Programs for the experience-dependent synaptic rewiring of geniculo-cortical inputs and the analysis of spatiotemporal receptive fields and orientation maps in the model visual cortex

Overview

This Readme describes how to use a set of programs to conduct the activity-dependent synaptic rewiring of afferent inputs from the model lateral geniculate nucleus (LGN) to the model visual cortex, and analyze simulated results such as orientation selectivity and spatiotemporal receptive fields of individual model neurons and orientation maps in the model visual cortex. "Synaptic rewiring" here is a form of self-organization of synaptic connections, in which a synaptic input from an LGN neuron is replaced by a synaptic input from another LGN neuron at each simulation step depending on the coincidence of pre- and postsynaptic neuronal activities. Therefore, synaptic rewiring is relevant to structural plasticity during the postnatal development of neural circuits.

The mathematical model, on which the main programs described in this Readme are based, is introduced in the paper entitled "Development and reorganization of orientation representation in the cat visual cortex: Experience-dependent synaptic rewiring in early life"by Shigeru Tanaka, Masanobu Miyashita, Nodoka Wakabayashi, Kazunori O'Hashi, Toshiki Tani.

Correspondence: Shigeru Tanaka
Email: shigeru@uec.ac.jp

Table of contents

1. Operating environment

    1.1 Overview of programs

        1.1.1 Programs for the simulation of synaptic rewiring

        1.1.2 Programs for the analysis of simulated results

    1.2 Linux environment

    1.3 Compilation environment and methods

2. Synaptic rewiring programs

    2.1 Visual pathway and data format

    2.2 Changing initial patterns of afferent inputs

    2.3 Outline and operation methods of the programs

3. Analysis programs

    3.1 To analyze responses to grating stimulations

    3.2 To extract preferred orientations and magnitudes

    3.3 To visualize orientation map on the screen

    3.4 To calculate orientation selectivity index (OSI)

    3.5 To display the OSI histogram on the screen

    3.6 To obtain spatiotemporal receptive fields of cortical neurons

    3.7 To visualize snapshots of a spatiotemporal receptive field on the screen

1. Operating environment

1.1 Overview of programs

1.1.1 Programs for the simulation of synaptic rewiring

(1) **init.f** (FOTRAN language)

This program generates an initial pattern of afferent inputs from model LGN neurons to visual cortical neurons randomly.

(2) **slforg_nml.f** (FOTRAN language)

This program simulates the activity-dependent self-organization of afferent inputs from model LGN neurons to visual cortical neurons according to our synaptic rewiring mechanisms. A sinusoidal drifting grating pattern with the spatial frequency of 0.4 cpd and the temporal frequency of 4 Hz is presented on the model visual field. The direction of stimulus movement changes with an increment of 15 ° in one stimulation from 0 ° to 360 °. Consequently, 24 stimulus directions are presented once (12 orientations presented twice) at each trial of synaptic update.

(3) **slforg_sgl.f** (FOTRAN language)

Similar to the slforg_nml.f program, this program simulates the activity-dependent synaptic rewiring of geniculo-cortical afferent inputs. Only a vertically oriented grating drifting leftward and rightward is presented.

1.1.2 Programs for the analysis of simulated results

(1) **respns.f** (FOTRAN language)

This program calculates responses of individual neurons based on the self-organized afferent inputs. The visual stimuli presented are the same as stimuli used in slforg_nml.f. The response strengths of visual cortical neurons are calculated at a step of 1 ° from 0 ° to 360 °. An orientation (or direction) tuning curve is drawn using the output data of this program.

(2) **map.f** (FOTRAN language)

This program extracts a preferred orientation and orientation magnitude of individual cortical neurons, applying the vector sum method to the output data of **rspns.f**.

(3) **ori.c** (C language)

This program visualizes the orientation map as a color-coded map on the screen, using the output of the map.f. Brightness indicates the orientation magnitude.

(4) **osi.f** (FOTRAN language)

This program calculates the values of the orientation selectivity index (OSI) of individual cortical neurons using the output data of **rspns.f**.

(5) **osi_dst.f** (FOTRAN language)

This program displays the OSI histogram (distribution of OSI values over the model cortex) on the screen using the output of the **osi.f**.

(6) **RFSsmp.f** (FOTRAN language)

This program generates the spatiotemporal receptive field of a visual cortical neuron at every 10 ms after the onset of visual stimulation.

