1 Program 3dExtrema

1.1 Purpose

Program 3dExtrema was developed to find local extrema (minima or maxima) of the functional dataset values for each sub-brick of the input dataset. The extrema may be determined either for each volume, or for each individual slice. Only those voxels whose corresponding intensity value is greater than the user specified data threshold will be considered as possible extrema.

1.2 Theory

1.2.1 Domain of Definition

The user has a choice of the domain for which the local extrema are determined. That is, extrema may be defined for the volume as a whole (option -volume), or extrema may be located on a slice-by-slice basis (option -slice). The set of voxels to be considered as possible extrema can be further delimited by the use of a mask input file (option -mask_file) along with a mask threshold (option -mask_thr).

The user can choose to consider for extrema only voxels that are inside but not *on* the boundary (option -interior), or voxels that are on the boundary may be considered as well (option -closure). The -interior option may be particularly helpful when the mask option is used, as this will prevent identification of voxels as extrema simply because they lie on the boundary of the mask area.

1.2.2 Merging of Extrema

Several options are available for merging of extrema which are close together. Program 3dExtrema calculates the distance between each pair of extrema (within the domain of definition). Two different extrema are merged into a single extrema only if they are the closest pair of extrema, and if they are less than the user specified distance d_{sep} apart. This process is repeated until all extrema are at least distance d_{sep} apart.

Suppose that extrema **A** has location (x_A, y_A, z_A) and extrema **B** has location (x_B, y_B, z_B). Then the Euclidean distance between these two extrema is defined:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}$$

If $d(\mathbf{A}, \mathbf{B})$ is the minimum distance between all pairs of extrema, and if $d(\mathbf{A}, \mathbf{B}) < d_{sep}$, then extrema **A** and **B** will be merged into a new extrema, **C**. This is illustrated by:

$$\left(\begin{array}{l} \text{Extrema } \mathbf{A} \\ \begin{array}{|l|l|} \hline \text{Value} & v_A \\ \hline \text{Location} & (x_A, y_A, z_A) \\ \hline \text{Count} & n_A \\ \hline \text{Sum} & s_A \\ \hline \end{array} \end{array} , \begin{array}{l} \text{Extrema } \mathbf{B} \\ \begin{array}{|l|l|} \hline \text{Value} & v_B \\ \hline \text{Location} & (x_B, y_B, z_B) \\ \hline \text{Count} & n_B \\ \hline \text{Sum} & s_B \\ \hline \end{array} \end{array} \right) \longrightarrow \begin{array}{l} \text{Extrema } \mathbf{C} \\ \begin{array}{|l|l|} \hline \text{Value} & v_C \\ \hline \text{Location} & (x_C, y_C, z_C) \\ \hline \text{Count} & n_C \\ \hline \text{Sum} & s_C \\ \hline \end{array} \end{array}$$

In the above diagram, "Value" refers to the intensity value of the extrema, "Location" is the coordinate vector of the extrema (in mm.), "Count" is the number of extrema which have been merged to form the current extrema, and "Sum" is the sum of intensity values of all extrema which make up the present extrema.

The manner in which the two extrema are replaced by a single extrema depends on the user selected merging option: -remove, -average, or -weight.

Removal

If the user elects to have extrema replacement by removal (option -remove), then the value and location of the new extrema **C** are set equal to the corresponding elements of the more extreme of **A** or **B**. For example, for finding local maxima, we have:

If $v_A \geq v_B$, then:
$$v_C = v_A$$
$$x_C = x_A, y_C = y_A, z_C = z_A$$
$$n_C = n_A + n_B$$
$$s_C = s_A + s_B$$

If $v_B > v_A$, then:
$$v_C = v_B$$
$$x_C = x_B, y_C = y_B, z_C = z_{BA}$$
$$n_C = n_A + n_B$$
$$s_C = s_A + s_B$$

For finding local minima, the reverse of the above inequalities is used.

Averaging

If the user elects to replace the two extrema by their average (option -average), then the new extrema **C** is defined as follows:

$$v_C = \frac{n_A v_A + n_B v_B}{n_A + n_B}$$

$$x_C = \frac{n_A x_A + n_B x_B}{n_A + n_B} \qquad y_C = \frac{n_A y_A + n_B y_B}{n_A + n_B} \qquad z_C = \frac{n_A z_A + n_B z_B}{n_A + n_B}$$

$$n_C = n_A + n_B$$

$$s_C = s_A + s_B$$

Weighted Averaging

Finally, if the user chooses to use the weighted (by their respective intensity values) average of the two extrema (option -weight), the new extrema **C** is given by:

$$v_C = \frac{s_A v_A + s_B v_B}{s_A + s_B}$$

$$x_C = \frac{s_A x_A + s_B x_B}{s_A + s_B} \qquad y_C = \frac{s_A y_A + s_B y_B}{s_A + s_B} \qquad z_C = \frac{s_A z_A + s_B z_B}{s_A + s_B}$$

$$n_C = n_A + n_B$$

$$s_C = s_A + s_B$$

This last option applies only for finding local maxima.

