.131.3.2.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.131.3.3 ContourOverlap

2.131.3.3.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.131.3.3.2 Default

- `"ignore"`

2.131.3.3.3 Examples

- `"ignore"`

- "honour_opposite_orientations"
- "overlapping_contours_cancel"
- "honour_opps"
- "overlap_cancel"

2.131.3.4 Inclusivity

2.131.3.4.1 Description

Controls how voxels are deemed to be ‘within’ the interior of the selected ROI(s). The default ‘center’ considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, ‘planar_corner_inclusive’, considers a voxel interior if ANY corner is interior. The second, ‘planar_corner_exclusive’, considers a voxel interior if ALL (four) corners are interior.

2.131.3.4.2 Default

- "planar_inc"

2.131.3.4.3 Examples

- "center"
- "centre"
- "planar_corner_inclusive"
- "planar_inc"
- "planar_corner_exclusive"
- "planar_exc"

2.131.3.5 ExteriorVal

2.131.3.5.1 Description

The value to give to voxels outside the specified ROI(s). Note that this value will be ignored if exterior overwrites are disabled.

2.131.3.5.2 Default

- "0.0"

2.131.3.5.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.131.3.6 InteriorVal

2.131.3.6.1 Description

The value to give to voxels within the volume of the specified ROI(s). Note that this value will be ignored if interior overwrites are disabled.

2.131.3.6.2 Default

- `"0.0"`

2.131.3.6.3 Examples

- `"0.0"`
- `"-1.0"`
- `"1.23"`
- `"2.34E26"`

2.131.3.7 ExteriorOverwrite

2.131.3.7.1 Description

Whether to overwrite voxels exterior to the specified ROI(s).

2.131.3.7.2 Default

- `"false"`

2.131.3.7.3 Examples

- `"true"`
- `"false"`

2.131.3.8 InteriorOverwrite

2.131.3.8.1 Description

Whether to overwrite voxels interior to the specified ROI(s).

2.131.3.8.2 Default

- `"true"`

2.131.3.8.3 Examples

- `"true"`
- `"false"`

2.131.3.9 NormalizedROILabelRegex

2.131.3.9.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.131.3.9.2 Default

- `".*"`

2.131.3.9.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.131.3.10 ROILabelRegex

2.131.3.10.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them.

All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.131.3.10.2 Default

- `".*"`

2.131.3.10.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.131.3.11 Filename

2.131.3.11.1 Description

The filename (or full path name) to which the DICOM file should be written.

2.131.3.11.2 Default

- `"/tmp/RD.dcm"`

2.131.3.11.3 Examples

- `"/tmp/RD.dcm"`
- `"./RD.dcm"`
- `"RD.dcm"`

2.131.3.12 ParanoiaLevel

2.131.3.12.1 Description

At low paranoia setting, only top-level UIDs are replaced. At medium paranoia setting, many UIDs, descriptions, and labels are replaced, but the PatientID and FrameOfReferenceUID are retained. The high paranoia setting is the same as the medium setting, but the PatientID and FrameOfReferenceUID are also replaced. (Note: this is not a full anonymization.) Use the low setting if you want to retain linkage to the originating data set. Use the medium setting if you don't. Use the high setting if your TPS goes overboard linking data sets by PatientID and/or FrameOfReferenceUID.

2.131.3.12.2 Default

- "medium"

2.131.3.12.3 Examples

- "low"
- "medium"
- "high"

2.132 UBC3TMRI_DCE

2.132.1 Description

This operation is used to generate dynamic contrast-enhanced MRI contrast enhancement maps.

2.132.2 Parameters

No registered options.

2.133 UBC3TMRI_DCE_Differences

2.133.1 Description

This operation is used to generate dynamic contrast-enhanced MRI contrast enhancement maps.

2.133.2 Notes

- This routine generates difference maps using both long DCE scans. Thus it takes up a LOT of memory! Try avoid unnecessary copies of large (temporally long) arrays.

2.133.3 Parameters

No registered options.

2.134 UBC3TMRI_DCE_Experimental

2.134.1 Description

This operation is an experimental operation for processing dynamic contrast-enhanced MR images.

2.134.2 Parameters

No registered options.

2.135 UBC3TMRI_IVIM_ADC

2.135.1 Description

This operation is an experimental operation for processing IVIM MR images into ADC maps.

2.135.2 Parameters

No registered options.

2.136 VolumetricCorrelationDetector

2.136.1 Description

This operation can assess 3D correlations by sampling the neighbourhood surrounding each voxel and assigning a similarity score. This routine is useful for detecting repetitive (regular) patterns that are known in advance.

2.136.2 Notes

- The provided image collection must be rectilinear.
- At the moment this routine can only be modified via recompilation.

2.136.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex

- Low
- High
- Channel

2.136.3.1 ImageSelection

2.136.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.136.3.1.2 Default

- `"last"`

2.136.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.136.3.2 NormalizedROILabelRegex

2.136.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.136.3.2.2 Default

- `".*"`

2.136.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.136.3.3 ROILabelRegex

2.136.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.136.3.3.2 Default

- `".*"`

2.136.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.136.3.4 Low

2.136.3.4.1 Description

The low percentile.