(7) **rfs.c** (C language)

This program displays spatial components of the spatiotemporal receptive field generated at every 10 ms after the onset of visual stimulation on the screen using the output data of **RFSsmp.f**.

1.2 Linux environment

The programs shown in the section 1.1 have been developed in the Linux OS environment. Linux OS version is

CentOS Linux release 7.2.1511 (Core).

These programs do not work on FORTRAN and visual C that run on windows OS because the environment is different. The version information for CentOS is shown below.

etc/redhat-release

CPU is the Intel Xeon chip

model name: Intel (R) Xeon (R) Platinum 8160 CPU @ 2.10GHz

We confirmed execution on this CPU. Two CPUs with 24 cores are installed. On Linux OS, the installed CPU information is in

/ proc / cpuinfo

Check the memory capacity with the “top” command. A large amount of memory is not required to execute the programs, but the output file size of **rspns.f** is about 20,736,000 bite. Please confirm that your HDD vacant space is sufficiently large.

It takes a long time to execute **slforg_nml.f**. It is recommended to prepare an environment that allows parallel computing with 4-8 cores.

1.3 Compilation environment and methods

To execute each program, it is necessary to compile and create an executable object file. The “makefile” is given as follows:

```
OPTIONS = -O3 -ipo -no-prec-div -parallel -par_threshold0 -mcmodel=large -
D_FILE_OFFSET_BITS=64
f:
/opt/intel/compilers_and_libraries_2017/linux/bin/intel64/ifort -o $@ $< $(OPTIONS)
.c:
cc $< -o $@ -L/usr/X11R6/lib64 -lX11 –lm
```

In **slforg_nml.f** and **slforg_sgl.f** written in FORTRAN language, the code of the OpenMp command is written as comment text. The execution time is almost the same whether you use parallel calculation with OpenMp or compile with the parallelization option described in the “makefile”. To specify the number of threads for parallelization as 8, set the following “setenv” command:

setenv OMP_NUM_THREADS 8

The number of threads indicates the number of cores for parallel computing. To execute parallel computing, Intel FORTRAN compiler is necessary.

Programs **ori.c** and **rfs.c** that display orientation maps and receptive fields can be compiled with “cc” or “gcc”. However, as described in the “makefile”, add the following options to compile the programs.

-L / usr / X11R6 / lib64

-lX11

A program written in C language is required to open the X-window to display an image on a screen, because the program sends drawing commands to the X window.

2. Synaptic rewiring programs

2.1 Visual pathway and data format

Visual pathway

Figure 1 is a schematic picture of the visual pathway. In the synaptic rewiring program, synaptic connections from the model LGN cells to the visual cortical cells are rewired in a self-organizing manner. The model visual cortex contains 48 × 48 cells, and each cell has 254 synaptic sites. The space of the model LGN is composed of 24 × 24 cells, and there are four types of cells with different spatiotemporal responses: On-center / nonlagged; On-center / lagged; Off-center / nonlagged and Off-center / lagged. Therefore, the model LGN has 24 × 24 × 4 cells. To avoid the edge effects, the periodic boundary condition is imposed on both the model visual cortex and the model LGN. That is, the visual cortex and LGN are assumed to be tori topologically.

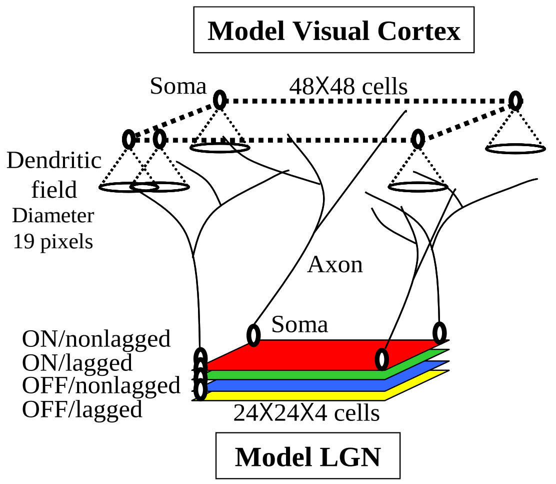

Figure 1 Schematic picture of the model visual pathway

Data format

The corner positions of cells in the model LGN and visual cortex are shown in Figure 2. Model visual cortical cells have a 2D disk-shaped dendritic field with a diameter of 19 pixels (9 + 1 + 9 pixels). It accommodates 254 (~ $\pi \times 9 \times 9$) afferent synaptic inputs from the LGN within a square region of 19 × 19. The synaptic connections are stored in the order (1, 1) ~ (48, 1) ~ (1, 48) ~ (48, 48).

