VXRE User Manual

VXRE is a software for reconstructing a 3D volume using multiple 2D X-ray projection images. The reconstruction process is based on the filtered back-projection algorithm for 3D cone-beam CT.

Table of Contents

System Requirements

	Minimum	Recommended
OS	Windows 7 SP1 (64-bit)	Windows 10 (64-bit)
Memory	4 GB*	16 GB*
Graphics card (GPU)	Nvidia GeForce 10 Series / 3 GB*	Nvidia GeForce 30 Series / 10 GB*
CPU	Intel Core i5	Intel Core i7 (8th gen or higher)
Disk free space	2 GB	64 GB

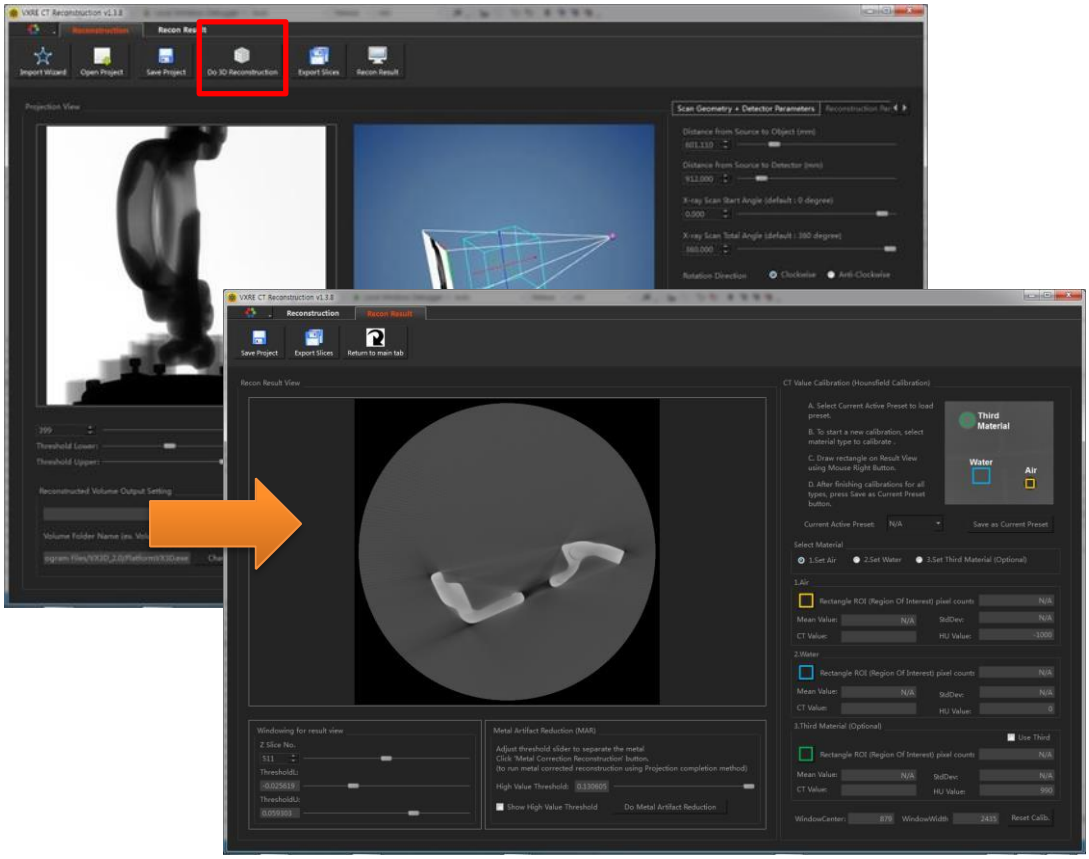
* Depending on data size.

Add-on Features

- Some features require an additional, separate license to use. The following are such licenses and the corresponding features:
    - VXRE AB
        - Autofocus
        - Beam hardening correction
    - VXRE Fly
        - Inline (on-the-fly) reconstruction
        - Projection data transfer (VXRE Communication API)

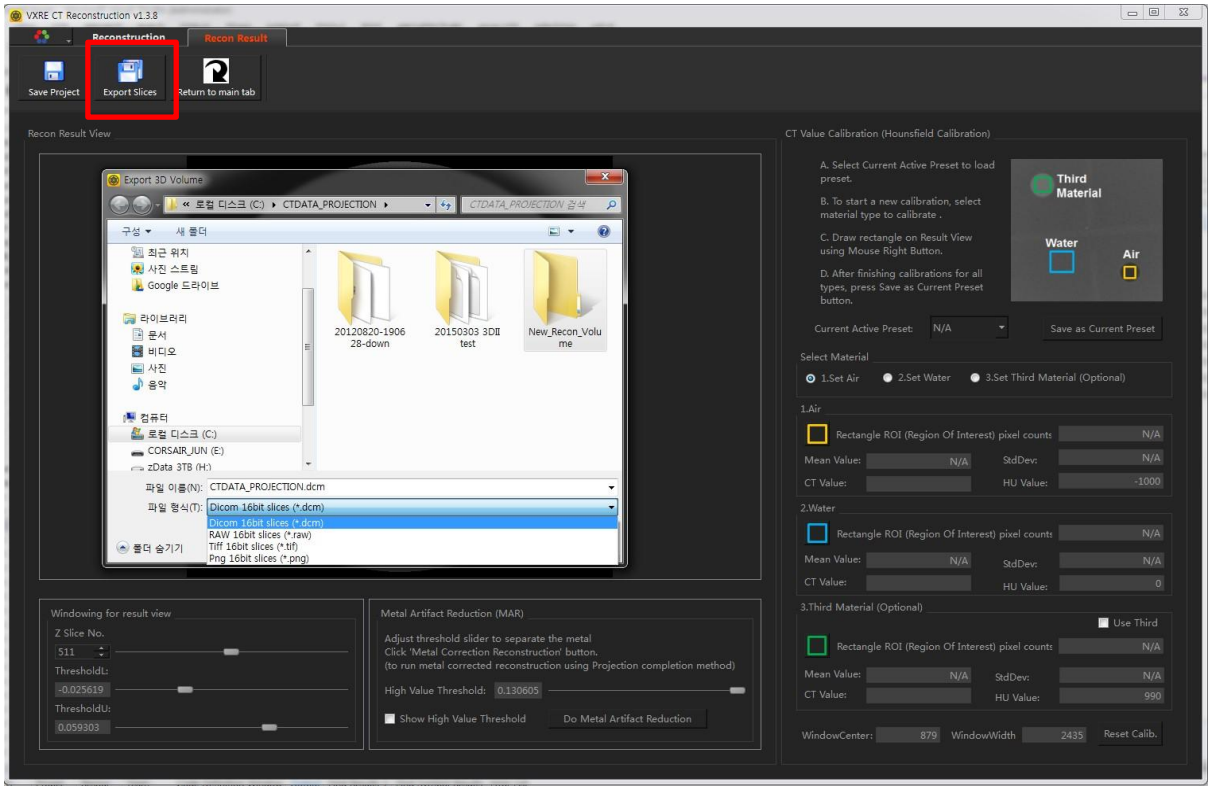
3D Cone Beam CT Reconstruction

- First, open a project file by clicking ‘Open Project’ button in the top toolbar.
- To start CT reconstruction, click ‘3D Reconstruction’ button in the top toolbar.
- Input projection (x-ray) images will be reconstructed as a 3D volume slices using FDK based filtered back-projection algorithm.



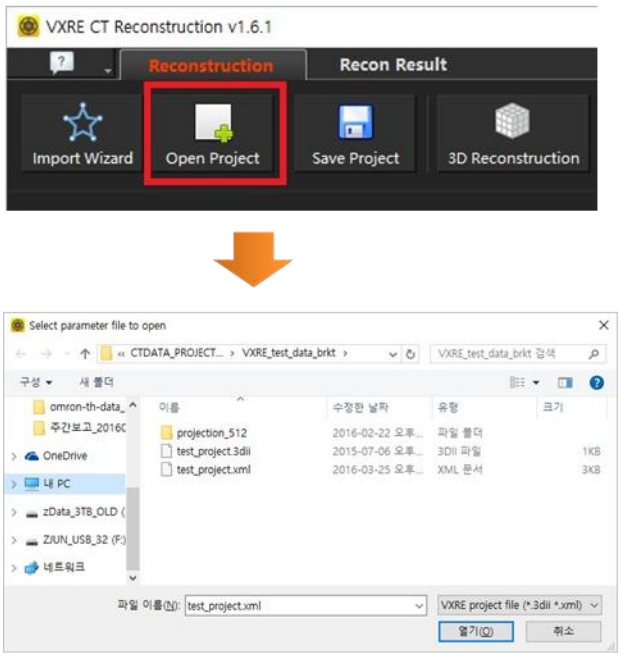
Export Slices

- After CT reconstruction, user can export (save) the reconstructed 3D volume data to slice files.
- The supported file formats are DICOM, RAW, TIFF and PNG.
- Volume slices will be saved as 16-bit by default.
    - For RAW format, 32-bit is also supported.



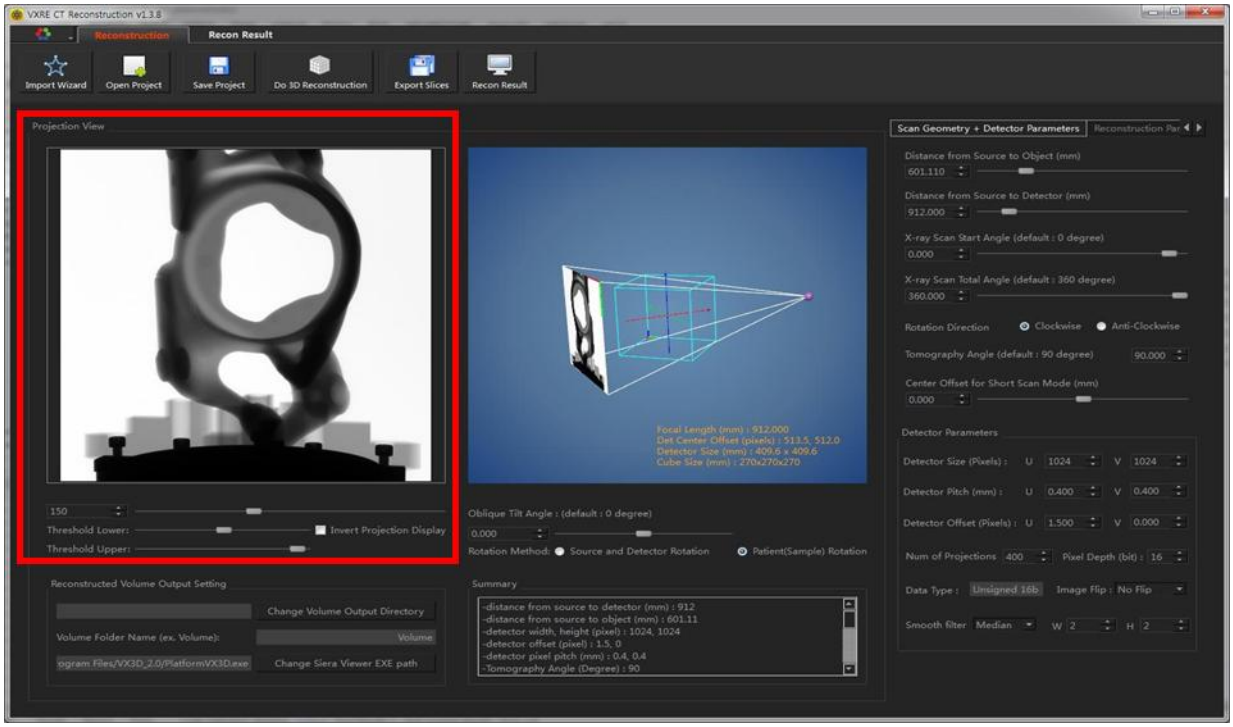
Main UI : Open Project

- Parameter setting can be done via a project file which contains all the parameters for CT reconstruction.
- When user click ‘Open Project’ button, a pop-up window will appear.
- Choose .xml project file with frame directory (which contains 2D projection images).
- By using the information from .xml file, 2D x-ray projection images will be loaded.
- The content of sample .xml file is shown above, including its relevant information such as Source to Detector distance, Source to object center, detector size, projection image information, etc. For more information about project file format, please refer to Appendix: Project File Format in this manual.



