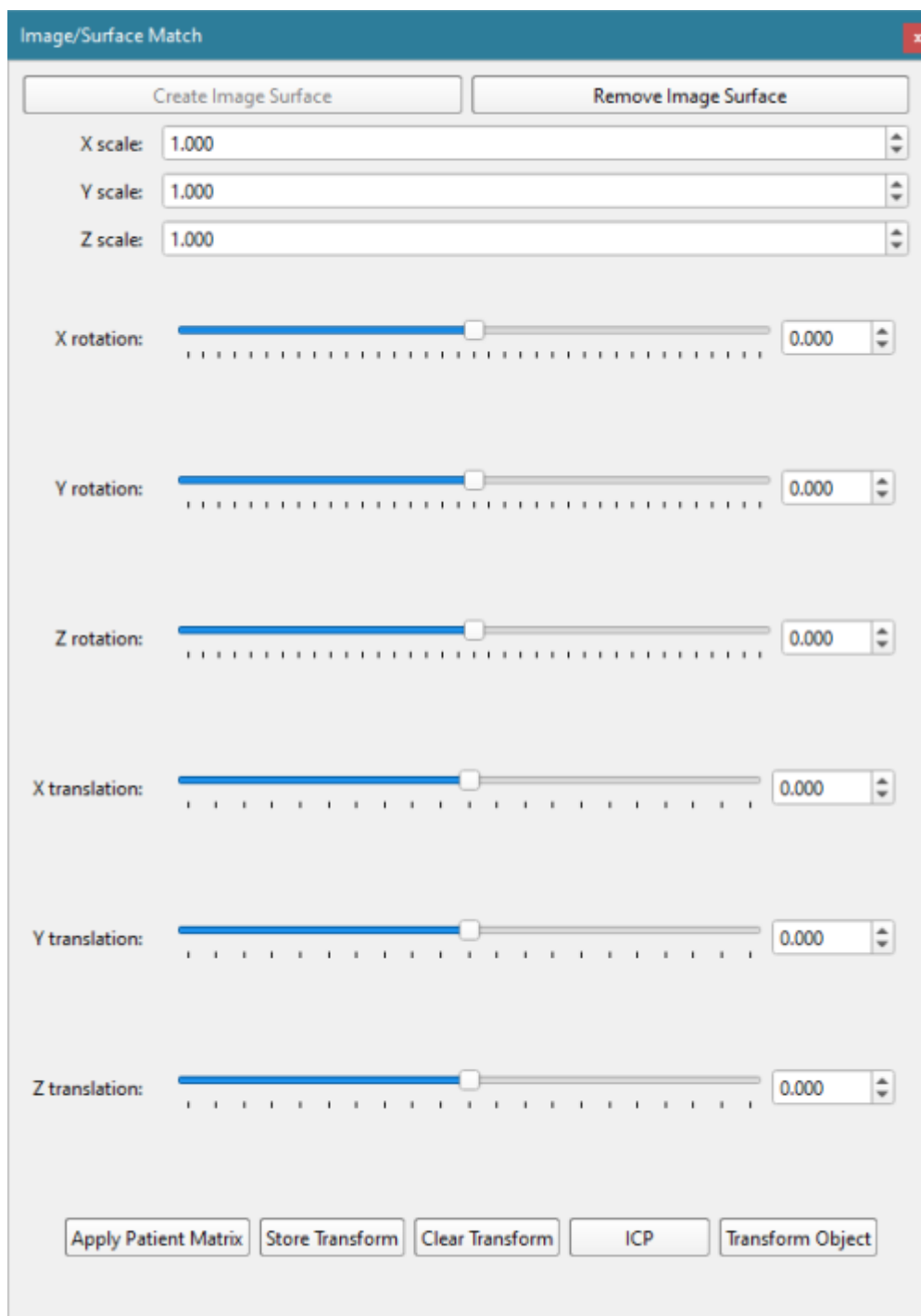


Image-Surface Match

The Image-Surface Match widget lets you check if an object's surface model is in the same reference frame as its image data. This is important in X4D so that the surface models are aligned with the DRRs. It works by creating an image surface from the image data, and displaying it in the 3D view with the object's surface model. If they are not aligned, you can use the widget controls to scale, rotate, and translate the image surface so that it aligns with the surface model. Then you can apply the inverse of this transformation to the surface model, so that it aligns with the initial pose of the image surface.



