Statistical Simulation of Deformations (SSD)

Section of Biomedical Image Analysis (SBIA)
Department of Radiology
University of Pennsylvania
Oct 26, 2005

1. Overview

This document describes how to use the "Statistical Simulation of Deformations (SSD)" software package. The related papers of SSD are:

[1]. Submitted paper: Zhong Xue, Dinggang Shen, Bilge Karacali, Joshua Stern, David Rottenberg, Christos Davatzikos, Simulating Deformations of MR Brain Images for Validation of Atlas-based Segmentation and Registration Algorithms, submitted to *NeuroImage*, 2005

[2]. Zhong Xue, Dinggang Shen, Bilge Karacali and Christos Davatzikos, Statistical Representation and Simulation of High-Dimensional Deformations: Application to Synthesizing Brain Deformations, in the *Proceedings of MICCAI 2005*, Palm Springs, California, Oct 26 to Oct 29, 2005, J. Duncan and G. Gerig (Eds.), LNCS 3750, pp. 500-508.

The purpose of SSD is to first estimate the statistical models of deformation fields and then using these models to simulate new deformation fields and respective images. Since the deformations of the simulated images are known, they can be used for evaluating various registration and atlas-based segmentation algorithms.

The SSD software package includes the following three modules:

(1). The model training module, *i.e.* training the statistical models from a number of input sample deformation fields;

(2). The random simulation module, *i.e.* generating randomly new deformation fields and respective images using the statistical models trained in module (1).

(3). The Landmark-Constrained SSD simulation module, *i.e.* generating a deformation field that not only matches the prescribed landmark map but also reflects the statistics defined by SSD.

The rest of this document is organized as follows. Section 2 describes how to install the SSD package, and Section 3 describes the detailed usage of the SSD package. The summary and brief descriptions of each program of the SSD package can be found in the document of the SSD source codes. Section 4 gives a quick example of the processes described in this document. After installing the SSD package, please first have a try according to Section 4, and then extensive experiments can be performed by referring to other sections in this document.

2. Installation of the SSD software package

If you see this document, you already installed the SSD package successfully. ☺

The SSD software package is organized in the following folders:

```
bin         : this is where the executables are
sample        :  this  is  where  the  sample  files  are  (need  to  download
ssd_samples.tar.gz)
doc         : documentation folder
dat         : some useful point data
temp        : this is an empty temp folder
```

After checking each of the above folders, please change you system PATH environment so that the “bin” folder above is included in the PATH.

The sample data are in ./sample/ folder. Read the readme.txt file in the sample folder and follow the instructions.

The source codes in src folder are for reference only. A detailed list of the source codes and corresponding executables can be found in the document of the SSD source codes. You can now study Section 3 in detail or go to Section 4 to follow the quick example.

3. Usage of the software

3.1. Prior to use the software package

Please be familiar with the following terms and files prior to using the software package:

i) The template and subject images

All the images are with 1-byte analyze format. All the orientations of the template image and the subject images are the same. Fig. 1 shows the coordinates of a 3D MR brain image.

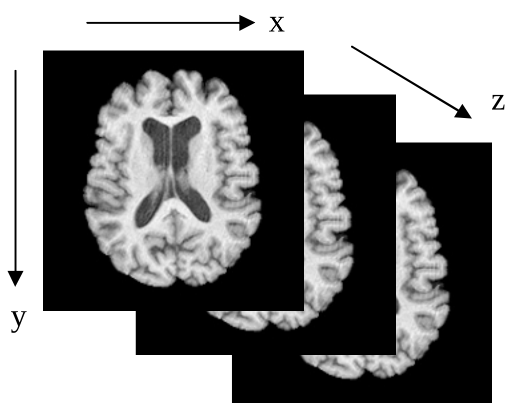

Fig.1 Coordinates of the 3D MR brain images.

The number of voxels along x direction is X, and along direction y is Y, and along direction z is Z respectively. Thus the size of the image is X*Y*Z, and each voxel is stored using 1-byte, and the size of the image file is X*Y*Z bytes.

ii) Segmented image

The segmented image labels the original image into three tissues. It has the same size with the original image, and each voxel is stored using 1-byte analyze format. The intensities (values) of the tissues are:

```
White Matter (WM) --- 250
Gray Matter (GM) --- 150
CSF --- 10
Background --- 0
```

In addition, the Ventricle area can also be labeled with intensity 50.

This definition of the segmented images is compatible with all the programs in the SSD package, and with all the programs of the HAMMER package (Note SSD and HAMMER also work if the label of ventricle area is marked as CSF).

iii) Deformation field

A deformation field defines the voxel-wise displacement (or offset) field from one image (the template image) to another image (subject image). That is, the deformation field defines an offset vector for each voxel of the template image.

One deformation field is represented by one file. It stores all the voxel-wise offsets using "float" and binary format. If the template image is with the size of X*Y*Z, the size of the deformation field is X*Y*Z*3*sizeof(float).