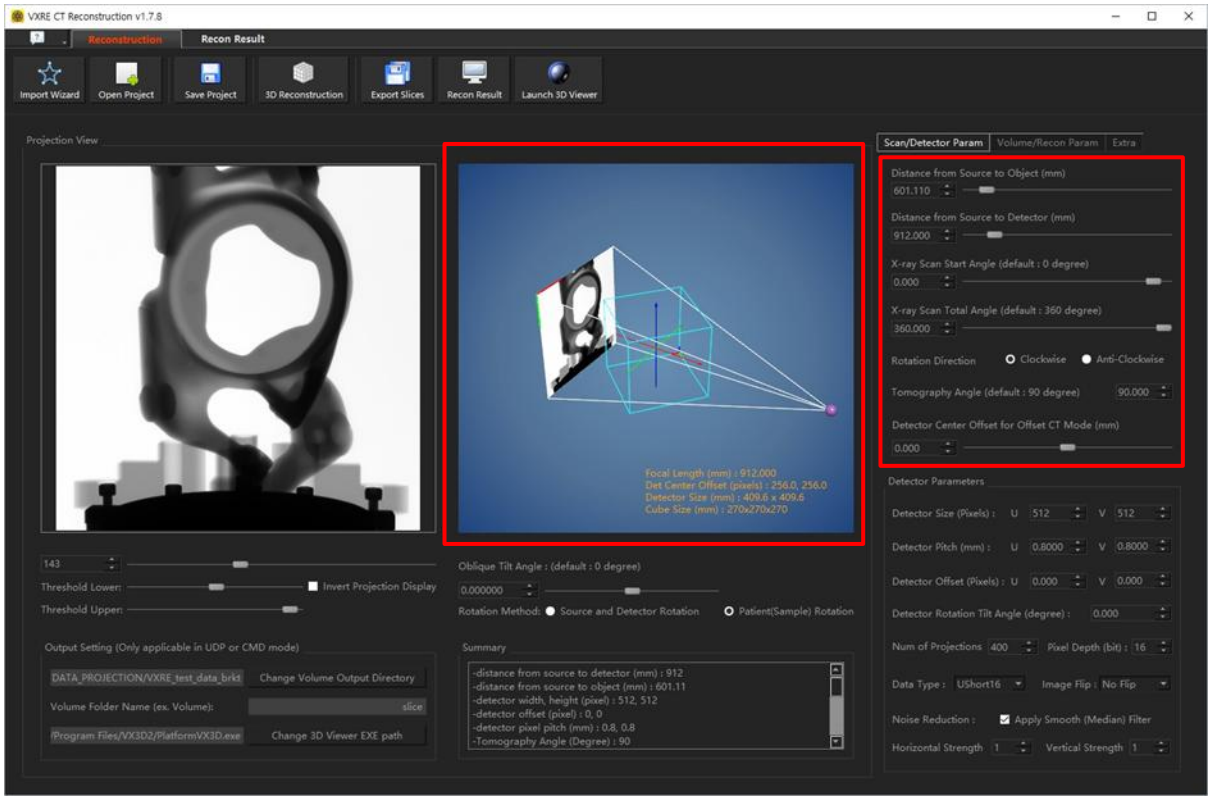
Main UI : Projection Image View

- After loading project file, the first projection image will be visualized on projection view.
- See other projection images by changing the image number of spin box control or by using the slider.
- Change the contrast of the projection image by adjusting ‘Threshold Lower’ and ‘Threshold Upper’ sliders.
- Invert the projection image result (White→Black, Black→White) by using the ‘Invert Projection Display’ check box.



Main UI : CT Geometry View (3D View)

- Current CT Geometry will be shown in geometry view as 3D lines and cube.
- If you change geometry parameters on right panel, CT Geometry view will get values and update immediately.
- Purple sphere represents the position of X-ray source position and the rectangle on the opposite side represents the detector plate.
- The blue box between the two represents the object (patient) region.
- The size and position of the geometry object are initialized based on the information from the loaded project file.
- Rotate the geometry object by left-clicking on CT geometry view and zoom-in and out object by mouse wheel operation.



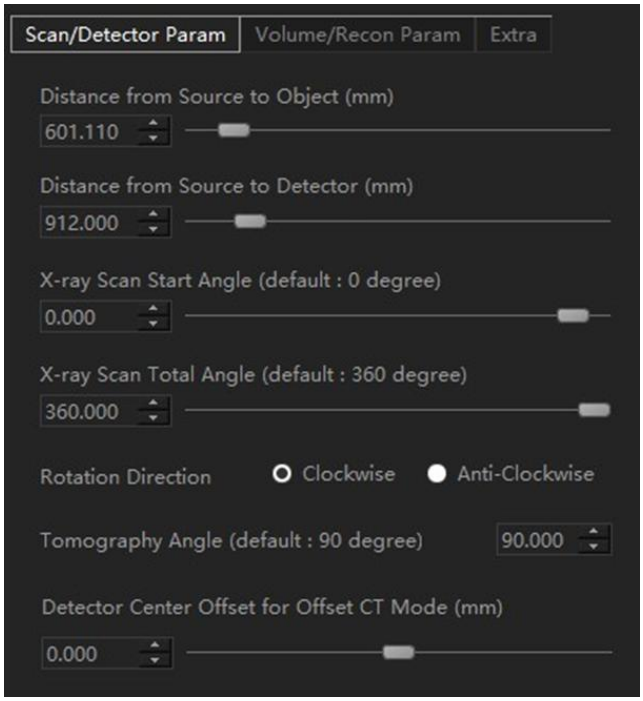
Main UI : Right Panel (Scan Geometry/Detector Parameter)

Modify CT geometry related settings such as distance from source to object, distance from source to detector, X-ray scan start/total angles on the right panel.

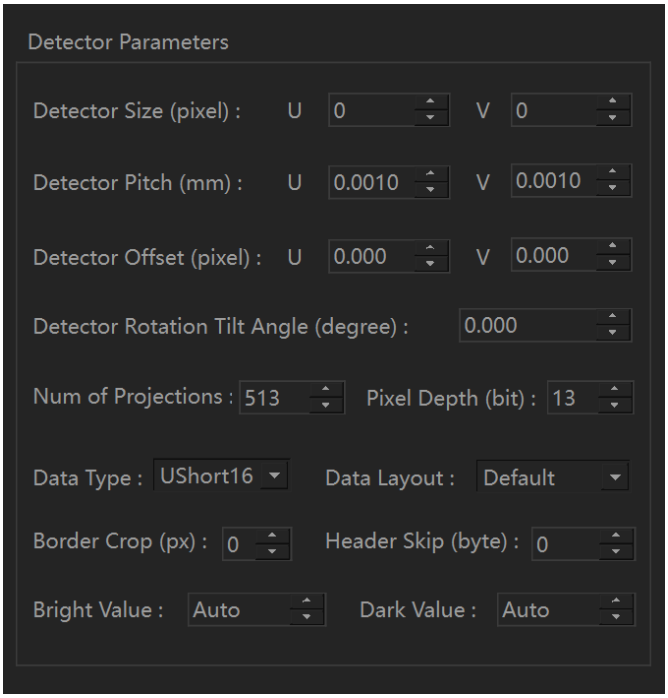
Choose either clockwise or counter-clockwise as the rotating direction of source and detector plate on ‘Rotation Direction’.

‘Tomography angle’ determines vertical movement of source and detector. For most cases, user doesn’t need to modify this value. (The value from project file will be used.)

For offset CT scan case (detector is shifted horizontally to shot bigger objects than detector size), please adjust ‘Detector Center Offset CT Mode’ slider to set the appropriate value.

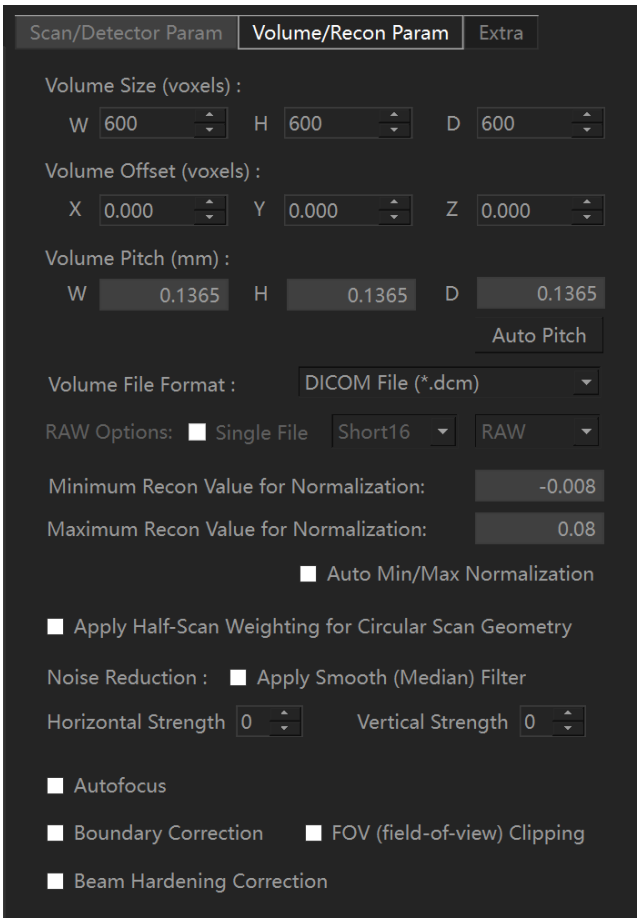


- Set the parameters related to the detector plate on 'Detector Parameters' panel.
- Possible parameters for setting are detector size in pixels, detector pitch in mm, horizontal/vertical offset in pixels, number of projections to be used for reconstruction process.
- It is also possible to specify the storage layout of projection image data. Use the 'Data Layout' option if projection data is stored in a non-standard fashion, e.g. horizontally flipped, vertically flipped, or rotated in 90 degrees.



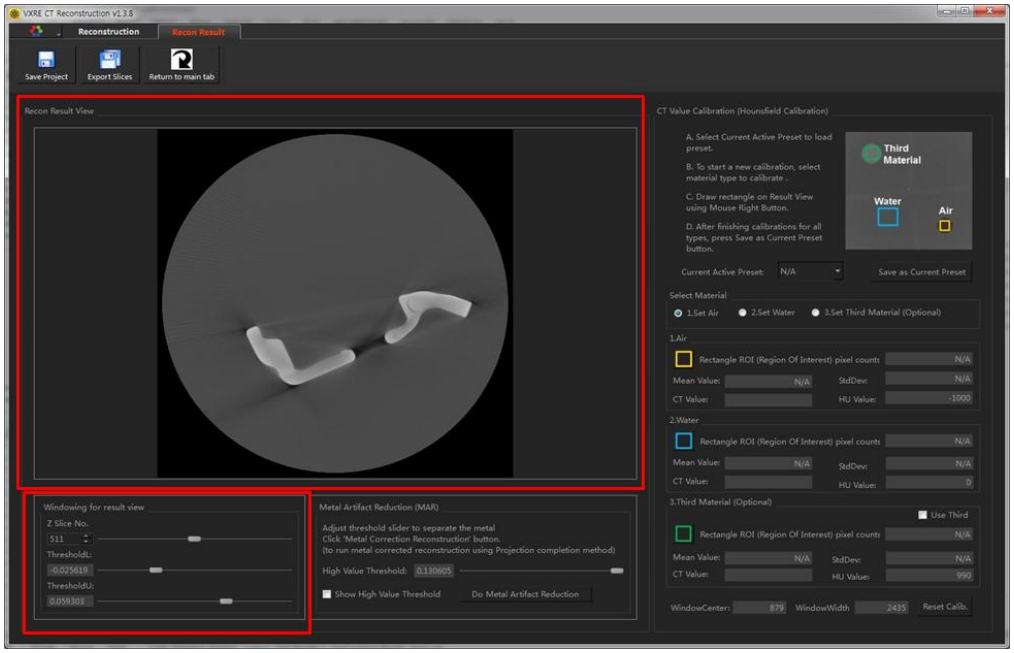
Main UI : Right Panel (Volume/Recon Parameters)

- Change the setting for reconstructed 3D volume such as volume size, pitch, data type and file format on 'Reconstruction Parameters' panel.
- Volume Pitch shows a size of single voxel. It can be set manually or automatically. (When click 'Auto Pitch' button, volume pitch values will be calculated and set automatically based on current FOV.)
- When 'Auto Min/Max Normalization' check box is checked, reconstructed volume's value will be normalized to minimum/maximum of the entire volume automatically. Otherwise, it will be normalized by the user-defined min/max value range.
- Smooth filter to reduce noises in projection images can be set. Check the 'Apply Smooth Filter' check box and set the filter kernel size. (e.g. when W=1 and H=1, the kernel size will be 3x3.)