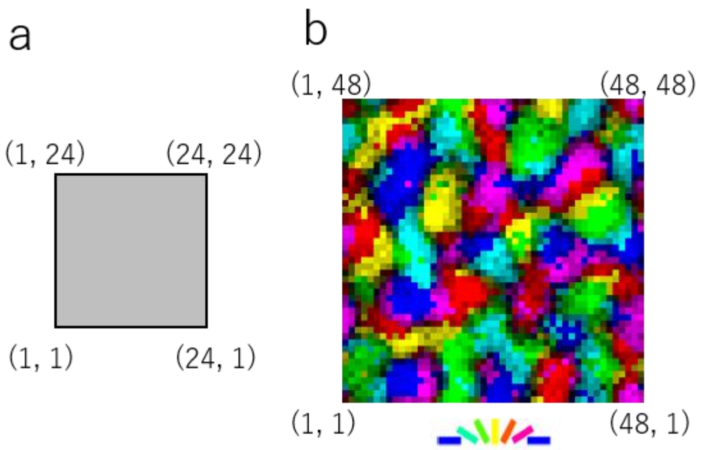

Figure 2 (a) Model LGN, and (b) Model visual cortex

Values of the data indicate cell types and retinotopic locations of individual LGN neurons as shown in Table 1.

Table 1. Correspondence between values of connection data and LGN neurons

Cell type	Number
On-center nonlagged	1 ~ 24×24 (1 ~ 576)
On-center lagged	24×24+1 ~ 24×24×2 (577 ~ 1152)
Off-center nonlagged	24×24×2+1 ~ 24×24×3 (1153 ~ 1728)
Off-center lagged	24×24×3+1 ~ 24×24×4 (1729 ~ 2304)

The following is the array in data0 representing synaptic connections from LGN neurons at the dendritic sites of the visual cortical neuron located at (1,1).

```
    0    0    0    0    0    0    0    0 1199 2161 2182    0    0    0    0    0    0    0    0
    0    0    0    0    0 2233 1749 1054  529 1177  647  600  574  480    0    0    0    0    0
    0    0    0    0  454 1082   25 1081 1607  648 1202 1131 2210 1703 2234    0    0    0    0
    0    0    0   20 2275 1706   48 1032 2260 1749 1609 1054 2233 1010 2230 1799    0    0    0
                                              :
                                              :
 1224 1105  504 2282  672 1105 1224  621    3 1154 2284  556 1631 1726  551   46 1084 1126 1060
 2229  597  503 1634 1149 1775 1776  552   49 1797 1198  554 2233 1730  551 1779  506 1178   52
  626  624 1147   19 1802  602  621 1221  529 2233 1108  503   45 1756 1060 2233 1204  533 1658
                                              :
                                              :
    0    0    0 1106 1200   74 1172 1799  625 1149 1778 1081 2284 1778 1251  598    0    0    0
    0    0    0    0   47 1775  553 1173   70 1801 1682  623    3 1181 2280    0    0    0    0
    0    0    0    0    0   50   25 1706  579  599 1156 1103 2255 1752    0    0    0    0    0
    0    0    0    0    0    0    0    0 1106  553 1106    0    0    0    0    0    0    0    0
```

The value "0" indicates the absence of synaptic inputs. For example, the value "1199" on the first line indicates that the LGN neuron belongs to the OFF-center nonlagged type, as shown in Table 1 (1153 < 1199 < 1728). 1199- (24 × 24 + 1) = 46 means that the 46th neuron of the LGN. That is, the dendritic site receives an input from the OFF-center nonlagged LGN neuron at the position (22, 2).

2.2 Changing initial patterns of afferent inputs

Programs for simulating the synaptic rewiring of afferent inputs, **slforg_nml.f** and **slforg_sgl.f** start with random afferent input patterns. The "data0" stores such an initial pattern of random afferent inputs. When you need to generate another pattern of random afferent inputs, you can use **init.f**. It reads a file of an afferent input pattern and creates another file of different random pattern.

init.f (FOTRAN language)

When you start **init.f**, the following message appears on the screen:

```
> ./init
# Input   old initial connection file =?
```

Enter the filename "data0", as shown below.

```
> ./init
# Input   old initial connection file =?
```

data0

Next, the program asks for entering a new filename. Type, for example, “data0_new”.

```
> ./init
 # Input   old initial connection file =?
data0
 # Output new initial connection file =?
data0_new
```

A new initial pattern of afferent inputs is generated and stored in “data0_new”, which is created under the executed directory.