The screenshot shows the 'Image/Surface Match' widget interface. It features a title bar with a close button. Below the title bar are two buttons: 'Create Image Surface' and 'Remove Image Surface'. The main area contains six rows of controls for X, Y, and Z scale, rotation, and translation. Each row has a slider with a blue bar and a numerical input field. The X, Y, and Z scale fields are set to 1.000. The X, Y, and Z rotation fields are set to 0.000. The X, Y, and Z translation fields are set to 0.000. At the bottom, there are five buttons: 'Apply Patient Matrix', 'Store Transform', 'Clear Transform', 'ICP', and 'Transform Object'.

- **Create Image Data** creates a partially transparent surface (the image surface) directly from

the object's image data and adds it to the 3D view. It loads the image file specified in the object definition in the loaded subject file. A surface model is created by running a marching cubes algorithm on the data, which is assumed to be segmented such that voxels not in the object of interest are zero. Before the surface is displayed it is smoothed and decimated.

- **Remove Image Surface** removes the image surface from the 3D view, which should be done after you have aligned it with the surface model and transformed the model.
- **Scale, Rotation, Translation** These number fields and sliders define the transformations that are applied to the image surface. They are applied in the order in which they are listed in the widget, and the rotations are body-fixed.
- **Apply Patient Matrix** applies the patient matrix, as specified for the object in xManager, to the surface model. This can be helpful if you are using third-party software to segment CT data and create surface models. For example, if the segmented object model is stored in the DICOM format, you may need to apply the patient matrix in the DICOM to the surface model in order to align them. The specific details of this process depend on how you export the Image File and Surface file from the third-party software.
- **Store Transform** builds a 4x4 transform from the current scale, rotation, and translation values, and stores it internally. It then resets the scales, rotations, and translations to their initial values. The total transform that is applied to the image surface is this stored transform followed by the current scale, rotation, and translation values. Storing the transform and then specifying additional manipulations of the image surface allows you to perform more complicated transformations, such as an X rotation followed by a Y rotation followed by another X rotation. If you store the transform when there is already one stored, the new one is concatenated onto the stored one.
- **Clear Transform** resets the stored transform to the identity matrix.
- **ICP** This command runs an Iterative Closest Point (ICP) algorithm to try to automatically align the image surface with the object's surface model.
- **Transform Object** This command applies the inverse of the total transform to the object's surface model. The total transform is the stored transform followed by the scale, rotation, and translation specified by the number fields and sliders. The stored transform is then reset to the identity matrix, and the scales, rotations, and translations are reset to their initial values. After transforming the object you should save it to a file with *File → Save Surface (global frame)*.

Iterative Closest Point Algorithm

Orient3D's Iterative Closest Point (ICP) algorithm starts with the surfaces in their current positions, then translates the image surface so its centroid coincides with the object surface's centroid, then runs the ICP algorithm. This algorithm often fails to align the surfaces if a rotation of more than 60 degrees is required, so for best results you should manually align them more closely than that before running the algorithm.

Note: Another limitation of ICP is that it usually doesn't produce perfect alignment because it operates on the vertices of the surfaces, and the surfaces usually have different vertex sets. Because of this, the pose that the ICP algorithm calculates is output to the scale, rotation, and translation sliders in the widget. Thus you can manually adjust the values to achieve better/perfect alignment before transforming the object.

See Also

Go back to the [Orient3D Overview](#).

From:

<https://has-motion.com/wiki/> - **Wiki Documentation Site**

Permanent link:

https://has-motion.com/wiki/doku.php?id=other:dsx:orient3d:image-surface_match

Last update: **2025/05/06 17:26**