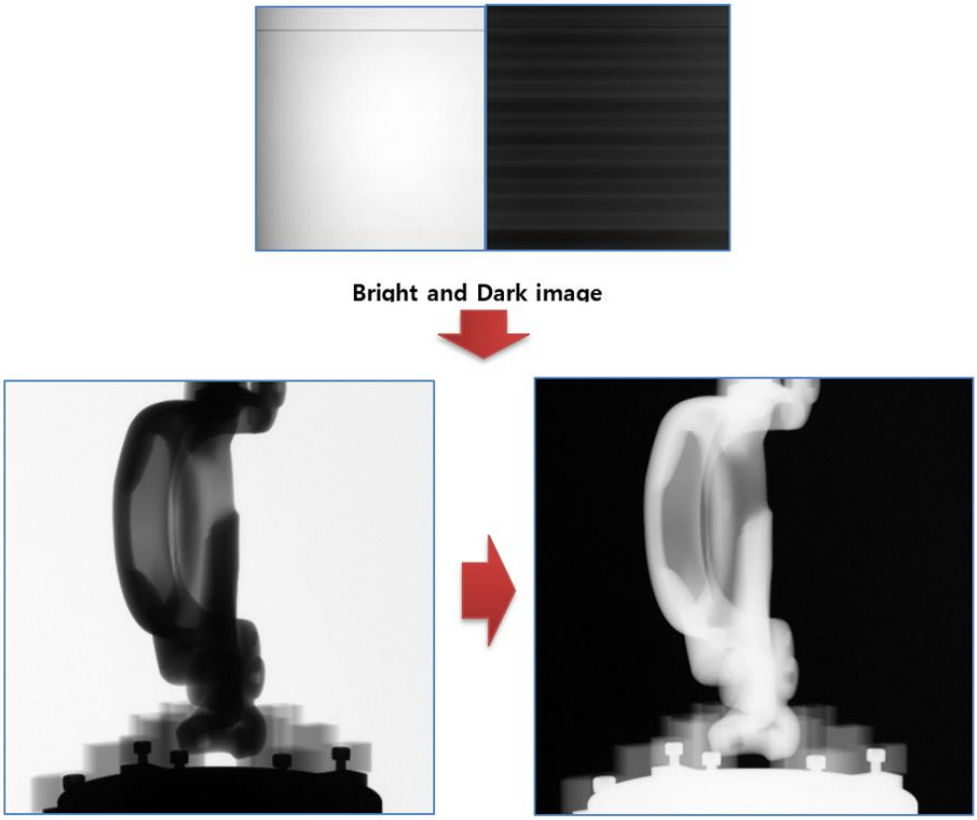
Main UI : Recon Result View

- After the reconstruction, 3D volume data will be shown in Recon Result View as 2D axial slices.
- Adjust 'Z Slice No.' slider to change current volume slice.
- Change slice bright/contrast by changing threshold sliders.



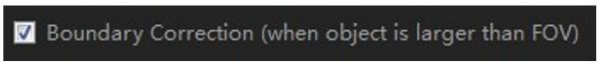
Basic Features

Dark/Bright Field Correction

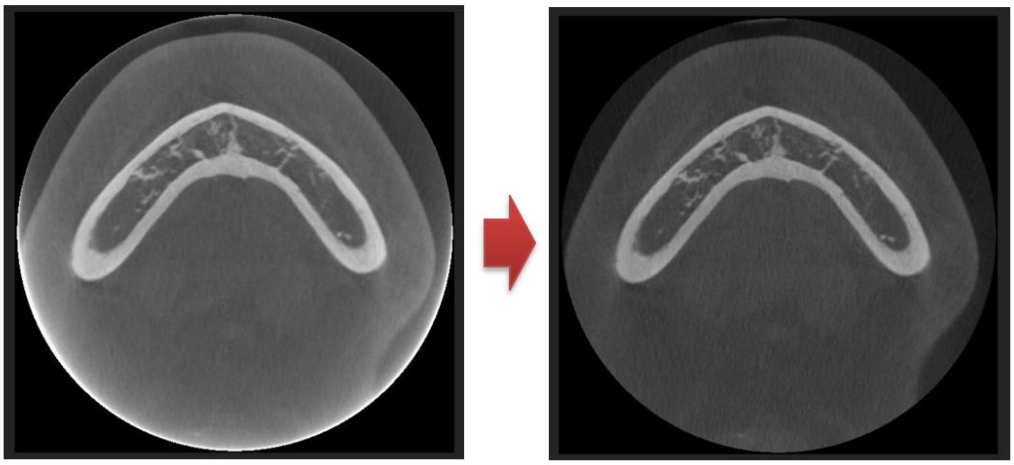


- When detector dark and bright field images are available, detector calibration (linearization) can be done using those images.
- Specify dark/bright file name inside project file (see Appendix) to apply detector calibration.

Outside FOV (Field Of View) Boundary Artifact Correction



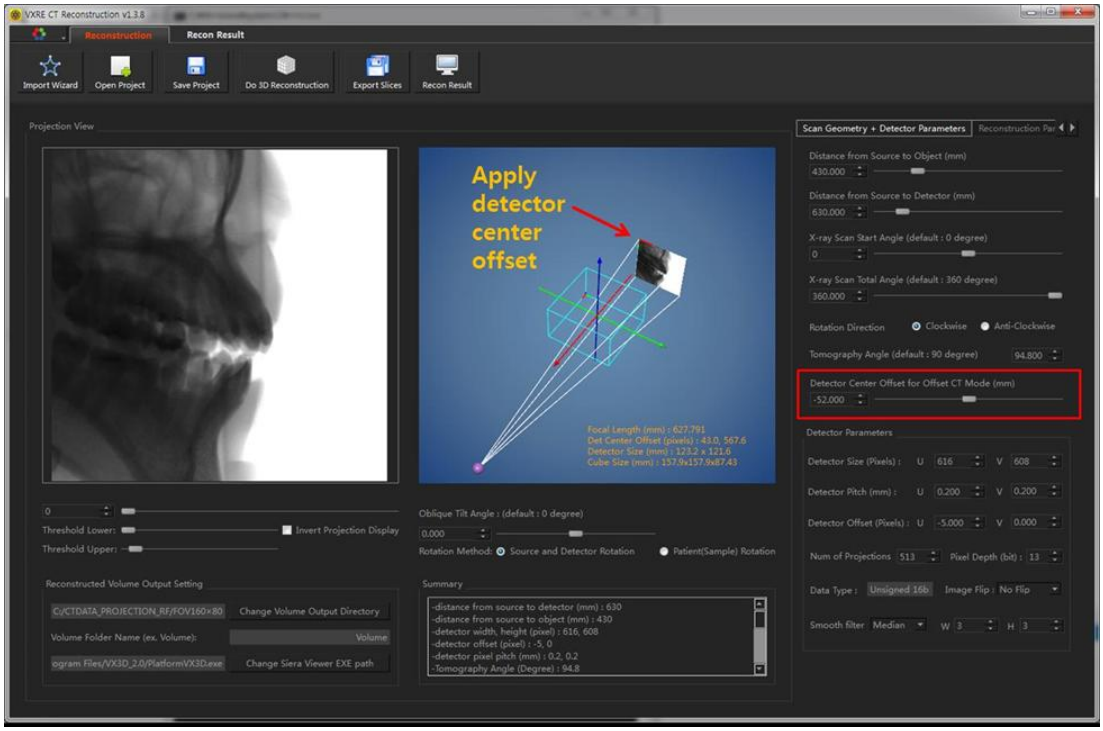
Boundary Correction Check Box



Before the correction **After the correction**

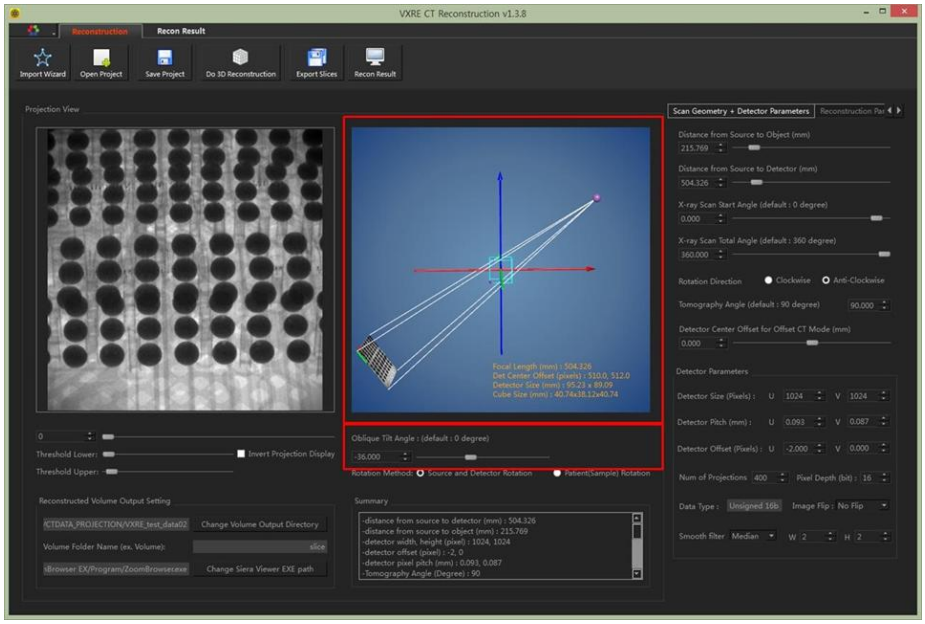
- When the object's size is larger than FOV (Field Of View) of CT scan geometry, artifacts will appear at the object boundaries.
- Set 'Boundary Correction' check box in Main UI's right panel to correct those artifacts.
- 'Boundary Correction' feature can slow down the reconstruction process, please turn on this feature only when the object is larger than FOV.

Offset CT Mode Support



- Offset CT mode scan support: adjust 'Detector Center Offset' slider in projection view to shot big size object using small size detector.

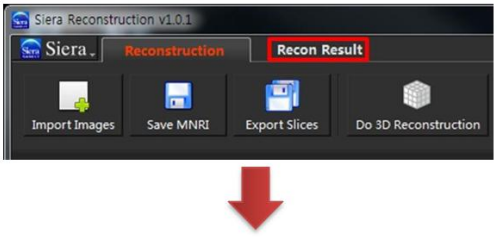
Oblique CT Mode Support

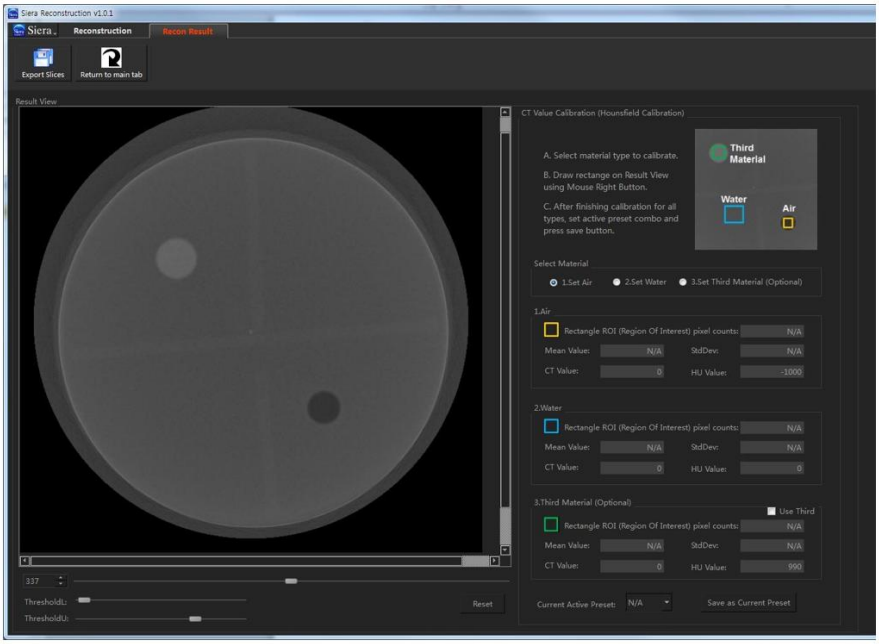


- For Oblique CT, adjust ‘Object Tilt Angle’ slider to set tilted angle value.
- Both of X-ray source and detector will be rotated together.

