Generating a Deformation Field from an `itk::Transform`

Release 0.00

Marius Staring[1]

June 17, 2008

[1]Leiden University Medical Center, The Netherlands

Abstract

This document describes a new class that takes an `itk::Transform` and evaluates the deformation at points in a user-defined region. The class is implemented using the Insight Toolkit `www.itk.org`. 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.

Contents

1 Introduction

In the field of medical imaging it is often necessary to analyse the deformation field resulting from a (non-rigid) registration process. The result of registration in ITK is however a transformation and not directly the deformation field.

This contribution provides a class that takes the transformation and produces a deformation field. The user needs to specify were and with what spacing the transformation should be evaluated. This is similar to the `itk::ResampleImageFilter`.

2 Implementation

Our class, which we named `TransformToDeformationFieldSource`, inherits from `itk::ImageSource`. It is implemented as a multi-threaded filter. Similar to the `itk::ResampleImageFilter`, advantage is taken in case the transformation is linear.

For every point $\boldsymbol{x}$ in the user-specified region we compute the transformed point $\boldsymbol{T}(\boldsymbol{x})$, and compute the deformation $\boldsymbol{d}$ at that point:

$$\boldsymbol{d}(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x}) - \boldsymbol{x}. \tag{1}$$

3 Usage

The filter is used as follows:

```
typedef itk::TransformToDeformationFieldSource<
  DeformationFieldImageType,
  CoordRepresentationType >            DeformationFieldGeneratorType;

TransformType::Pointer transform
  = TransformType::New();
// .. define the transform (set the parameters, etc.)

DeformationFieldGeneratorType::Pointer defGenerator
  = DeformationFieldGeneratorType::New();
defGenerator->SetOutputSize( size );
defGenerator->SetOutputSpacing( spacing );
defGenerator->SetOutputOrigin( origin );
defGenerator->SetOutputIndex( index );
defGenerator->SetOutputDirection( direction );
defGenerator->SetTransform( transform );
try
{
  defGenerator->Update();
}
catch ( itk::ExceptionObject & e )
{
  // Print error message
}
```

The `size, spacing, origin`, etc. parameters are user-defined. After generation of the deformation field, it can for example be written to file, or further analysis can be applied.