Radiotherapy Support Functions (RSF)

Radiotherapy Support Functions are general radiotherapy worker functions. Most RSFs are used in various Dose Calculation Functions and Design Task Functions and are exposed through the external API to allow users a more low level access to the system. The RSF category encompasses the remaining functions not classified as a DTF or DCF. The available RSFs are listed below in groupings according to their typical usage and purpose.

Image Processing

Below is a list of some common image processing functions and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **override_image_inside_structure:**
  - Returns a new 3D image where the value of each voxel that is more than cutoff % contained within the structure is set to the provided override value
- **override_image_outside_structure:**
  - Modifies an image where the value of each voxel that is more than cutoff % contained within the structure is set to the provided override value (values outside the structure are not modified)
- **override_image_variant_outside_structure:**
  - Returns a new 3D image where the value of each voxel that is more than cutoff % outside the structure is set to the provided override value
- **image_histogram:**
  - Creates a histogram using the specified 1D image
- **combine_images_<N>d:**
  - Where N is the size of the image
  - Combine multiple images into single image (note this is NOT combining slices to construct higher order images, it is overlaying image data)
- **image_bounding_box_<N>d:**
  - Where N is the size of the image
  - Returns the bounding box of an image of size N
- **bounding_box_<N>d:**
  - Where N is the size of the image
  - Returns the bounding box of an image_geometry of size N
- **image_min_max_<N>d:**
  - Where N is the size of the image
  - Get the minimum and maximum values in the given image
- **image_list_min_max_<N>d:**
  - Where N is the size of the image
  - Get the overall minimum and maximum values for a list of images
- **create_uniform_image_on_grid_<N>d:**
  - Where N is the size of the image

    - Create an image of uniform pixel values (e.g. water phantom) over a grid of dimension N
    - See thinknode™ Examples for python examples of using this function

Contour Modification

Below is a list of some common polygon and polyset functions and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **polygon_centroid:**
    - Computes the geometric center of a polygon
- **scale_polygon:**
    - Scales a polygon shape in XY (independently) based on a vector2D factor
- **scale_polyset:**
    - Scales a polyset shape in XY (independently) based on a vector2D factor
- **polyset_expansion:**
    - Expands a polyset uniformly around the edges by the given amount. This function can be used to either expand or contract a polyset
- **polyset_combination:**
    - Compute a combination of two or more polysets. This function can operate as a union, intersection, difference, or “exclusive or” (xor)
- **point_list_bounding_box_<N>d:**
    - Where N is the size of the vector (1, 2, 3 dimensions)
    - Computes the bounding box of a list of N dimensional vectors
- **point_in_polygon / point_in_polyset:**
    - Test if a point is inside a polygon / polyset
- **distance_to_polyset:**
    - Get the distance from point to a polyset (inside < 0)

Structure Modification

Below is a list of some common structure manipulation functions and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **structure_combination:**
    - Compute a combination of two or more structures. This function can operate as a union, intersection, difference, or exclusive or (xor)
- **structure_2d_expansion:**
    - Compute the 2D expansion of a structure. The 2D expansion of a structure is computed by independently expanding each slice of the structure within its 2D plane. This function can be used to either expand or contract a structure
- **structure_3d_expansion:**
    - When computing the 3D expansion of a structure, the structure's slices are allowed to expand into other slices. This function can be used to either expand or contract a structure

- **structure_volume:**
  - Gets the volume of a structure_geometry
- **structure_centroid:**
  - Gets the centroid of a structure_geometry

Geometric Primitives

Below is a list of some common creation functions for geometric primitives and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **make_cube:**
  - Creates a triangle mesh representing a 3D box
  - See thinknode™ Examples for python examples of using this function
- **make_cylinder:**
  - Creates a triangle mesh representing an axis aligned, right 3D cylinder
- **make_pyramid:**
  - Creates a triangle mesh representing a rectangular based, right 3D pyramid
- **make_sphere:**
  - Creates a triangle mesh representing a 3D sphere
- **make_sliced_box:**
  - Creates a structure geometry representing a 3D box (using a sliced mesh)
- **make_sliced_cylinder:**
  - Creates a structure representing an axis aligned, right 3D cylinder (using a sliced mesh)
- **make_sliced_parallelepiped:**
  - Creates a structure representing a generalized 3D parallelepiped (using a sliced mesh)
- **make_sliced_pyramid:**
  - Creates a structure representing a rectangular based, right 3D pyramid (using a sliced mesh)
- **make_sliced_sphere:**
  - Creates a structure representing a 3D sphere (using a sliced mesh)

Degrader Manipulation

Below is a list of some common degrader manipulation functions and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **make_block:**
  - Create a degrader representing a block. A block has a uniform thickness within its shape and 0 thickness outside. Note that the shape is specified at the downstream edge of the block
- **make_shifter:**
  - A range shifter is modelled as extending infinitely in the X and Y directions, so its thickness is uniform across the entire field
- **make_rc:**
  - Create a degrader representing a range compensator. A range compensator is a degrader

whose thickness is specified as an image. The image is specified in the plane of the downstream edge of the RC
- **make_rc_nurb:**
  - Create a degrader representing a nurbs range compensator. A nurbs range compensator is a degrader whose thickness is specified as a smooth surface. The surface is specified in the plane of the downstream edge of the RC
- **truncate_rc:**
  - Shifts a range compensator surface such that the minimum thickness is set to the specified value
- **make_uniform_rc:**
  - Create a degrader representing a uniform thickness range compensator
- **make_linear_rc:**
  - Create a degrader representing a linearly varying thickness range compensator

By changing the input (shape, image, etc) passed into the the degrader make functions, the resulting degrader can be manipulated as desired.

PBS Energy Layers

Below is a list of some common functions that are helpful to compute appropriate machine deliverable energy layers and settings for PBS beams and a brief explanation of their intended usage (Specific details of each function, argument parameters, and return values are provided at the Dosimetry App Manifest Guide).

- **compute_range_extents_from_structure_geometry:**
  - Computes the minimum and maximum radiological path length necessary to reach the provided target for the a proton beam
- **create_pbs_layer_list:**
  - Creates a vector of machine deliverable energy layers (that contain no spots) that cover the specified range extents and are spaced according to the specified spacing strategy
- **create_pbs_layers_for_target:**
  - Computes a list of deliverable layers and spots that will fill the given target structure plus the nearby area as specified by the given margins and spot & layer spacing strategies
- **create_pbs_layers_from_spots:**
  - Converts a list of spots to deliverable calculation layers according to the provided machine definition; an error will be thrown if the provided spot list contains undeliverable spot energies

---