2.3 Outline and operation methods of the programs

The simulation of the activity-dependent synaptic rewiring of afferent inputs from the model LGN to the model visual cortex under balanced exposure to 24 directions of stimulus movement (12 orientations) is performed by using

slforg_nml.f. (FOTRAN language)

In this program, a sinusoidal drifting grating with the spatial frequency of 0.4 cpd and the temporal frequency of 4 Hz in various directions of movement is presented on the model visual field. The direction of movement changes with the increment of 15 ° from 0 ° to 360 ° (24 directions and 12 orientations).

The simulation of the activity-dependent synaptic rewiring of afferent inputs from the model LGN to the model visual cortex under the exposure to a single orientation is performed by using

slforg_sgl.f. (FOTRAN language)

In this program, only a sinusoidal grating oriented vertically, which moves leftward and rightward, is presented.

The operation methods of **slforg_nml.f** and **slforg_sgl.f** are the same. When you start **slforg_nml.f**, the following message is displayed:

```
> ./slforg_nml
# Input synaptic connection file =?
data0_new
```

The program asks for the filename of an initial pattern of afferent inputs from the LGN to the visual cortex. Here, the example shows that you use data0_new as an initial pattern of random afferent inputs generated by **init.f**. Then, the program ask for entering the Monte Carlo step (MCs) to perform as follow:

```
> ./slforg_nml
# Input synaptic connection file =?
data0_new
Monte Caro Step =?
```

Here, for example, type “1.57”.

```
> ./slforg_nml
# Input synaptic connection file =?
```

data0_new

Monte Caro Step =?

1.57

Enter the number of executed steps =?

The program asks for entering the total number of MCs, for which you have performed simulations in a continual sequence so far. If this run is a first one, type “0”.

> ./slforg_nml

Input synaptic connection file =?

data0

Monte Caro Step =?

1.57

Enter the number of executed steps =?

0

Output synaptic connection file =?

Next, the program asks for entering a filename to store the simulation results. It is convenient to define the output filename so that you can see the total number of MCs of simulations that have been performed. For example, let the output filename be “data1.57”, here.

> ./slforg_nml

Input synaptic connection file =?

data0

Monte Caro Step =?

1.57

Enter the number of executed steps =?

0

Output synaptic connection file =?

data1.57

How to continue simulations

If you want to resume simulation by 3.15 MCs with data1.57 as an input pattern, you will enter as follows:

> ./slforg_nml

Input synaptic connection file =?

data1.57

Monte Caro Step =?

3.15

Enter the number of executed steps =?

1.57

Output synaptic connection file =?

data4.72

If you want to continue simulation further by 1.57 MCs, you will enter as follows:

> ./slforg_nml

Input synaptic connection file =?

data4.72

Monte Caro Step =?

1.57

Enter the number of executed steps =?

4.72

Output synaptic connection file =?

data6.29

Likewise, if you want to run simulation under single-orientation exposure by 3.15 MCs as a continuation of "data4.72", enter filenames of connection pattern data and MCs as follows:

> ./slforg_sgl

Input synaptic connection file =?

data4.72

Monte Caro Step =?

3.15

Enter the number of executed steps =?

4.72

Output synaptic connection file =?

data7.87_sgl

3. Analysis programs

3.1 To analyze responses to grating stimulations

respns.f (FOTRAN language)

Using the stimuli presented on the model visual field same as those used in the synaptic rewiring program **slforg_nml.f**, the response of model visual cortical neurons to each direction stimulus is calculated. The stimulus direction increases from 0 ° to 360 ° by the step of 1 °. Therefore, the output data is composed of 360 values of stimulus responses for each direction of movement in a single model cortical neuron.

How it works

When you start **rspns.f**, the following message is displayed:

```
> ./rspns
# Input synaptic connection file =?
data1.57
```

The program asks for entering the filename of a pattern of afferent inputs from the LGN to the visual cortex. Here, for example, enter “data1.57” generated by **slforg_nml.f**.

```
> ./rspns
# Input synaptic connection file =?
data1,57
# Output direction response data  (drsp#) =?
drsp1.57
```

Then, the program asks for entering a filename to store the result. Type, for example, “drsp1.57”.