Denoting the offset of voxel (x, y, z) as (dx, dy, dz) (please see Fig.1 for definition of the coordinates x, y and z), in the deformation field file generated by HAMMER, for each voxel, the sequence of storing the offset is dy → dx → dz. While someone prefers to use the sequence dx→dy→dz.

Therefore, please use the argument "-S" to indicate that the deformation field is stored in the same way as HAMMER.

Please refer to the functions read_deformationfield() and write_deformationfield() in xzdgshen.c for details.

iv) Training samples of deformation fields

A number of sample deformation fields have to be available prior to training the SSD software package. The requirements of these sample deformation fields are as follows:

- Each deformation field defines the deformation from the template image to each individual subject image that has been globally registered with the template image (i.e. after affine-transformation). The reason to use the deformation field after global transformation is that global transformation is not supposed to be trained in the statistical models.

- A text file will be used to summarize all the deformation fields used as the training samples, the format of this text file is as follows:

  Number of Files
  Filename1
  Filename2
  …
  FilenameN

  One just needs to manually edit/modify this file to indicate the training sample deformation fields before training the statistical models. Ideally, it is better that the number of training samples is large, but 30-40 training samples should be sufficient to demonstrate good simulation performance of the SSD.

3.2. Construct the statistical models

The procedure to construct the statistical models is as follows:

- Run script: **LinuxTrainSSD.pl**

- One can get the command line help by running the above script without any options.

  The usage is list as follows:

```
This program trains the statistical models of SSD, including W-PCA
model of deformations and W-PCA model of Jacobians

Usage: LinuxTrainSSD.pl [options]

-h                      : help

-T tempalte             : the template image
-L filenamelist         : the text file that list all the filenames
of all the sample deformation fields.
-D X,Y,Z                : the size of input template image, or the
size of all the training deformation fields
-M WPCAName             : the base (prefix) name of the output W-PCA
models
-S                      : use this only if the input deformation
fields are generated by HAMMER
-P percent              : the percentage of the total variance
```

```
-O summarize_file      : output the summarize file that will be used
by LinuxPerformSSD.pl

-J                     : optional, use this option if you already
calculated the Wavelet-Packet-Transform of deformation fields.

Example:

LinuxTrainSSD.pl -T template_orig.img -L filenamelisttest.txt -D
256,256,181 -M PCA8 -S -P .98 -O train8.txt
```

This program will generate all the files of the statistical models (filename start with PCA39). File train39.txt is the summary file of the training stage.

The process of the above program includes (please refer to the comments marked within the program **LinuxTrainSSD.pl**):

- Topology correction of the input deformation fields if their Jacobian determinants are not positive;
- Calculate the Jacobian determinants of all the (topology-corrected) input deformation fields;
- Calculate the Wavelet Packet Transform (WPT) of all the deformation fields and all the Jacobian determinants;
- Calculate the Wavelet-PCA model of deformation fields and Wavelet-PCA model of Jacobian determinants.

After running **LinuxTrainSSD.pl,** a lot of files were generated in the folder, they include:

- The W-PCA model files, all the files whose names start with the model prefix name: WPCAName
- The WPT files of the deformation fields, with name "fieldname.jak.wpt"
- The WPT files of the Jacobian determinants of deformation fields, with name "fieldname.wpt"

After training the SSD, the following programs (Section 3.3 and Section 3.4) can be used to simulate new deformation fields and images.

3.3. Randomly simulate deformation fields using the SSD

This process requires that the statistical models have been properly trained (see 3.2 in detail). The procedure to randomly generate a simulated deformation field is described as follows:

- Run script **LinuxPerformSSD.pl**

- One can get the command line help by running the above script without any options. The usage is list as follows:

```
This program simulates one new deformation and respective image using
the SSD

Notice that the statistical models have to be trained using
LinuxTrainSSD.pl prior to using this program

Usage: LinuxPerformSSD.pl [options]

-h                    : print this help

Inputs:

-T tempalte_image     : the template image
-t template_segment   : the segmented image of the template image
-M basename for Model: the base(prefix) filename of the WPCA models
-S summary_file       : the summary file generated by LinuxTrainSSD

-m alpha,beta         : alpha determines how far the random sample from
the origin in the EigenSpace, if alpha is large, the random samples are
far from the origin, and thus yield large deformations. beta indicates
how large is the variance of the random samples. Alpha and beta are
optional, default values are alpha = 0, beta = 1.

Outputs:

-F simulated_field    : the output simulated deformation field
-f inverse_field      : the inverse deformation field of the output
simulated deformation field, (optional)
-O simulated_image    : the output simulated image
-o simulated_seg      : the output simulated segmention image (optional)

-Q                    : quick simulation, without regularization of
deformation field. (This option is for quick simulation, and the
topology of the simulated deformation is not guaranteed to be correct.)

Example:

LinuxPerformSSD.pl -T template_orig.img -t template_segmented.img -M
PCA8 -F sim.field -O sim.img -S train8.txt -o sim.seg
```

The above program performs:

- Randomly generate a deformation field from the Wavelet-PCA model of deformation fields;
- Iteratively project it onto the space of a valid-Jacobain and the space of a valid deformation field, and performs the smoothing (here we use a simple neighborhood smoothing in stead of MRF for fast implementation), until the smoothed deformation field and its Jacobian are all located within the valid region according to their statistics. (only one iteration is used in the script)
- Perform inverse WPT and get the final simulated deformation field
- Calculate the warped template according to the simulated deformation field.