Advanced Features

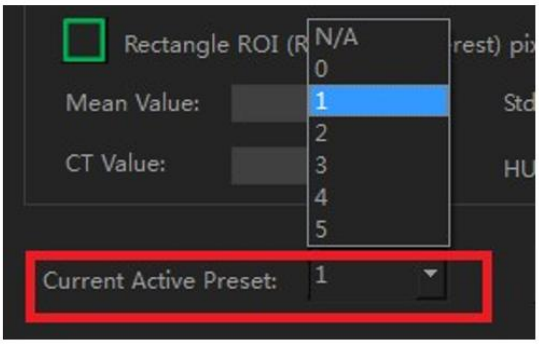
CT Value Calibration (Hounsfield Calibration)





CT Value (HU) Calibration

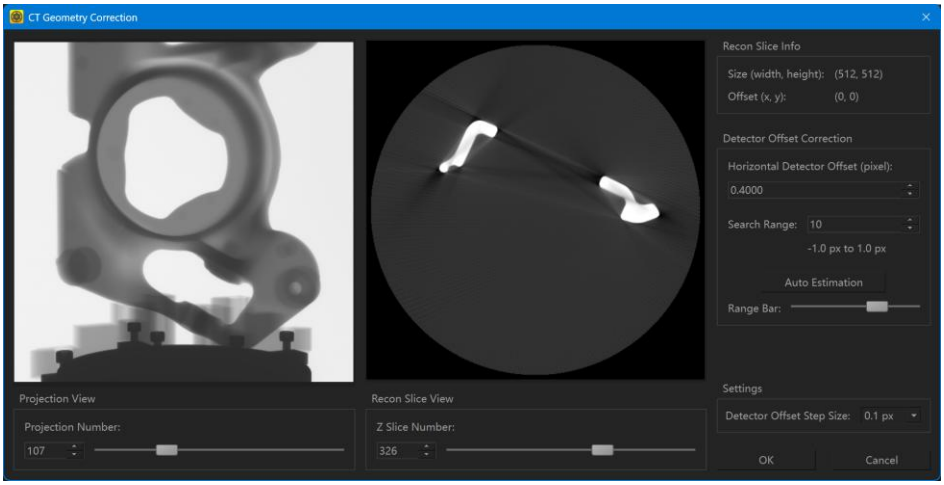
- After the reconstruction is finished, change the 'Reconstruction' view to 'Recon Result' view by clicking 'Recon Result' tab button.
- On Recon Result view, you can calibrate CT value to Hounsfield Unit value. For example, HU value is set as -1000 HU in the air region, and 0 HU in the water region.
- To calibrate the CT value to HU value, first choose 'Current Active Preset' by clicking the combo box. Currently six CT value presets are supported. If N/A is selected, calibration will not be applied.



Current Active Preset Combo Box

- To calibrate CT value for Air, press '1.Set Air' button and draw a rectangle on the desired air region using the right mouse button. This yellow rectangle represents the ROI (Region Of Interest) of air. The mean CT value of selected ROI will be calculated and set as CT Value automatically. It is also possible to modify CT value to any desired value.
- Repeat the same process for Water calibration. The blue rectangle represents the ROI of water. Its mean value will be set as CT Value as well.
- Calibration for any third material is optional.
- Click 'Save as Current Preset' button after calibration, to save the calibrated preset as current active preset. Whenever any current active preset is chosen among the combo box, its saved setting will be automatically loaded.
- Calibrated preset will be saved to the following path: 'C:\Users***\Documents\VXRE\CTValuePreset.ini'.

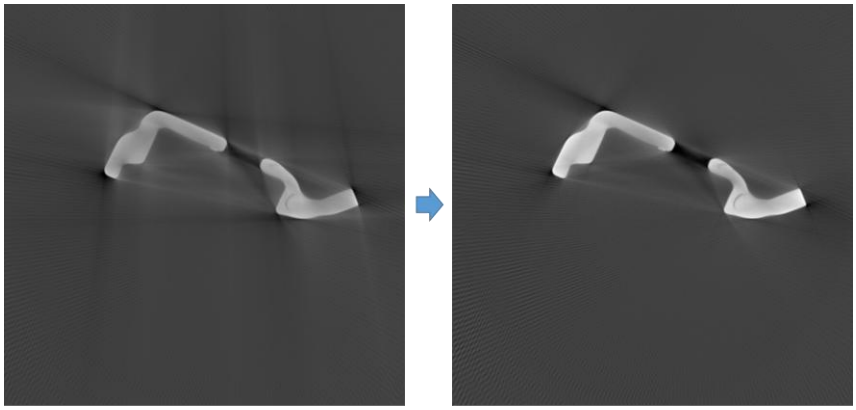
CT Geometry Correction (Horizontal Detector Offset)



- Click 'Geometry Correction' button in the top toolbar to open CT Geometry Correction dialog.

- Select projection image using Projection Number slider/spinbox to view detector positioning.
- Choose reconstruction slice using Z Slice Number slider/spinbox.
- Set Search Range parameter to define auto-estimation search window.
- Click ‘Auto Estimation’ button to automatically calculate optimal detector offset.
- Browse different offset results using Range Bar slider after the Auto Estimation.
- Fine-tune Horizontal Detector Offset manually if needed.
- Verify geometry corrections have improved image alignment and reduced artifacts.
- Click ‘OK’ to apply geometry correction parameter to reconstruction settings.

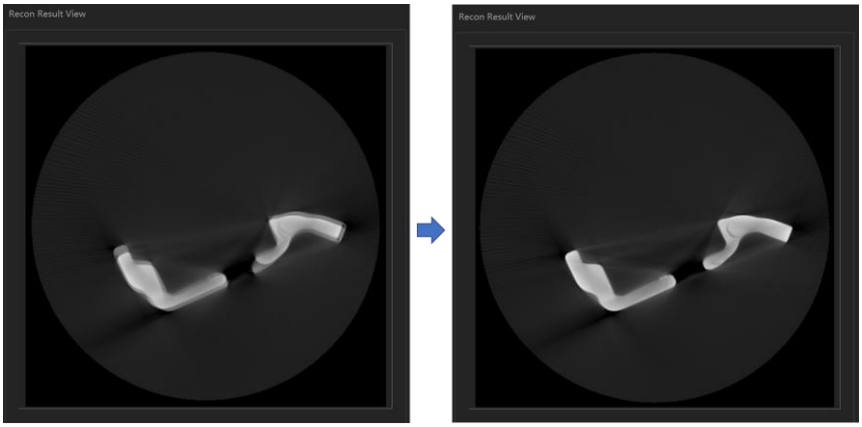
Half-Scan Weighting for Circular Scan Geometry



Without Half-Scan Weighting With Half-Scan Weighting

- In circular cone-beam CT geometry where the total range of scan angle is smaller than 360°, you can apply half-scan weighting to improve the reconstruction image quality.
- If the scan angle is smaller than 180° plus the fan angle or greater than 360°, the half-scan weighting cannot be applied and ignored if set.

Autofocus (Automatic Misalignment Correction)

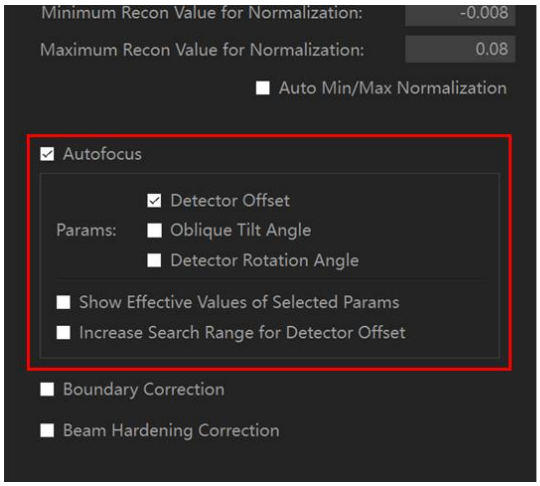


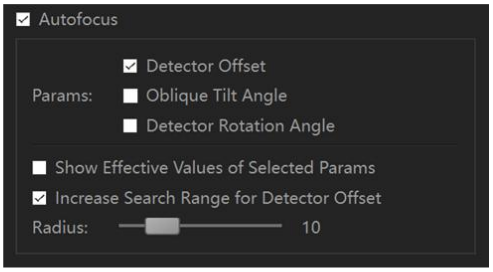
Without autofocus With autofocus

NOTE: *This feature requires the VXRE AB license.*

- The Autofocus feature provides a simple way to handle potential misalignment in scan geometry by enhancing the standard reconstruction with fully automatic geometry correction.

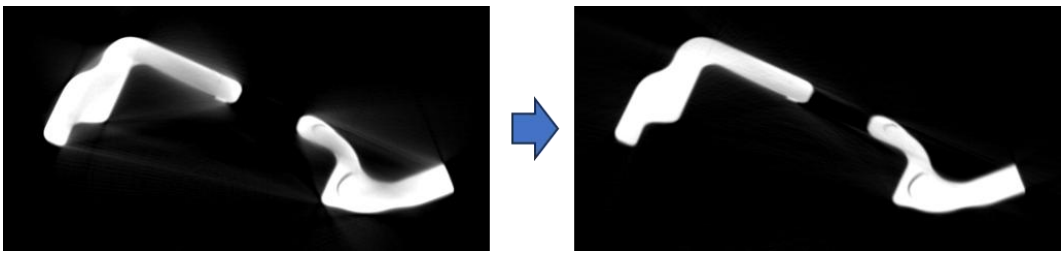
- To enable autofocus, check the option 'Autofocus' in the Volume/Recon Param tab. When it is checked, more options for Autofocus will appear (see below).





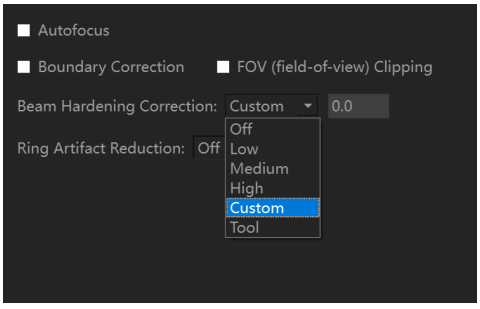
- Selecting more kinds of 'Params' in the Autofocus Options can result in more accurate, focused reconstruction images, at the expense of longer reconstruction time.
- When the option 'Increase Search Range for Detector Offset' is set, Autofocus searches the detector u-offset more thoroughly, the degree of which can be adjusted by
- Compared to the CT Geometry Correction tool, Autofocus allows more parameters to be optimized, with the optimization completely incorporated with the reconstruction step. In contrast, the CT Geometry Correction has limited support for parameter estimation (only horizontal detector offset in a cone-beam scan geometry).