2.136.3.4.2 Default

- `"0.05"`

2.136.3.4.3 Examples

- `"0.05"`
- `"0.5"`
- `"0.99"`

2.136.3.5 High

2.136.3.5.1 Description

The high percentile.

2.136.3.5.2 Default

- `"0.95"`

2.136.3.5.3 Examples

- `"0.95"`
- `"0.5"`
- `"0.05"`

2.136.3.6 Channel

2.136.3.6.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.136.3.6.2 Default

- `"-1"`

2.136.3.6.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.137 VolumetricSpatialBlur

2.137.1 Description

This operation performs blurring of voxel values within 3D rectilinear image arrays.

2.137.2 Notes

- The provided image collection must be rectilinear.

2.137.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Estimator

2.137.3.1 ImageSelection

2.137.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.137.3.1.2 Default

- `"last"`

2.137.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.137.3.2 NormalizedROILabelRegex

2.137.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.137.3.2.2 Default

- `".*"`

2.137.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.137.3.3 ROILabelRegex

2.137.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.137.3.3.2 Default

- `".*"`

2.137.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.137.3.4 Channel

2.137.3.4.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.137.3.4.2 Default

- `"-1"`

2.137.3.4.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.137.3.5 Estimator

2.137.3.5.1 Description

Controls which type of blur is computed. Currently, ‘Gaussian’ refers to a fixed sigma=1 (in pixel coordinates, not DICOM units) Gaussian blur that extends for 3*sigma thus providing a 7x7x7 window. Note that applying this kernel N times will approximate a Gaussian with sigma=N. Also note that boundary voxels will cause accessible voxels within the same window to be more heavily weighted. Try avoid boundaries or add extra margins if possible.

2.137.3.5.2 Default

- `"Gaussian"`

2.137.3.5.3 Examples

- `"Gaussian"`

2.138 VolumetricSpatialDerivative

2.138.1 Description

This operation estimates various spatial partial derivatives (of pixel values) within 3D rectilinear image arrays.

2.138.2 Notes

- The provided image collection must be rectilinear.

2.138.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- Channel
- Estimator
- Method

2.138.3.1 ImageSelection

2.138.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.138.3.1.2 Default

- `"last"`

2.138.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.138.3.2 NormalizedROILabelRegex

2.138.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.138.3.2.2 Default

- `".*"`

2.138.3.2.3 Examples

- `".*"`
- `".*Body.*"`
- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.138.3.3 ROILabelRegex

2.138.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.138.3.3.2 Default

- `".*"`

2.138.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.138.3.4 Channel

2.138.3.4.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.138.3.4.2 Default

- `"-1"`

2.138.3.4.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.138.3.5 Estimator

2.138.3.5.1 Description

Controls the finite-difference partial derivative order or estimator used. All estimators are centred and use mirror boundary conditions. First-order estimators include the basic nearest-neighbour first derivative and Sobel estimators. 'XxYxZ' denotes the size of the convolution kernel (i.e., the number of adjacent pixels considered).

2.138.3.5.2 Default

- `"Sobel-3x3x3"`

2.138.3.5.3 Examples

- `"first"`
- `"Sobel-3x3x3"`

2.138.3.6 Method

2.138.3.6.1 Description

Controls partial derivative method. First-order derivatives can be row-, column-, or image-aligned, All methods also support magnitude (addition of orthogonal components in quadrature).

2.138.3.6.2 Default

- `"magnitude"`

2.138.3.6.3 Examples

- `"row-aligned"`
- `"column-aligned"`
- `"image-aligned"`
- `"magnitude"`
- `"non-maximum-suppression"`

2.139 VoxelRANSAC

2.139.1 Description

This routine performs RANSAC fitting using voxel positions as inputs. The search can be confined within ROIs and a range of voxel intensities.

2.139.2 Notes

- This operation does not make use of voxel intensities during the RANSAC procedure. Voxel intensities are only used to identify which voxel positions are considered.

2.139.3 Parameters

- ImageSelection
- NormalizedROILabelRegex
- ROILabelRegex
- ContourOverlap
- Inclusivity
- Channel
- Lower
- Upper
- GridSeparation

2.139.3.1 ImageSelection

2.139.3.1.1 Description

Select one or more image arrays. Note that image arrays can hold anything, but will typically represent a single contiguous 3D volume (i.e., a volumetric CT scan) or '4D' time-series. Be aware that it is possible to mix logically unrelated images together. Selection specifiers can be of two types: positional or metadata-based key@value regex. Positional specifiers can be 'first', 'last', 'none', or 'all' literals. Additionally '#N' for some positive integer N selects the Nth image array (with zero-based indexing). Likewise, '#-N' selects the Nth-from-last image array. Positional specifiers can be inverted by prefixing with a '!'. Metadata-based key@value expressions are applied by matching the keys verbatim and the values with regex. In order to invert metadata-based selectors, the regex logic must be inverted (i.e., you can *not* prefix metadata-based selectors with a '!'). Multiple criteria can be specified by separating them with a ';' and are applied in the order specified. Both positional and metadata-based criteria can be mixed together. Note regexes are case insensitive and should use extended POSIX syntax.