3.2 To extract preferred orientations and magnitudes

map.f (FOTRAN language)

This program uses the output data of **rspns.f** to calculate the value of the magnitude and preferred orientation of each cortical neuron using the vector sum method. The magnitude is given by

$$m_i = \sqrt{c_i^{\,2} + s_i^{\,2}}\ . \qquad (1)$$

where

$$\begin{aligned} c_i &= \frac{1}{N}\sum_{p=0}^{N-1} a_i(\theta_p)\cos(2\theta_p), \\ s_i &= \frac{1}{N}\sum_{p=0}^{N-1} a_i(\theta_p)\sin(2\theta_p), \end{aligned} \qquad (2)$$

$a_i(\theta_p)$ represents the response of the *i* th visual cortical neuron to the stimulus orientation in the interval of $0-180^\circ$. N is the number of grating stimuli, $N = 180$. θ_p is given by $\theta_p = \pi \cdot p/N$, $(p = 0, 1, 2, \ldots. N-1)$, m_i is the value of magnitude. The preferred orientation is calculated using the following formula:

$$\theta_i^{pref} = \begin{cases} \varphi_i & \text{for } c_i > 0 \text{ and } s_i > 0 \\ \varphi_i + \dfrac{\pi}{2} & \text{for } \quad c_i < 0 \quad , \\ \varphi_i + \pi & \text{for } c_i > 0 \text{ and } s_i < 0 \end{cases} \tag{3}$$

$$\text{here, } \varphi_i = \frac{1}{2}\tan^{-1}\left(\frac{s_i}{c_i}\right), \ \left(\frac{\pi}{2} > \tan^{-1} x > -\frac{\pi}{2} \ for x \in (-\infty, +\infty)\right).$$

The data format for one visual cortical neuron is set to be

preferred direction, 0, 0, 0, preferred orientation, magnitude, 0

Here, the locations of "0" are reserved as those of parameter values that we may use in the future, for example, direction selectivity index, tuning width, preferred spatial frequency, and so on.

How it works

When you start **map.f**, the following message is displayed.

```
> ./map
# response file (drsp#) =?
drsp1.57
```

The program asks for entering a filename of response data created by **rspns.f**. Here, type, for example, "drsp1.57".

```
> ./map
# response file (drsp#) =?
drsp1.57
# Output dir. ori. magnitude data (dmap#) =?
dmap1.57
```

Then, the program asks for entering a filename of the map data to store. For example, you may type "dmap1.57".

3.3 To visualize orientation maps on the screen

ori.c (C language)

The orientation polar map is displayed on the screen using the result of **map.f**. The brightness indicates the intensity of the magnitude, and when the value of magnitude is 0.85 or more, it is displayed at the maximum brightness. When the execution of this program is completed, the bmp file of the displayed orientation map is created under the executed directory with the name "map.bmp".

How it works

When you start **map.f**, the following is displayed.

```
> ./ori
input file name : dmap1.57
Hit Return Key
```

When you press "Enter" key, **map.f** ends and a map.bmp file is created.

3.4 To calculate the orientation selectivity index (OSI)

osi.f (FOTRAN language)

The orientation selectivity index (OSI) of model visual cortical neurons can be calculated according to the following definition using the orientation tuning curve (Fig. 3).

$$\mathrm{OSI}(i)=\left(1-\frac{r_{\min}(i)}{r_{\max}(i)}\right)\left(1-\frac{w(i)}{180}\right), \tag{4}$$

where $r_{min}(i)$ and $r_{max}(i)$ show the minimum and maximum of the response of the i th neuron, respectively, as

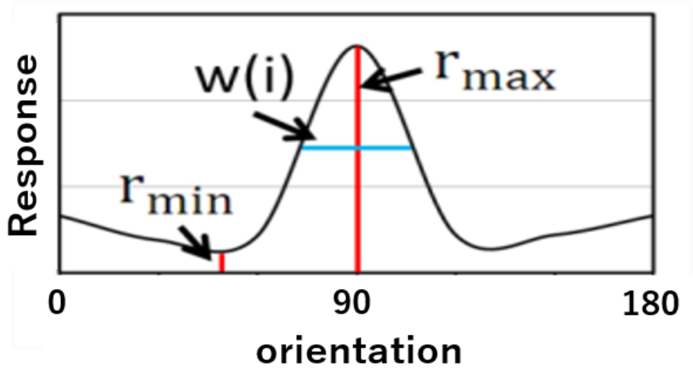

Figure 3 orientation tuning curve

shown in Fig. 3. $w(i)$ indicates the full width at half height.