1.3 Usage

1.3.1 Syntax

The syntax for execution of program 3dExtrema is as follows:

3dExtrema [**-prefix** *pname*] [**-session** *dir*] [**-quiet**]
[**-mask_file** *mname*] [**-mask_thr** m_{thr}] [**-data_thr** d_{thr}] [**-sep_dist** d_{sep}]
[**-minima** | **-maxima**] [**-strict** | **-partial**] [**-interior** | **-closure**]
[**-slice** | **-volume**] [**-remove** | **-average** | **-weight**] **datasets**

The different command line options are explained below.

1.3.2 Options

-prefix *pname*
or
-output *pname*
Use *pname* for the output dataset prefix name. Note: If this option is not used, then the only program output will be written to the screen.

-session *dir*
Use *dir* for the output dataset session directory. The default is *dir* = ./ = current working directory.

-quiet
The optional -quiet command is used to suppress screen output as the program proceeds.

-mask_file *mname*
Use file *mname* as a mask dataset. Only voxels that are inside the mask will be considered as possible extrema. If file *mname* contains more than one sub-brick, the specific mask sub-brick must be specfied. (Default: no mask)

-mask_thr m_{thr}
Only voxels whose corresponding mask value is greater than, or equal to, m_{thr} in absolute value will be considered. (Default: $m_{thr} = 1.0$)

-data_thr d_{thr}
Only voxels whose intensity is greater than, or equal to, d_{thr} in absolute value will be considered. (Default: $d_{thr} = 0.0$)

-sep_dist d_{sep}
Minimum separation distance (in mm.) for extrema to be considered distinct.
Extrema separated by less than d_{sep} will be merged. (Default: $d_{sep} = 0.0$)

Choose (one and only one) type of extrema:

-minima	=	Find local minima.
-maxima	=	Find local maxima. (Default)

Choose (one and only one) form of the binary relation:

-strict	=	$>$ for maxima; $<$ for minima. (Default)
-partial	=	$\geq$ for maxima; $\leq$ for minima.

Choose (one and only one) boundary criteria:

-interior	=	Extrema must be interior points only. (Default)
-closure	=	Extrema can be boundary points.

Choose (one and only one) domain for finding extrema:

-slice	=	Each slice is considered separately. (Default)
-volume	=	The volume is considered as a whole.

Choose (one and only one) method for merging extrema:

-remove	=	Remove all but strongest of neighboring extrema. (Default)
-average	=	Replace neighboring extrema with average.
-weight	=	Replace neighboring extrema with weighted average.

Command line arguments after the above are taken to be the names of input datasets. A dataset is specified using one of these forms:

prefix+view   prefix+view.HEAD   prefix+view.BRIK

1.3.3 Sub-brick selection

You can also add a sub-brick selection list after the end of the dataset name. This allows only a subset of the sub-bricks to be used for finding extrema (by default, all of the input dataset sub-bricks are used for finding extrema). A sub-brick selection list looks like one of the following forms:

fred+orig[5]	==>	use only sub-brick #5
fred+orig[5,9,12]	==>	use #5, #9, and #12
fred+orig[5..8] or [5-8]	==>	use #5, #6, #7, and #8
fred+orig[5..13(2)] or [5-13(2)]	==>	use #5, #7, #9, #11, and #13

Sub-brick indexes start at 0. You can use the character '$' to indicate the last sub-brick in a dataset; for example, you can select every third sub-brick by using the selection list:

fred+orig[0..$(3)]

The '$', '(', ')', '[', and ']' characters are special to the shell, so you will have to escape them. This is most easily done by putting the entire dataset plus selection list inside single quotes, as in 'fred+orig[5..7,9]'.

1.4 Examples

Example 1.

Suppose that the bucket dataset myData.bucket+orig is the output of some analysis (such as 3dDeconvolve or 3dNLfim). The user wishes to find the local maxima for the parameter estimates contained in sub-bricks #7 and #11. Also, a separate analysis has yielded dataset signal.max+orig, consisting of a single sub-brick. The user wishes to find local maxima for this sub-brick as well.