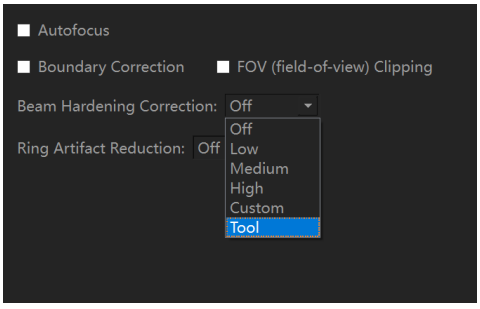
Beam Hardening Correction (BHC)



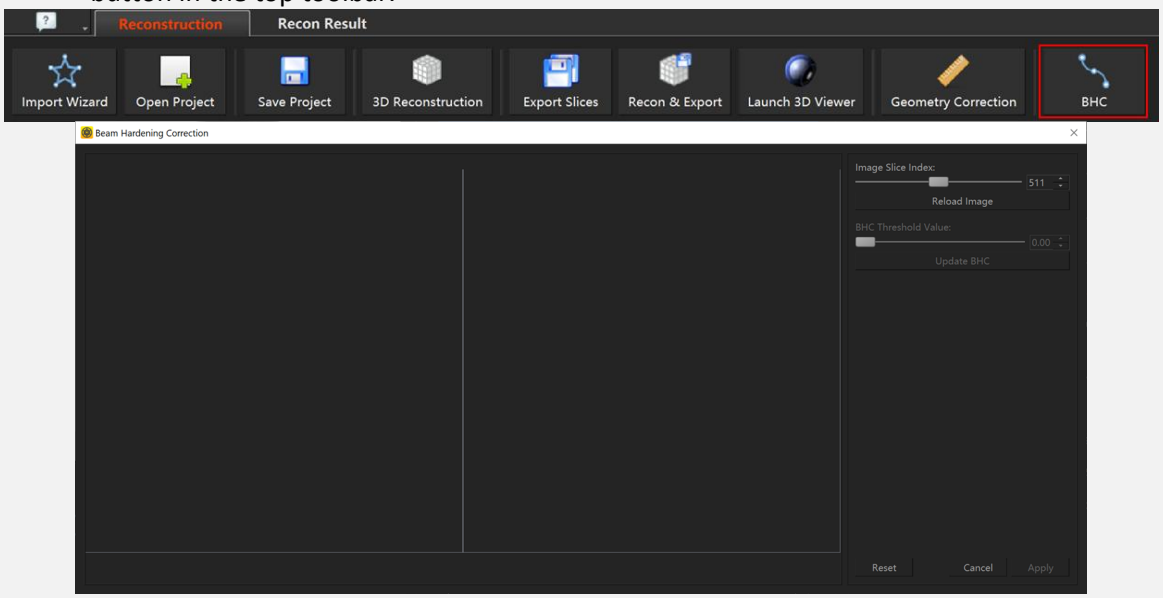
NOTE: *This feature requires the VXRE AB license.*

- The beam hardening correction (BHC) improves the uniformity of single-material objects in reconstruction images.
- To enable BHC, select the BHC mode (such as Low, Medium and High).
- To enable BHC with a custom strength, select the Custom mode and set the strength value (0.0=Off, 1.0≈Low, 2.0≈Medium, 3.0≈High).

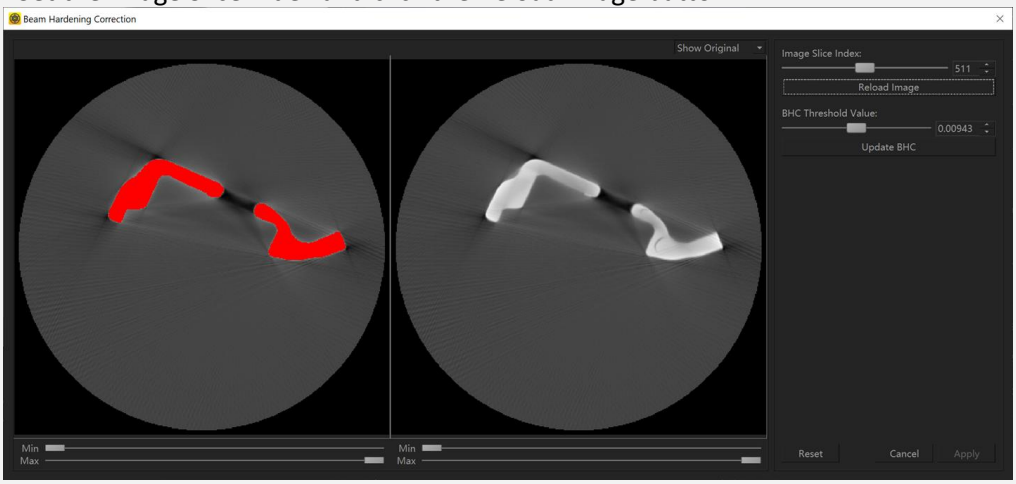




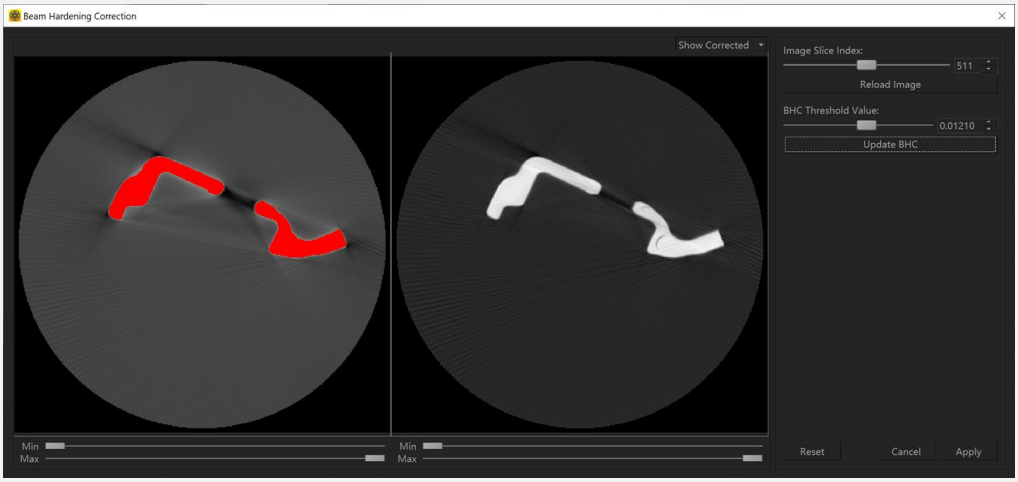
- To use the previous version of BHC (using the BHC Tool), select Tool or click the BHC button in the top toolbar.



- Set the Image Slice Index and click the Reload Image button.

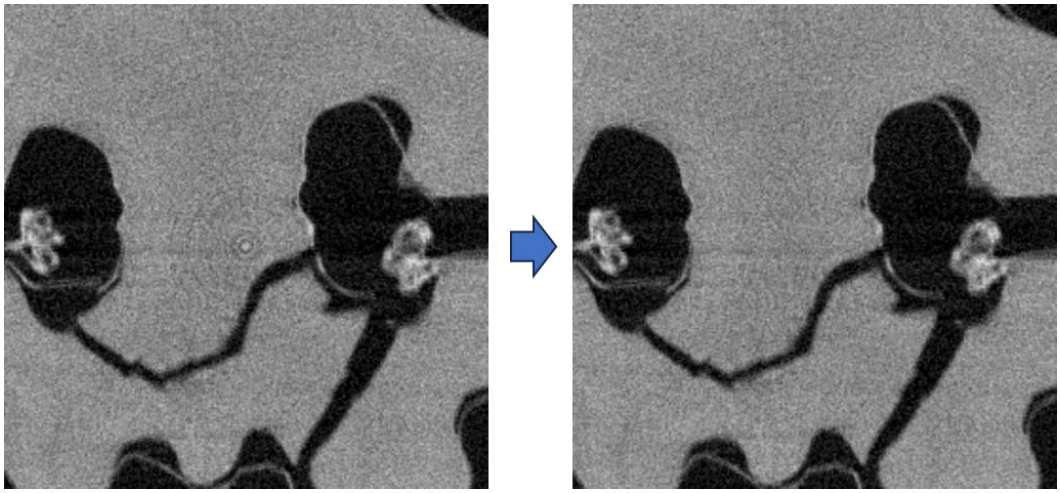


- Adjust the BHC Threshold Value to mark the object region in red. Then click the Update BHC button.

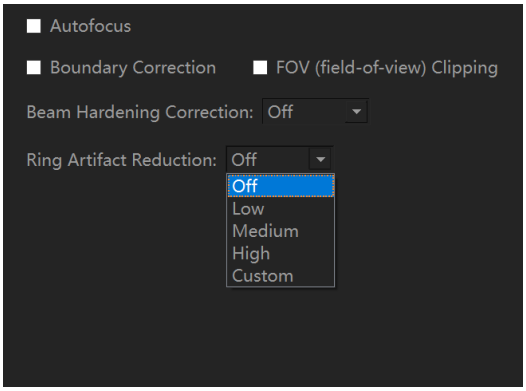


- Click the Apply button to use the current BHC settings for reconstruction.

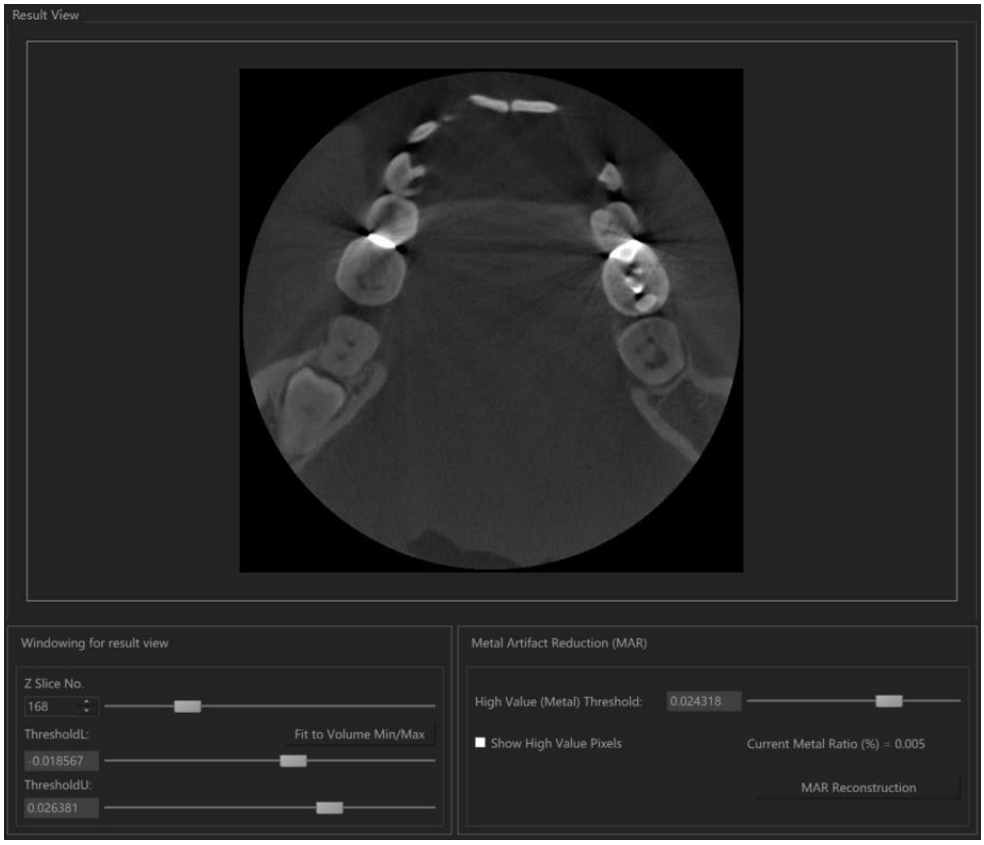
Ring Artifact Reduction (RAR)



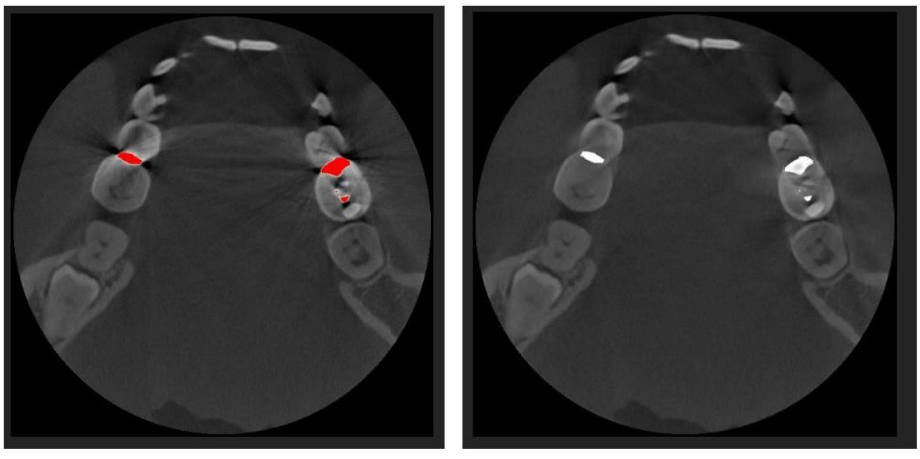
- The ring artifact reduction (RAR) alleviates circular patterns in reconstructed images due to non-ideal characteristics of detector cells.
- To enable RAR, select the RAR mode (such as Low, Medium and High).
- To enable RAR with a custom strength, select the Custom mode and set the strength value (0.0=Off, 1.0≈Low, 2.0≈Medium, 3.0≈High).