The orientation tuning curve may have multiple peaks, as shown in Fig. 4. Among such cases, when the horizontal line at the half maximum crosses the tuning curve three times or more, the OSI is defined to be 0.

How it works

When you start **osi.f**, the following message is displayed:

```
> ./osi
# Input response file (drsp#) =?
drsp1.57
```

The program asks for entering a filename in which responses of the model visual cortex neurons is stored. Type "drsp1.57", a name of the file created by **rspns.f**.

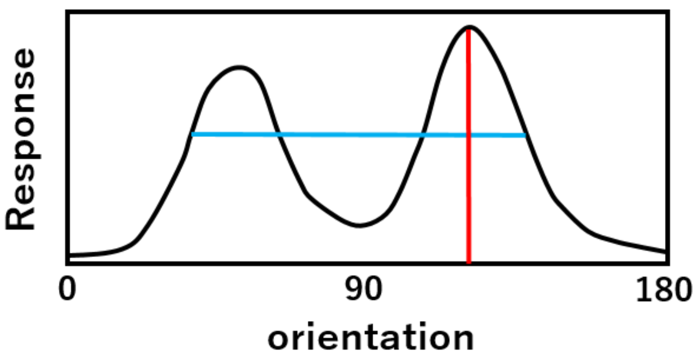

Figure 4 Orientation tuning curve with double peaks

```
> ./osi_idx
# Input response file (drsp#) =?
drsp1.57
# Output OSI file        (dosi#) =?
dosi1.57
```

Then, the program asks for entering a name of a file in which the results are stored. Type, for example, “dosi1.57”.

3.5 To display the OSI histogram

osi_dst.f (FOTRAN language)

This program displays the OSI histogram of model visual cortical neurons on the screen using the OSI values of individual neurons generated by **osi.f**. The bin width is 0.08.

How it works

When you start **osi_dst.f**, the following message appears on the screen. The program asks for entering a name of a file, in which the OSI values are stored. Type “dosi1.57” created by **osi.f**.

```
> ./osi_dst
# input OSI data (dosi#) =?
dosi1.57
```

```
     ** OSI distribution **
   OSI              number ratio of neurons
-------------------------------------------------
  0.0000000E+00       0.6493056
  7.9999998E-02      2.6475694E-02
  0.1600000          3.3420138E-02
  0.2400000          5.6423612E-02
  0.3200000          6.7274310E-02
  0.4000000          8.0729164E-02
  0.4800000          5.3385418E-02
  0.5600000          2.9513890E-02
  0.6400000          2.6041667E-03
  0.7200000          8.6805556E-04
  0.8000000          0.0000000E+00
  0.8800000          0.0000000E+00
-------------------------------------------------
```

3.6 To obtain a spatiotemporal receptive field of cortical neuron

RFSsmp.f (FOTRAN language)

By entering a pair of values representing the 2D position of a visual cortical neuron, its spatiotemporal receptive field is obtained (See Fig. 2 for the coordinate system of the model visual cortex). Information on the snapshots of the spatiotemporal receptive field every 10 ms is stored.

How it works

When you start **RFSsmp.f**, the following message is displayed:

```
# Input synaptic connection file =?
data1.57
```

The program asks for entering the filename, in which a pattern of afferent inputs from the LGN to the visual cortex is stored. Type, for example, “data1.57”.

```
> ./RFSsmp
# Input synaptic connection file =?
data1.57
# Cortex cell location =(X, Y)?
24, 24
```

Then, the program asks for entering a position vector of the visual cortex. For example, if you want to show the spatiotemporal receptive field of cortical neuron locate at (24, 24), type “24, 24”.

```
>./RFSsmp
# Input synaptic connection file =?
data1.57
# Cortex cell location =(X, Y)?
24, 24
# Output spatio-tempral RF file =?
drfs1.57
```

The program asks for entering a name of a file to store the results. Type, for example, “drfs1.57”.

3.7 To visualize snapshots of a spatiotemporal receptive field on the screen

rfs.c (C language)

This program displays the spatiotemporal receptive field generated by **RFSsmp.f** on the screen as a model visual field of 24 × 24 pixels.

How it works

When you start **rfs.c**, the following message is displayed:

```
> ./rfs
input file name : drfs1.57
Hit Return Key
```

Enter, for example, “drfs1.57”, which is generated by **RFSsmp.f**. Press Enter key to exit the program.