3.4. Generate a deformation field using the Landmark-Constrained SSD

This process requires that the statistical models have been properly trained (see Section 3.2 for detail). In addition, a landmark mapping, or correspondences of a number of landmarks from the template image to an individual subject image, has to be known.

The procedure to generate a simulated deformation field using the Landmark-Constrained SSD is described as follows:

- Run script **LinuxPerformLCSSD.pl**

- One can get the command line help by running the above script without any options. The usage is list as follows:

```
This program simulates one new deformation and its respective image
using the Landmark-Constrained SSD

Notice that the statistical models have to be trained using
LinuxTrainSSD.pl prior to using this program

Usage: LinuxPerformLCSSD.pl [options]

-h                   : print this help

Inputs:

-T tempalte_image     : the template image
-t template_segment   : the segmented image of the template image
-M basename for Model : the base(prefix) filename of WPCA models
-S summary_file       : the summary file generated by LinuxTrainSSD

-L template landmark  : landmark file of template image
-l subject landmark   : landmark file of subject image (or the image to
be simulated)

Outputs:

-F simulated_field    : the output simulated deformation field
-f inverse_field      : the inverse deformation field of the simulated
deformation field (optional)
-O simulated_image    : the output simulated image
-o simulated_seg      : the output simulated segmention image
(optional).

Other options:

-J                    : skip the process to calculate the matrix (don't
use this option for the first run)
-Q                    : skip the Jacobian regularization process

Example:

LinuxPerformLCSSD.pl -T template_orig.img -t template_segmented.img -M
PCA8 -S train8.txt -L model02011.pts -l jg_02001_rig.pts -F
sim.ldmk.field -O sim.ldmk.img
```

The above program performs:

- Generate a deformation field so that it not only matches the input landmark mapping, but also reflects the statistics described in SSD.
- Calculate the warped template according to the simulated deformation field.

This program is similar with LinuxPerformSSD.pl, except that it uses a landmark mapping to constrain the SSD.

The format of landmark file is described as follows:

```
No of Landmarks
Coordinates(x,y,z) of Landmark 1
Coordinates(x,y,z) of Landmark 2
…
Coordinates(x,y,z) of Landmark N
```

Please refer to Fig.1 for the definition about the coordinates of the voxels in the MR brain images.

Example:

```
5
10,25,62
81,62,84
61,52,45
131,26,15
161,81,32
```

Note: in order to be able to use Landmark-Constrained SSD, the level of WPT should be fixed to 2.

Some further explanation about the argument of **LinuxPerformLCSSD.pl**

```
-J : skip the process to calculate the matrix (don't use this option
for the first run)
```

In order to perform the Landmark-Constrained SSD, some matrixes have to be calculated (please refer to the paper). These matrixes remain unchanged if the statistical models are unchanged and if the template landmarks (and their sequence) are unchanged. Therefore, one can use this option (-J) to skip the calculation of the matrixes if the statistical models and the template landmarks are unchanged in the subsequent simulations, i.e. in the case that only landmark points for subject images are changed.

4. A quick example

This section gives an example of the above process. After installing the SSD package and setting the path environment of the executables, please enter into the sample folder. There are 8 sample deformation fields generated by HAMMER from the ICBM images in the Talairach space. The files in the sample folder are:

```
filenamelisttest.txt
icbmfun_02001.field
icbmfun_02002.field
icbmfun_02003.field
icbmfun_02004.field
icbmfun_02006.field
icbmfun_02007.field
icbmfun_02008.field
icbmfun_02009.field
template_orig.img
template_segmented.img
```

Then the following three steps are used to quickly test the SSD package.

Step 1 will train the SSD models.

```
LinuxTrainSSD.pl -T template_orig.img -L filenamelisttest.txt -D
256,256,181 -M PCA8 -S -P .98 -O train8.txt
```

Step 2 will then simulate a new deformation and the deformed image using SSD.

```
LinuxPerformSSD.pl -T template_orig.img -t template_segmented.img -M
PCA8 -F sim.field -O sim.img -S train8.txt -o sim.seg
```

After Step 2, sim.img is the simulated image, and sim.seg is the simulated segmented image.

Step 3 simulates the deformation and image using Landmark-Constrained SSD

```
LinuxPerformLCSSD.pl -T template_orig.img -t template_segmented.img -M
PCA8 -S train8.txt -L ../dat/model02011.pts -l ../dat/jg_02001_rig.pts
-F sim.ldmk.field -O sim.ldmk.img
```

After Step 3, “sim.ldmk.field” is the simulated deformation field, and “sim.ldmk.img” is the simulated image.