MAR (Metal Artifact Reduction) for Dental CT



- MAR (Metal Artifact Reduction) feature is for Medical/Dental CT.
- If the patient has dental filling or metal pins near the soft/bone tissue, reconstructed volume will show metal artifact near metal regions.
- After the initial (normal) reconstruction, 'Metal Artifact Reconstruction' feature will be enabled.
- In result view, check 'Show High Value Pixels' check box to visually display the metal region (The metal area will appear as red color in the result view). Adjust 'High Value (Metal) Threshold' slider to change the metal region.
- Click 'MAR Reconstruction' button to run metal-corrected reconstruction. Metal-corrected result will be shown in result view.

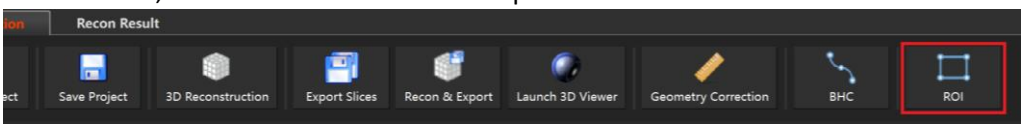


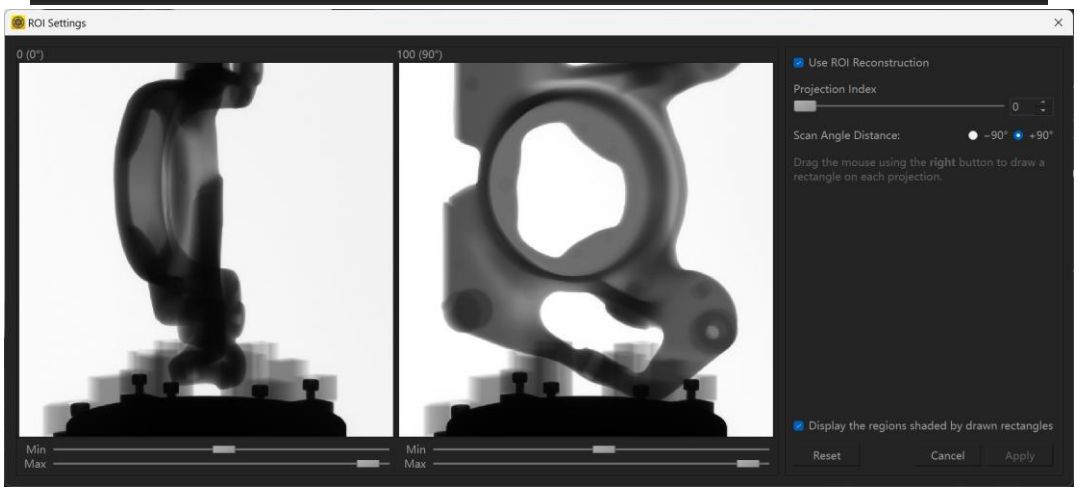
Left : Metal regions appear as red colr

Right : Metal corrected reconstruction result

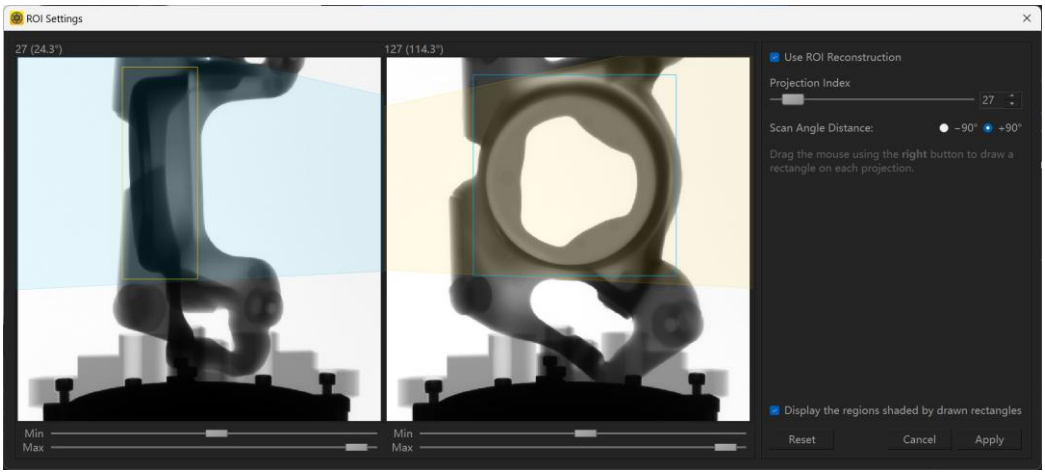
Projection-Based ROI Settings

- The ROI Settings tool provides an intuitive means of defining a 3D region of interest (ROI) based on a pair of projection views, which can then be used to specify the volume geometry for reconstruction.
- To define an ROI, click the ROI button in the top toolbar.





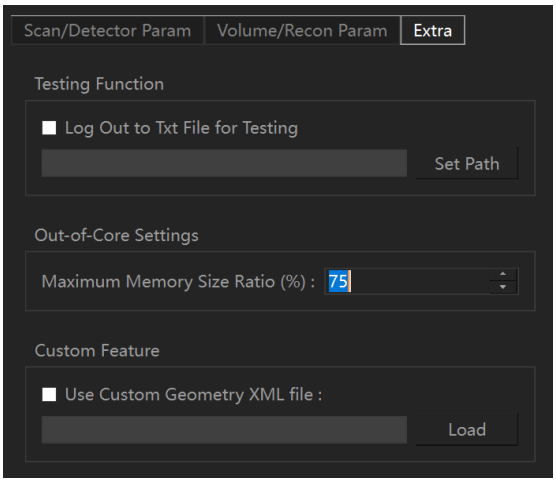
- Adjust the Projection Index and/or the Scan Angle Distance (if needed) to set the projection views aligned in a desirable way.
- Draw a rectangle around a region of interest on each projection by dragging the mouse with the right button.



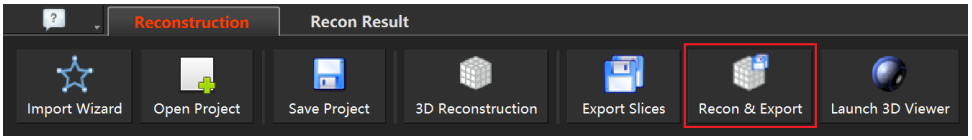
- Click the Apply button to use the defined ROI in reconstruction.
    - After an ROI is applied, it is also possible to disable ROI reconstruction and do the standard reconstruction (where the volume geometry is defined by the usual parameters: volume size, pitch & offset) by unchecking the Use ROI Reconstruction check box and clicking Apply in ROI Settings.
        - In this case, the ROI itself is preserved so it can be reused when ROI reconstruction is enabled again (by re-checking Use ROI Reconstruction in ROI Settings).
    - To remove the ROI and clear any changes in ROI Settings, click the Reset button and then Apply.

Out-of-Core Reconstruction (Large Volume)

- When the size of volume is larger than the memory space reserved for VXRE, the reconstruction is automatically performed in the out-of-core mode.
    - In the out-of-core mode, the entire volume will be divided into multiple subsets so that each subset data can be fitted into GPU memory and processed one-by-one.
- To control the size of the memory space reserved for VXRE, set the 'Maximum Memory Size Ratio' parameter in the 'Out-of-Core Settings'.
    - If there are issues with reconstruction, lower the parameter value.



- Alternatively, click 'Recon & Export' button in the top toolbar or use the -re_recon_save command on CLI to directly export the reconstruction results to disk. This can be used to further save the memory space needed for reconstruction.
    - The volume data will directly be written to the volume output folder in Output Setting.



Inline (On-the-Fly) Reconstruction

NOTE: *This feature requires the VXRE Fly license.*

- In contrast to standard reconstruction, which requires all projection files to be available at the start of reconstruction, inline (on-the-fly) reconstruction can process one projection file at a time sequentially whenever it becomes available.
- Inline reconstruction is available via CLI and VXRE Communication API (see each appendix for details).

Appendix : Project File Format

VXRE uses an XML document as its project file (encoded in UTF-8 with the .xml extension) describing various parameters for reconstruction. They are organized under a single root element (with the <VXRE> tag) as follows.

Supported Tags

<ScanGeometry>

- <SOD> : Distance from X-ray source to Object center (mm)
- <SDD> : Distance from X-ray source to Detector center (mm)
- <StartAngle> : Start angle (degree)
- <ScanAngle> : Total scan angle (e.g. 360 degree)
- <ProjCount> : Number of total projection images
- <ProjStart> : Index of first projection image (e.g. 0)
- <TomoTheta> : Tomography angle (e.g. 90 degree)
- <TiltAngle> : Oblique tilt angle (degree)
- <RotDirection> : Object (or source/detector) rotation direction (CW, CCW)
- <RotMethod> : 0=Source/Detector rotation, 1=Object rotation

<DetectorInfo>

- <DetSizeX> : Detector Width (pixel)
- <DetSizeY> : Detector Height (pixel)
- <DetPitchX> : Detector pitch in horizontal direction (mm)
- <DetPitchY> : Detector pitch in vertical direction (mm)
- <DetOffsetX> : Detector offset in horizontal direction (pixel)
- <DetOffsetY> : Detector offset in vertical direction (pixel)
- <DetRotAngle> : Detector rotation tilt angle (degree)
- <DetCenterOffset> : Detector center offset for half beam mode (mm)
- <ImgDataLayout> : Storage layout of projection data (default, {h,v,hv}flip, {ccw,cw}rot)
- <ImgCropBorders> : Crop projection image border pixels (pixel)
- <ImgSkipHeader> : Skip header in projection image (byte)
- <BitDepth> : Bit depth of the projection image
- <DetDarkValue> : Detector (Projection image) offset constant value
- <DetBrightValue> : Detector (Projection image) air constant value
- <DetBrightAuto> : ‘Air value’ will be estimated from the first projection image
- <ImgFormat> : Projection image data type (UShort16, Short16, Byte8, Float32)
- <ImgScaleValue> : This value will be multiplied when loading projection images

<VolumeInfo>

- <CubeSizeX> : Reconstruction volume width (voxel)
- <CubeSizeY> : Reconstruction volume height (voxel)
- <CubeSizeZ> : Reconstruction volume depth (voxel)
- <CubePitchX> : Reconstruction volume pitch in x direction (mm)
- <CubePitchY> : Reconstruction volume pitch in y direction (mm)
- <CubePitchZ> : Reconstruction volume pitch in z direction (mm)
- <CubeOffsetX> : Reconstruction volume offset in x direction (voxels)
- <CubeOffsetY> : Reconstruction volume offset in y direction (voxels)
- <CubeOffsetZ> : Reconstruction volume offset in z direction (voxels)

- <CubeFlipX> : Reconstructed slice flip along x direction (0 or 1)
- <CubeFlipY> : Reconstructed slice flip along y direction (0 or 1)
- <CubeFlipZ> : Reconstructed slice flip along z direction (0 or 1)

<ReconInfo>

- <PreProcessFilterType> : Smooth filter type for projection images (No, Median)
- <PreProcessFilterW> : Half kernel width for the projection smooth filter
- <PreProcessFilterH> : Half kernel height for the projection smooth filter
- <HalfScanWeighting> : Half-scan weighting for circular scan geometry (true, false)
- <BoundaryCorrection> : Boundary correction for outside FOV object (true, false)
- <FovClipping> : Clipping outside FOV (field-of-view) region (true, false)
- <ApplyROI> : Use ROI reconstruction (true, false)
- <ReconValueMinMaxAuto> : 3D will be normalize to volume min/max (true, false)
- <ReconValueMinValue> : Minimum recon value for normalization
- <ReconValueMaxValue> : Maximum recon value for normalization
- <VolumeFileFormat> : Volume file format (DICOM, RAW, TIFF, PNG, BMP, JPG)
- <RawVolumeSingleFile> : For RAW volume, save it as a single file (true, false)
- <RawVolumeImgFormat> : For RAW volume, volume data type (Short16, UShort16, Float32)
- <RawVolumeExportFormat> : For RAW volume, use VX3D-custom RAW format (DEN, MHD)
- <ExportSliceAxis> : Volume file slice orientation (XY, XZ, YX, YZ, ZX, ZY)
- <ExportSliceFlipX> : Volume file slice flip along x direction (0 or 1)
- <ExportSliceFlipY> : Volume file slice flip along y direction (0 or 1)
- <ExportSliceFlipZ> : Volume file slice flip along z direction (0 or 1)