Since sub-brick #0 of myData.bucket+orig contains the constant offset, the intensity levels in this sub-brick can be used as a crude mask for determining which voxels are inside the brain. By observation, it is determined that a constant offset of greater than 1333 roughly corresponds to voxels that are inside the brain, whereas voxels with a constant offset of less that 1333 lie outside the brain.

The objective is to find the locations of local maxima, for each individual slice, within the 3 selected volumes. This can be accomplished using the following script:

Command Line for Example 1

```
3dExtrema \
-prefix myData.extrema \
-mask_file 'myData.bucket+orig[0]'  -mask_thr 1333.0 \
-data_thr 400.0 \
-maxima  -strict  -slice  -interior \
-sep_dist 10.0  -average  \
'myData.bucket+orig[7,11]' \
signal.max+orig
```

■

The command -mask_file 'myData.bucket+orig[0]' is used to specify that sub-brick #0 from dataset myData.bucket+orig is to be used as the mask. The command -mask_thr 1333.0 indicates that only voxels whose corresponding mask voxel has absolute value $\geq$ 1333.0 will be considered in finding the extrema.

The command -data_thr 400.0 indicates that only voxels whose absolute value is $\geq$ 400.0 will be considered in finding the extrema.

The next two lines list the various command options for finding the extrema. The commands -maxima and -strict, taken together, indicate that only those voxels whose value is $>$ (not $\geq$) each of its neighbors will be considered as extrema. The command -slice indicates that local maxima are to be found on a slice-by-slice basis. The command -interior indicates that local maxima must be interior to, and not on the boundary of, the set of allowable voxels.

The command -sep_dist 10.0 specifies that extrema less than 10 mm. apart should be merged into a single extrema. The command -average indicates that the merged extrema should be calculated as the average (in position, and in intensity) of the original extrema.

The last two lines of the batch command file specify the input datasets. Sub-bricks #7 and #11 of myData.bucket+orig, as well as the single sub-brick of signal.max+orig, will be used as input for finding local maxima.

The command -prefix myData.extrema indicates that the output bucket dataset, which will contain 3 sub-bricks, is to be written to file myData.extrema+orig.

Screen output for this example is depicted below.

Program 3dExtrema Screen Output from Example 1

```
 Program:  3dExtrema
 Author:   B. Douglas Ward
 Date:     06 April 2001

Reading mask dataset:   myData.bucket+orig[0]
Number of voxels above threshold = 9642
Reading input dataset:   myData.bucket+orig[7,11]
Reading volume #0
Reading volume #1
Reading input dataset:   signal.max+orig
Reading volume #2
Number of volumes = 3

 Maxima for Volume #0 and Slice #0:
  Index  Intensity  RL[mm]  AP[mm]  IS[mm]  Count  Dist[mm]

      1    500.895    1.88   -1.88   28.00      1    18.750
      2    446.787   35.62  -13.12   28.00      1    13.521
      3    439.408  -43.12   20.62   28.00      1    48.750
      4    428.267    1.88  -20.62   28.00      1    18.750
      5    423.741    1.88   39.38   28.00      1    41.250
      6    401.188   43.12   -1.88   28.00      1    13.521
```

⋮

etc.

⋮

```
 Maxima for Volume #2 and Slice #1:
  Index  Intensity  RL[mm]  AP[mm]  IS[mm]  Count  Dist[mm]

      1    603.184   -1.88   65.62   20.00      1    30.233
      2    469.856   50.62    1.88   20.00      1    22.810
      3    446.860    1.88   35.62   20.00      1    30.233
      4    433.850    1.88  -54.38   20.00      1    56.250
      5    417.544   46.88  -20.62   20.00      1    22.810
```

⋮

etc.

⋮

```
 Maxima for Volume #2 and Slice #3:
  Index  Intensity  RL[mm]  AP[mm]  IS[mm]  Count  Dist[mm]

      1    528.884   56.25   -9.38    4.00      2
```

⋮

etc.

⋮

```
Output dataset will have 3 sub-bricks
Computing sub-brick statistics
Writing output to ./myData.extrema+orig.HEAD and ./myData.extrema+orig.BRIK
```

■

Note that the local maxima are listed separately for each slice and for each volume. The coordinates for each local maxima, in mm., are printed out in the format specified by the *AFNI* environmental variable. The "Count" output indicates how many extrema have been merged to form the current extrema. The "Dist" output indicates the distance, in mm., to the nearest extrema.

The output dataset contains 3 sub-bricks. These 3 sub-bricks contain a "1" at the locations of all local maxima within the 3 input volumes. The data type for the output dataset is "byte".