minc-stuffs (formerly mice-minc-tools)

> ⓘ **minc-stuffs repository**
>
> The mice-stuffs package was formerly called MICe-minc-tools are or mice-minc-tools. The updates to the code can be found on the GitHub page: minc-stuffs github repository. The code can be downloaded and installed from there.

minc-stuffs overview

minc-stuffs is a software package containing disparate command line tools operating on MINC files developed at the Mouse Imaging Centre. The tools are:

Command	Brief Description
#minc_displacement	Create a vector volume containing the displacements at each voxel specified by a transform (.xfm file).
#label_volumes_from_jacobians	Compute the volume of segmented structures given an atlas and Jacobian determinants.
#xfm2tag	Create a tag file from a transform.
#lin_from_nlin	Compute the linear part of a non-linear transform.
#scale_voxels	Multiply values in a file by the combined scale factor of a linear transform.
#smooth_vector	Blur a vector file.
#grid_object_manipulator	Pretty pictures of deformation fields
#add_noise_to_volume	Add normally distributed noise to a volume.
TFCE	Threshold-free cluster enhancement
minc_label_ops	Manipulate label (segmentation) files

Installation instructions

Get the code here

The C code relies on MINC-2.0.09 or greater installed. #smooth_vector requires python, and #add_noise_to_volume requires R and RMINC, and the rest requires perl, Getopt::Tabular, and mni_perllib.

The following commands will perform the install:

```
./autogen.sh
./configure --with-minc2 --prefix=/directory/to/install/to --with-build-path=/directory/containing/minc2
make
make install

python setup.py install --prefix=/directory/to/install/to
```

Description of included programs.

minc_displacement

Usage:

```
minc_displacement [options] input.mnc input.xfm output_grid.mnc
```

minc_displacement evaluates a transform - linear nonlinear, concatenated, etc - at every voxel of an input file and produces a 4D output volume containing the displacement in x,y,z at every voxel.

label_volumes_from_jacobians

Usage:

```
label_volumes_from_jacobians structures.mnc jacobians.mnc
```

label_volumes_from_jacobians produces a list of labels and their volumes from a segmented atlas and a volume of log jacobians. Some important notes:

- The exponent of the input jacobians is taken - it thus has to be a volume of logs of jacobians.
- The atlas volume and the jacobians have to be in the same space with the same sampling - i.e. evaluations take place in voxel coordinates.

xfm2tag

Usage:

```
xfm2tag [options] input.mnc input.xfm output_tags.tag
```

xfm2tag takes a transform and an input volume in the space of which the transform can be evaluated and produces a tag file similar in composition to what register would produce after manual registration. This tag file can then be used to create a newer and perhaps simpler transform using tagtoxfm. xfm2tag is the key tool behind lin_from_nlin.

An optional argument allows one to specify a mask - the transform will only be evaluated inside the mask.

lin_from_nlin

Usage:

```
lin_from_nlin [options] input_volume.mnc input_nonlin.xfm output_lin.xfm
```

lin_from_nlin takes a non-linear transform and an input volume over which the non-linear transform can be evaluated and computes the linear component of the non-linear transform. Options can be used to specify lsq12, lsq9, or lsq6 for the resulting linear transform.

scale_voxels

Usage:

```
scale_voxels [options] transform.xfm volume.mnc output.mnc
```

scale_voxels multiplies every voxel in an input volume by the scale factor determined from a linear transform.

smooth_vector

Usage:

```
smooth_vector --filter --fwhm=x input_vector.mnc output_vector.mnc
```

Blurs a vector volume by separating the vector components, blurring them individually using mincblur, then recombining them.

grid_object_manipulator

Usage:

```
grid_object_manipulator -mincfile mincfile.mnc -transform transformation.xfm
```

grid_object_manipulator uses ray_trace to create pictures of what a deformation grid looks like. Lots of options to control the grid-point spacing, use of overlays, and other ray_trace options. The image below is one example generated by grid_object_manipulator (click on it for a larger image):
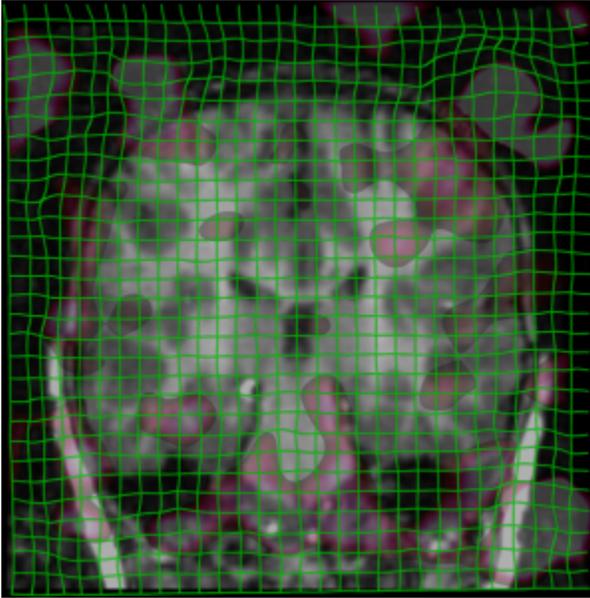
add_noise_to_volume

Usage:

```
add_noise_to_volume [options] -noise x input.mnc output.mnc
```

What every lab needs - a way to make their images look worse. add_noise_to_volume adds x percent normally distributed noise to an input volume. Needs R with the RMINC library installed.

minc_label_ops

Usage:

```
minc_label_ops [options] input_labels.mnc output_labels.mnc

# example: consider the following scenario, you have two atlases
# atlas1.mnc and atlas2.mnc and you want to add labels 56 and 99
# from atlas1.mnc to atlas2.mnc. Run the following:

minc_label_ops --select 56,99 atlas1.mnc atlas1_labels_56_99.mnc
minc_label_ops --merge atlas1_labels_56_99.mnc atlas2.mnc atlas2_with_labels_from_altas1.mnc
```