
Introduction to ITK Resample In-Place Image Filter

Release 0.00

Wei Lu¹, Hans Johnson²³

October 28, 2010

¹Department of Electrical and Computer Engineering, The University of Iowa

²Department of Psychiatry, The University of Iowa

³Department of Biomedical Engineering, The University of Iowa

Abstract

In this document we present an ITK [1] image filter that outputs a new image with identical voxel contents but with modified physical space representation. The physical space representation of an image is modified to reflect the physical space transform on the input image. The advantage of this filter is that it is free from interpolation error.

This paper is accompanied with the source code, input data, parameters and output data that the authors used for validating the algorithm described in this paper. This adheres to the fundamental principle that scientific publications must facilitate reproducibility of the reported results.

Latest version available at the [Insight Journal](http://hdl.handle.net/10380/3224) [<http://hdl.handle.net/10380/3224>]
Distributed under [Creative Commons Attribution License](#)

Contents

1 Principle	2
2 Validation	3
3 Usage	4
4 Acknowledgement	4

The current ITK resample image filter transforms, interpolates, and resamples an input image I into a new voxel lattice and physical space specified by the programmer. The newly created output image updates the voxel contents with interpolated values from the input image I .

A side effect of this process is that it introduces interpolation error, i.e., even given the inverse of the nontrivial input transform and the resampled image, it is generally impossible to obtain the original image. The problem can be even worse if several of resample filters are being used to generating intermediate output images in the entire process.

This filter takes advantage of the fact that the voxel lattice to physical space transform is represented as a rigid transform in ITK. If all the transforms to be applied are rigid, then by composing the transforms before hand, we can achieve the desired physical space representation without accumulating interpolation errors.

1 Principle

An ITK image class has a rigid transform representation that maps a physical space coordinate system onto the discrete voxel lattice. The physical space representation includes the orientation matrix of the image \mathbf{R}_0 , the origin vector of the image \mathbf{T}_0 , and the spacing vector of the image \mathbf{S}_0 . Thus each ITK image \mathbf{I} can be represented as a composition of an image \mathbf{I}_0 with isotropic spacing, zero origin, and identity orientation, and an affine transform \mathbf{A} by

$$\mathbf{I} = \mathbf{A}\mathbf{I}_0 \quad (1)$$

Where

$$\begin{aligned} \mathbf{A} &= \mathbf{T}_0\mathbf{R}_0\mathbf{S}_0 \\ &= \begin{bmatrix} \mathbf{R}_0\mathbf{S}_0 & \mathbf{T}_0 \\ & 1 \end{bmatrix} \end{aligned} \quad (2)$$

Recall that a rigid transform matrix \mathbf{M} is composed of a translation matrix \mathbf{T} and a rotation matrix \mathbf{R} , i.e.,

$$\begin{aligned} \mathbf{M} &= \mathbf{TR} \\ &= \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ & 1 \end{bmatrix} \end{aligned} \quad (3)$$

The output image \mathbf{I}_1 can be obtained by

$$\mathbf{I}_1 = \mathbf{MI} = (\mathbf{MA})\mathbf{I}_0 \quad (4)$$

Therefore, we have the composed image info matrix

$$\begin{aligned} \mathbf{MA} &= \mathbf{TRT}_0\mathbf{R}_0\mathbf{S}_0 \\ &= \begin{bmatrix} (\mathbf{RR}_0)\mathbf{S}_0 & \mathbf{RT}_0 + \mathbf{T} \\ & 1 \end{bmatrix} \end{aligned} \quad (5)$$

By contrasting **Equations 2** and **5** we have

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{RT}_0 + \mathbf{T} \\ \mathbf{R}_1 &= \mathbf{RR}_0 \\ \mathbf{S}_1 &= \mathbf{S}_0 \end{aligned} \quad (6)$$

Where \mathbf{T}_1 , \mathbf{R}_1 , and \mathbf{S}_1 are the output image origin vector, orientation matrix, and spacing vector, respectively.