<FilePath>

- <ProjFolder> : Folder path which contains projection images
- <SliceFolder> : Default name for folder where volume slice files will be written
- <ProjNameFormat> : Projection image name format (e.g. %03d.raw → 000.raw, 001.raw, …)
- <ProjIndexStep> : Projection index step (default: 1)
- <DarkFrameName> : File name of dark (offset) image for detector correction
- <BrightFrameName> : File name of bright (air) image for detector correction
- <CalibGeoFilePath> : Path to calibration geometry (.geo) file

<MAR>

- <UseMAR> : Use metal artifact reduction in reconstruction (true, false)
- <MetalThreshold> : High value (metal) threshold value

<Autofocus>

- <UseAutofocus> : Use autofocus for scan geometry misalignment correction (true, false)
- <FindDetOffsetX> : Include detector u-offset as autofocus parameter (true, false)
- <FindTiltAngle> : Include oblique tilt angle as autofocus parameter (true, false)
- <FindDetRotAngle> : Include detector rotation angle as autofocus parameter (true, false)
- <ScanRadiusDetOffsetX> : Radius of increased search range for detector u-offset

<BHC>

- <BHCMode> : Beam hardening correction mode (Off, Low, Medium, High, Custom, Tool)
- <BHCStrength> : Strength of beam hardening correction for Custom mode

<RAR>

- <RARMode> : Ring artifact reduction mode (Off, Low, Medium, High, Custom)
- <RARStrength> : Strength of ring artifact reduction for Custom mode

<CustomRecon>

- <ROIVolSizeX> : ROI reconstruction volume width (voxel)
- <ROIVolSizeY> : ROI reconstruction volume height (voxel)
- <ROIVolSizeZ> : ROI reconstruction volume depth (voxel)
- <ROIVolPitch> : ROI reconstruction volume pitch (mm)
- <ROIVolPitchX> : ROI reconstruction volume pitch in x direction (mm)
- <ROIVolPitchY> : ROI reconstruction volume pitch in y direction (mm)
- <ROIVolPitchZ> : ROI reconstruction volume pitch in z direction (mm)
- <ROIFilePath> : Add an ROI for reconstruction as defined by the ROI file
- <ZProjection> : Apply z-projection to each volume after reconstruction (No, Mean)
- <ROITracking > : Use ROI tracking (true, false)

<Settings>

- <CompatibilityMode> : VXRE compatibility mode (Standard, Legacy)
- <Profile> : GUID for specifying usage of specialized interface & behavior
- <VX3DAutoLaunch> : Launch 3D Viewer after reconstruction is complete (true, false)
- <VxredirPath> : Specify custom (absolute) path to .vxredir folder

Project File Example

```
<?xml version="1.0" encoding="UTF-8"?>
<VXRE>
    <ScanGeometry>
        <SOD>200.0</SOD>
        <SDD>800.0</SDD>
        <StartAngle>0.0</StartAngle>
        <ScanAngle>360.0</ScanAngle>
        ...
```

Appendix : Background Launch

To start a VXRE process in the background, run "vxre.exe -bg".

When launched this way, the main window will not be displayed and instead a tray icon will appear in the system notification area. Click the tray icon to see the main window.

Appendix : Command-Line Interface

VXRE provides a command-line interface (CLI) for controlling its behavior and operations, in addition to the standard graphical user interface. After a VXRE process has been started, you can control the running process by using vxrecli.exe as follows.

Synopsis

vxrecli [<command>] [<argument>...]

Commands

-re_close

Terminates the VXRE process.

-re_easy_mode_recon <project-file-path>

Invokes the commands -re_load_project, -re_recon and -re_save in sequence.

After that, VX3D is launched (if installed) to open the reconstruction volume.

Example

vxrecli -re_easy_mode_recon "C:\CTDATA\project.xml"

-re_inline_recon_start

NOTE: *This feature requires the VXRE Fly license.*

Initializes inline 3D reconstruction.

This command must be followed by multiple invocations of the -re_inline_recon_single command and then by -re_inline_recon_finish.

-re_inline_recon_single <proj-file-path>

NOTE: *This feature requires the VXRE Fly license.*

Performs inline 3D reconstruction with the specified projection file.

Example

vxrecli -re_inline_recon_single "C:\CTDATA\projection\0000.raw"

-re_inline_recon_finish

NOTE: *This feature requires the VXRE Fly license.*

Finalizes inline 3D reconstruction.

-re_load_project <project-file-path>

Opens a project file.

Example

vxrecli -re_load_project "C:\CTDATA\project.xml"

-re_recon

Performs 3D reconstruction.

-re_recon_mar

Performs 3D reconstruction using MAR.

-re_recon_mar_reuse

Performs 3D reconstruction using MAR, reusing the existing non-MAR recon volume.

-re_recon_save [<path>]

Performs 3D reconstruction and saves the result to disk, potentially in out-of-core mode if necessary. See '-re_recon' and '-re_save'.

-re_roi

Opens the ROI window.

-re_save [<path>]

Saves the reconstructed data to disk.

The reconstructed volume data is saved to disk. It will be saved as a set of slice files (or, depending on the Volume/Recon settings, as a single volume file).

The file format is determined by the Volume File Format option (the VolumeFileFormat XML tag). If the format is RAW, additional format options are available in the Volume/Recon Param tab (XML tags such as RawVolumeSingleFile, RawVolumeImgFormat, and RawVolumeExportFormat). Note that using MHD-based export format implies a single-file RAW output.

Options

<path>

Specifies (i) the path to the directory where slice files are to be saved, if the volume data is being saved as a set of slice files; or (ii) the single-volume file path, if the volume data is being saved as a single-volume file.

Relative paths are interpreted with respect to the project file location.

If omitted, the default path is determined based on a combination of the project file location, the project file name, and the slice folder name setting (the SliceFolder XML tag).

-re_set_cmdlog [<options>] [<cmdlog-file-path>]

Sets the command log file path for the current VXRE process.

VXRE determines the command log file path based on the following order of precedence, from highest to lowest:

- Path set by the -re_set_cmdlog command (per-process).
- Path set by the -re_log_path command (global).
- The default path on fresh installation (typically "C:\Users\%USERNAME%\Documents\communication.txt").

The per-process log file path is initially not set (cleared), and the command logs are written to the file as specified by the -re_log_path command.

When the per-process log file path (cmdlog-file-path) is set by this command, it takes precedence over the global log file path and the command logs will be written to that file. The global log file path is ignored but not discarded. To restore the usage of the global log file, clear the per-process log file path using the --reset option.

Options

--disable

Suppress the generation of command logs. The cmdlog-file-path argument is ignored.

--reset

Clear the current per-process log file path. The command log will then be written to the global log file (whose path is set by the -re_log_path command). The cmdlog-file-path argument is ignored.

This also disables the prefixing of timestamps to the log lines (if enabled previously).

--timestamp

Prefix timestamps to the command log lines. The form of a log line becomes '<date> <time> <command> <status>' (see Checking for Completion of Command subsection below for comparison).

Examples

vxrecli -re_set_cmdlog --timestamp "C:\cmdlog.txt"

vxrecli -re_set_cmdlog --disable

vxrecli -re_set_cmdlog --reset

-re_set_option <name> <value>

Changes the value of an internal option <name>, which can be used to control the behavior of the running VXRE process.

Recognized options are as follows:

communication.timestamp

Alias for cmdlog.timestamp (deprecated).

cmdlog.timestamp (default: off)

Specify whether to prefix timestamps to the command log lines. Setting this on has the same effect as using the --timestamp option for the -re_set_cmdlog command.

Valid values

off, on

Example

vxrecli -re_set_option cmdlog.timestamp on

ui.minimize_to_tray (default: off)

Specify whether minimizing the main window makes VXRE minimize to the system tray instead of the taskbar.

On background launch, the default value becomes on.

Valid values

off, on

Example

vxrecli -re_set_option ui.minimize_to_tray on

ui.minimize_to_tray_on_close (default: off)

Specify whether closing the main window makes VXRE minimize to the system tray instead of terminating it.

Valid values

off, on

Example

vxrecli -re_set_option ui.minimize_to_tray_on_close on

ui.show_progress (default:on)

Specify whether to display a progress bar for operations.

Valid values

off, on

Example

vxrecli -re_set_option ui.show_progress off

ui.show_cmd_progress

Alias for ui.show_progress (deprecated).

Deprecated Commands

-re_log_path <log-file-path>

NOTE: *This command is deprecated. Use the -re_set_cmdlog command instead.*

Sets the command log file path.

The default path on fresh installation is "C:\Users\<username>\Documents\communication.txt".

This setting is persistent and global; that is, the path will be used not only by the currently running VXRE process but also by future ones, and if multiple versions of VXRE are installed, all of them will use the same file path set by the -re_log_path command.

Example

vxrecli -re_log_path "C:\comm.txt"

-re_recon_save_dcm

NOTE: *This command is deprecated. Use the -re_recon_save command instead.*

Performs 3D reconstruction and saves the result as DICOM files.

-re_save_dcm

NOTE: *This command is deprecated. Use the -re_save command instead.*

Saves reconstructed data as DICOM files.

-re_save_raw_single [<option>] <raw-file-path>

NOTE: *This command is deprecated. Use the -re_save command instead.*

Saves reconstructed data to the specified path as a single raw file (16-bit integer type).

A signed integer type is used by default (use the -u option to change it to an unsigned type).

If a relative path is given, the path is interpreted with respect to the folder that contains the project file currently loaded.

Option

-u

Use an unsigned integer type for the raw file.

Example

vxrecli -re_save_raw_single "C:\CTDATA\volume.raw"

-re_save_raw_single_mhd [<options>] <raw-file-path>

NOTE: *This command is deprecated. Use the -re_save command instead.*

Saves reconstructed data as a single raw file accompanied by an .mhd file for VX3D.

The raw file is saved in accordance with the -re_save_raw_single command. You can open the .mhd file with VX3D to load and view the saved volume data.

Options

-u

Use an unsigned integer type for the raw file.

Example

vxrecli -re_save_raw_single_mhd "C:\CTDATA\volume.raw"

Checking for Completion of Command

If you send a command to VXRE via the command-line interface, the corresponding operation is performed and its completion status is written to the command log file (or appended at the end of the file, if it already exists). The path to the log file can be specified by the -re_set_cmdlog command.

The status is written as one log line in the form of '<command> <status>', where <command> is the requested command and <status> is the completion status ('true' if successful; 'false' otherwise). For example, if you run the -re_recon command, a new line containing '-re_recon true' will be appended to the log file upon successful completion of 3D reconstruction.

Appendix : VXRE Communication API

VXRE Communication API is a C/C++ library of types and functions that can be used to communicate with the running VXRE process.

To use it, take the following steps (The default %VXRE_Dir% is "C:\Program Files\VXRE").

- Include vxre\vxre_comm_api.h located in %VXRE_Dir%\include.
- Link with vxrecomm.lib located in %VXRE_Dir%\lib.
- Make sure your executable can locate vxrecomm.dll, which is distributed in %VXRE_Dir%\bin.
- See example codes located in %VXRE_Dir%\example.