2.139.3.1.2 Default

- `"last"`

2.139.3.1.3 Examples

- `"last"`
- `"first"`
- `"all"`
- `"none"`
- `"#0"`
- `"#-0"`
- `"!last"`
- `"!#-3"`
- `"key@.*value.*"`
- `"key1@.*value1.*;key2@^value2$;first"`

2.139.3.2 NormalizedROILabelRegex

2.139.3.2.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.139.3.2.2 Default

- `".*"`

2.139.3.2.3 Examples

- `".*"`
- `".*Body.*"`

- `"Body"`
- `"Gross_Liver"`
- `".*Left.*Parotid.*|.*Right.*Parotid.*|.*Eye.*"`
- `"Left Parotid|Right Parotid"`

2.139.3.3 ROILabelRegex

2.139.3.3.1 Description

A regex matching ROI labels/names to consider. The default will match all available ROIs. Be aware that input spaces are trimmed to a single space. If your ROI name has more than two sequential spaces, use regex to avoid them. All ROIs have to match the single regex, so use the 'or' token if needed. Regex is case insensitive and uses extended POSIX syntax.

2.139.3.3.2 Default

- `".*"`

2.139.3.3.3 Examples

- `".*"`
- `".*body.*"`
- `"body"`
- `"Gross_Liver"`
- `".*left.*parotid.*|.*right.*parotid.*|.*eyes.*"`
- `"left_parotid|right_parotid"`

2.139.3.4 ContourOverlap

2.139.3.4.1 Description

Controls overlapping contours are treated. The default 'ignore' treats overlapping contours as a single contour, regardless of contour orientation. The option 'honour_opposite_orientations' makes overlapping contours with opposite orientation cancel. Otherwise, orientation is ignored. The latter is useful for Boolean structures where contour orientation is significant for interior contours (holes). The option 'overlapping_contours_cancel' ignores orientation and cancels all contour overlap.

2.139.3.4.2 Default

- `"ignore"`

2.139.3.4.3 Examples

- `"ignore"`
- `"honour_opposite_orientations"`
- `"overlapping_contours_cancel"`
- `"honour_opps"`
- `"overlap_cancel"`

2.139.3.5 Inclusivity

2.139.3.5.1 Description

Controls how voxels are deemed to be 'within' the interior of the selected ROI(s). The default 'center' considers only the central-most point of each voxel. There are two corner options that correspond to a 2D projection of the voxel onto the image plane. The first, 'planar_corner_inclusive', considers a voxel interior if ANY corner is interior. The second, 'planar_corner_exclusive', considers a voxel interior if ALL (four) corners are interior.

2.139.3.5.2 Default

- `"center"`

2.139.3.5.3 Examples

- `"center"`
- `"centre"`
- `"planar_corner_inclusive"`
- `"planar_inc"`
- `"planar_corner_exclusive"`
- `"planar_exc"`

2.139.3.6 Channel

2.139.3.6.1 Description

The channel to operated on (zero-based). Negative values will cause all channels to be operated on.

2.139.3.6.2 Default

- `"0"`

2.139.3.6.3 Examples

- `"-1"`
- `"0"`
- `"1"`

2.139.3.7 Lower

2.139.3.7.1 Description

Lower threshold (inclusive) below which voxels will be ignored by this routine.

2.139.3.7.2 Default

- `"-inf"`

2.139.3.7.3 Examples

- `"-inf"`
- `"0.0"`
- `"1024"`

2.139.3.8 Upper

2.139.3.8.1 Description

Upper threshold (inclusive) above which voxels will be ignored by this routine.

2.139.3.8.2 Default

- `"inf"`

2.139.3.8.3 Examples

- `"inf"`
- `"1.0"`
- `"2048"`

2.139.3.9 GridSeparation

2.139.3.9.1 Description

The known separation of the grid (in DICOM units; mm) being sought.

2.139.3.9.2 Default

- `"nan"`

2.139.3.9.3 Examples

- `"1.0"`
- `"1.5"`
- `"10.0"`
- `"1.23E4"`

3 Known Issues and Limitations

3.1 Hanging on Debian

The SFML_Viewer operation hangs on some systems after viewing a plot with Gnuplot. This stems from a known issue in Ygor.

3.2 Build Requirements

DICOMautomaton depends on several heavily templated libraries and external projects. It requires a considerable amount of memory to build.

3.3 DICOM-RT Support Incomplete

Support for the DICOM Radiotherapy extensions are limited. In particular, only RTDOSE files can currently be exported, and RTPLAN files are not supported at all. Read support for DICOM image modalities and RTSTRUCTS are generally supported well. Broader DICOM support is planned for a future release.