2 Validation

Assume we have an image I_1 , which is centered at the origin of an LPS coordinate system, and a rigid transform T that will first translate the image 300 mm along $-P$ -axis, and then a 0.5 rad rotation along $-L$ -axis about the origin. The mid-sagittal plane of both the input image and the output image I_2 from the *itkResampleInPlaceImageFilter* are shown in the same physical space but lie in their own index space in **Figure 2**. Also, we have labeled a distinctive landmark (*ac*) for each image (*i.e.*, the point A in I_1 and the point B in I_2). Their locations in index space and physical space are listed in **Table 1**.

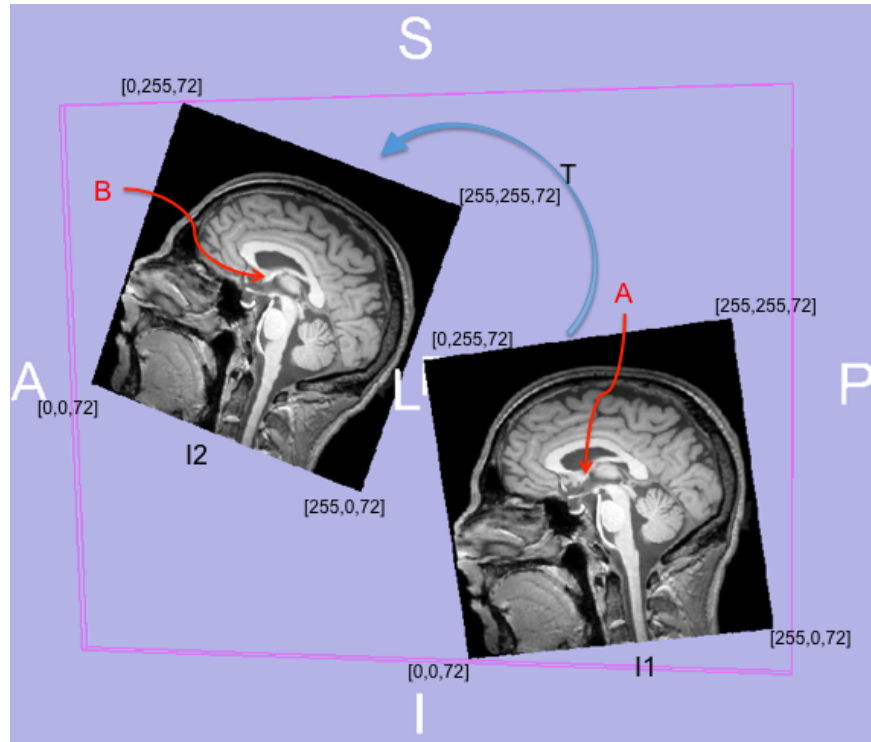


Figure 1: Input image I_1 and the transformed output image I_2 are shown in the same physical space but lie in their own index space. The index extent are labeled in black for each image.

	Index space	Physical space
Input (A)	[119, 141, 72]	(0, 29.0, 19.4)
output (B)	[119, 141, 72]	(0, 280.3, 174.8)

Table 1: This table shows the ac locations in input/output and index/physical space. The output image has the same voxel contents as that of the input image but with different physical representation due to the impact of an input rigid transform.

3 Usage

First include the header file by

```
1 #include "itkResampleInPlaceImageFilter.h"
```

Define the resample in-place image filter by

```
1 typedef itk::ResampleInPlaceImageFilter< ImageType, ImageType > FilterType;
```

Set up the filter and process the input volume

```
1 FilterType::Pointer filter = FilterType::New();  
2 filter->SetInputImage( inputImage );  
3 filter->SetRigidTransform( transform );  
4 filter->Update();  
5 ImagePointer outputImage = filter->GetOutput();
```

4 Acknowledgement

This research was supported by funding from grants NS050568 and NS40068 from the National Institute of Neurological Disorders and Stroke and grants MH31593, MH40856, from the National Institute of Mental Health.

References

- [1] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, <http://www.itk.org/ItkSoftwareGuide.pdf>, second edition, 2005. ([document](#))