API Changes

1.12.7

- Add VxreCommGetVolQueryDescription function

1.12.5

- Fix progress dialog not being closed after call to VxreCommSendProjReset in some cases

1.12.4

- Add VxreCommInvokeCliAsync function

1.12.3

- Add VxreCommGetVolQueryInfo function

1.12.1

- Make VxreCommSendProjReset capable of cancelling inline reconstruction

1.12.0

- Add handle-based APIs for volume data retrieval
  - VxreCommGetVolHandle structure
  - VxreCommGetVolOpenHandle function
  - VxreCommGetVolObtainData function
  - VxreCommGetVolCloseHandle function
- Add VxreCommStatus enumerator
  - VXRE_COMM_INDEX_OUT_OF_RANGE
- Ensure VxreCommInvokeCli to return appropriate error code

1.11.4

- Add VxreCommGetVolOpen, VxreCommGetVolClose functions
- Add VxreCommGetVolData structure
- Add is_vol_data_available member variable to VxreCommHostStatus structure
- Add VxreCommStatus enumerators
  - VXRE_COMM_CONNECTION_SETUP_ERROR
  - VXRE_COMM_CONNECTION_TIMEOUT_ERROR
  - VXRE_COMM_DATA_UNAVAILABLE
  - VXRE_COMM_ALREADY_OPENED
- Improve robustness of communication with VXRE

1.11.2

- Add proj_index parameter to VxreCommSendProj function

1.10.11

- Add VxreCommInvokeCli function
- Add VXRE_COMM_CLI_ERROR to VxreCommStatus enumeration
- Add VxreCommGetHostStatus function
- Add VxreCommHostStatus structure
- Add VxreCommSendProjReset function
- Improve stability of projection data transfer

1.10.10

- Initial release

1.10.9

- Initial beta

General

VxreCommHostStatus structure

Contains information about the current status of VXRE.

```
typedef struct
{
    bool is_recon_running;
    bool is_vol_data_available;
} VxreCommHostStatus;
```

is_recon_running

If true, this indicates the reconstruction is currently ongoing in VXRE.

is_vol_data_available

If true, this indicates the volume data reconstructed by VXRE is available for retrieval.

VxreCommGetHostStatus function

Retrieves the current status of VXRE.

```
VxreCommStatus VxreCommGetHostStatus(VxreCommHostStatus *status);
```

status

A pointer to a VxreCommHostStatus structure that receives status information.

VxreCommInvokeCli function

Invokes a CLI command for VXRE and waits for the command to complete.

```
VxreCommStatus VxreCommInvokeCli(const wchar_t *cmdargs);
```

This function returns after VXRE receives the CLI command and finishes running the command.

If the command completes successfully, VXRE_COMM_OK is returned. If the command completes unsuccessfully, VXRE_COMM_CLI_ERROR is returned.

Use VxreCommInvokeCliAsync if you need to return immediately after the invocation of a command.

cmdargs

A pointer to a null-terminated wide string that contains the command-line argument to vxrecli (see Appendix : Command-Line Interface for reference). For example,

```
VxreCommInvokeCli(LR"(-re_load_project "C:\CT DATA\project.xml")");
...
```

VxreCommInvokeCliAsync function

Invokes a CLI command for VXRE and returns immediately.

```
VxreCommStatus VxreCommInvokeCliAsync(const wchar_t *cmdargs);
```

This function returns immediately after VXRE receives the CLI command, not waiting for it to finish running the command.

Use VxreCommInvokeCli if you need to wait for the completion of a command.

cmdargs

A pointer to a null-terminated wide string that contains the command-line argument to vxrecli (see Appendix : Command-Line Interface for reference).

Projection Data Transfer

NOTE: *This feature requires the VXRE Fly license.*

VxreCommSendProjDesc structure

Specifies options for projection data transfer.

```
typedef struct
{
    bool inline_recon;
    bool no_wait_for_recon;
} VxreCommSendProjDesc;
```

It is recommended that a VxreCommSendProjDesc structure is always zero-initialized to its default value before individual members are set. In C++, for example,

```
VxreCommSendProjDesc desc = {};
desc.inline_recon = true;
...
```

inline_recon

If true, reconstruction is performed (inline) along with projection data transfer. Each projection image is used, as soon as it is received, for reconstruction by VXRE. The default is false.

no_wait_for_recon

With inline_recon enabled, specifies the return behavior of the VxreCommSendProjFinalize function. If set to false (default), VxreCommSendProjFinalize returns after both data transfer and reconstruction are complete. If true, it returns immediately when the transfer is done, without waiting for the reconstruction to finish. Ignored when inline_recon is false.

VxreCommSendProjInit function

Initializes projection data transfer to VXRE.

```
VxreCommStatus VxreCommSendProjInit(const VxreCommSendProjDesc *desc);
```

Call this function to initiate projection data transfer for the currently loaded XML project file.

desc

A pointer to a VxreCommSendProjDesc structure that specifies options for projection data transfer. If nullptr, the default options (see VxreCommSendProjDesc) are used.

VxreCommSendProj function

Sends a projection image to VXRE.

```
VxreCommStatus VxreCommSendProj(int proj_index, const void *proj_data,
                                size_t proj_data_size);
```

The projection image to be sent is determined as proj_data_size bytes from the memory area pointed by proj_data.

This function must be called after projection data transfer has been initialized by VxreCommSendProjInit. It is intended to be called multiple times for each projection image, in the order of the projections, with proj_index being $0, 1, \ldots, N-1$ (N is the number of projections).

proj_index

The index of the projection image within the array of the entire projection images.

proj_data

A pointer to the memory area containing the projection image to be sent.

proj_data_size

The number of bytes to be sent. This must be equal to sizeof(<ImgFormat>) * <DetSizeU> * <DetSizeV> (see Appendix : Project File Format for description of the corresponding XML tags).

VxreCommSendProjFinalize function

Finalizes projection data transfer to VXRE.

```
VxreCommStatus VxreCommSendProjFinalize(void);
```

This function must be called after all projection images have been sent by VxreCommSendProj.

In inline-recon mode, it is possible to call this function either synchronously or asynchronously, depending on the value of no_wait_for_recon (see VxreCommSendProjDesc). To abort (cancel) an ongoing asynchronous reconstruction, use the VxreCommSendProjReset function.

VxreCommSendProjReset function

Resets projection data transfer to the uninitialized state.

```
VxreCommStatus VxreCommSendProjReset(void);
```

This function clears up any internal states in VXRE concerning projection data transfer. If inline-recon mode is set, this also stops any ongoing reconstruction.

After the call to this function, VxreCommSendProjInit can be called for a fresh start of projection data transfer.

Call this if an error occurs during projection data transfer to make sure the failed projection data transfer is properly terminated.

There is no need to call this function under normal circumstances. Calling this function after VxreCommSendProjFinalize returns VXRE_COMM_OK is unnecessary and redundant (but allowed).

Volume Data Retrieval

VxreCommGetVolData structure

Contains information about the volume data.

```
typedef struct
{
    int dim_x;
    int dim_y;
    int dim_z;
    float pitch_x;
    float pitch_y;
    float pitch_z;
    const float *raw_ptr;
} VxreCommGetVolData;
```

The value of any voxel in the volume can be accessed via raw_ptr. For example, to get the value of the voxel whose index is (x, y, z),

```
VxreCommGetVolData data;
...
size_t nx = data.dim_x;
size_t nxy = nx * data.dim_y;

int x; // = 0, 1, ..., data.dim_x - 1
int y; // = 0, 1, ..., data.dim_y - 1
int z; // = 0, 1, ..., data.dim_z - 1
...
float val = data.raw_ptr[nxy * z + nx * y + x];
```

The memory area pointed by raw_ptr is read-only; writing to it results in undefined behavior.

dim_x
The number of voxels in the x-direction.

dim_y
The number of voxels in the y-direction.

dim_z
The number of voxels in the z-direction.

pitch_x
The length of voxel spacing in the x-direction.

pitch_y
The length of voxel spacing in the y-direction.

pitch_z
The length of voxel spacing in the z-direction.

raw_ptr

A pointer to the read-only memory area containing the voxel values. The values are organized in row-major order.

VxreCommGetVolHandle structure

Holds handle for volume data retrieval.

```
typedef struct VxreCommGetVolHandleType_ *VxreCommGetVolHandle;
```

A VxreCommGetVolHandle is created by VxreCommGetVolOpenHandle and destroyed by VxreCommGetVolCloseHandle.

VxreCommGetVolQueryDescription function

Gets descriptive name of reconstruction volume.

```
VxreCommStatus VxreCommGetVolQueryDescription(wchar_t **description_out);
```

This function retrieves a descriptive name for the volume. The name is returned as a pointer to a null-terminated wide string. The string is intended to be accessed read-only and its memory and lifetime are managed by VXRE.

description_out

A pointer to a pointer to a null-terminated wide string that receives a desciptive name for the volume.

VxreCommGetVolQueryInfo function

Gets information about reconstruction volume.

```
VxreCommStatus VxreCommGetVolQueryInfo(int vol_index,
                                       VxreCommGetVolData *vol_data_out);
```

This function retrieves the information about the volume, including the number of voxels and the lengths of voxel spacing. It is unnecessary for the volume data to be actually reconstructed in VXRE. The returned VxreCommGetVolData has its raw_ptr always set to nullptr.

vol_index

The index of the volume whose information is to be received.

vol_data_out

A pointer to a VxreCommGetVolData structure that receives the information about the volume.

VxreCommGetVolOpen function

Obtains the reconstructed volume data from VXRE.

```
VxreCommStatus VxreCommGetVolOpen(VxreCommGetVolData *vol_data);
```

This function retrieves the latest volume data that has been successfully reconstructed in VXRE. The returned volume data includes voxel values as well as the number of voxels and the lengths of voxel spacing.

When done with using the volume data, call VxreCommGetVolClose on vol_data to release any system resources that may have been internally acquired by VXRE. For example,

```
VxreCommStatus status;

VxreCommGetVolData data;

status = VxreCommGetVolOpen(&data);
if (status != VXRE_COMM_OK) exit(1);

... // Use VxreCommGetVolData.

status = VxreCommGetVolClose(&data);
if (status != VXRE_COMM_OK) exit(1);
```

vol_data

A pointer to a VxreCommGetVolData structure that receives the volume data.

VxreCommGetVolClose function

Relinquishes the obtained volume data.

```
VxreCommStatus VxreCommGetVolClose(VxreCommGetVolData *vol_data);
```

This function must be called for each volume data obtained from VxreCommGetVolOpen after using the volume data is done and it is no longer needed.

After the call, accessing the memory area once pointed by raw_ptr in vol_data results in undefined behavior.

vol_data

A pointer to a VxreCommGetVolData structure that specifies the volume data to be relinquished.

VxreCommGetVolOpenHandle function

Opens a handle for retrieval of volume data from VXRE.

```
VxreCommStatus VxreCommGetVolOpenHandle(VxreCommGetVolHandle *vol_handle);
```

This function prepares for retrieval of volume data from VXRE. The handle for the retrieval is returned as vol_handle.

Call VxreCommGetVolObtainData on vol_handle to actually retrieve the volume data. When done with retrieving volume data, call VxreCommGetVolCloseHandle on vol_handle.

During a handle is open, any operation that causes VXRE to modify the volume data results in undefined behavior. Close the handle first for VXRE to safely perform such an operation.

vol_handle

A pointer to VxreCommGetVolHandle that receives a handle for volume data retrieval.

VxreCommGetVolObtainData function

Obtains the reconstructed volume data via a handle for volume data retrieval.

```
VxreCommStatus VxreCommGetVolObtainData(VxreCommGetVolHandle vol_handle,
                                        int vol_index,
                                        VxreCommGetVolData *vol_data_out);
```

This function retrieves the latest volume data that has been successfully reconstructed in VXRE. The returned VxreCommGetVolData includes voxel values as well as the number of voxels and the lengths of voxel spacing.

If in Custom Recon mode, set vol_index to a zero-based index to specify from which volume to obtain the volume data. Otherwise, set vol_index to 0.

```
VxreCommStatus status;

VxreCommGetVolHandle handle;
VxreCommGetVolData data;

status = VxreCommGetVolOpenHandle(&handle);
if (status != VXRE_COMM_OK) exit(1);

for (int vol_index = 0; vol_index < vol_count; ++vol_index)
{
    status = VxreCommGetVolObtainData(handle, vol_index, &data);
    if (status != VXRE_COMM_OK) exit(1);

    ... // Use VxreCommGetVolData for current vol_index.
}

status = VxreCommGetVolCloseHandle(handle);
if (status != VXRE_COMM_OK) exit(1);
```

vol_handle
A handle for volume data retrieval.

vol_index
The index of the volume from which the volume data retrieval is to be performed.

vol_data_out
A pointer to a VxreCommGetVolData structure that receives the volume data.

VxreCommGetVolCloseHandle function

Closes a handle for volume data retrieval.

```
VxreCommStatus VxreCommGetVolCloseHandle(VxreCommGetVolHandle vol_handle);
```

This function must be called for a handle opened by VxreCommGetVolOpenHandle after obtaining the volume data is done.

After the call, accessing the memory area once pointed by raw_ptr in VxreCommGetVolData returned by VxreCommGetVolObtainData results in undefined behavior.

vol_handle
A handle for volume data retrieval to be closed.