Sehbau: Focus Extraction (Vector Selection)

The use of the program for selecting descriptors from a vector file is explained, named **focx**. The program takes as input a vector file (**vec**) and a region specified as bounding box, called the *focus*. The program then extracts the descriptors contained in that focus, and saves them to a separate file with extension **vef**. This separate output can then be matched to representations of object instances or object categories using the vector matching program **mvec**.

Read	still needs to be applied and written.
Repository	`https://github.com/Sehbau/FocExtr`
Docu	overview: `https://github.com/Sehbau/Docu/blob/main/overview.pdf`

The directories in the repository contain the following:

- /**Desc**	sample vector files (as generated by **dscx**)
- /**Focii**	output directory for focal selections, the **.vef** files
- /**UtilMb**	Matlab scripts to read the output data files

The program comes in three variants:

- **focxv** : extracts the subset of descriptor **v**ectors for one focus, ie. for full vector-by-vector matching.
- **focxh1** : generates an attribute **h**istogram for one focus, for rapid classification.
- **focxhL** : generates multiple attribute histograms for focii specified in a text file.

The histogram output of the latter two programs allow for rapid classification and therefore rapid visual orienting. The Matlab scripts `runFocxv1.m`, `runFocxh1.m`, `runFocxhL.m` and `runFocxFew.m` demonstrate how to deploy those programs. They call the corresponding wrapper functions named `RennFocxv1.m`, `RennFocxh1.m` and `RennFocxhL.m`

1 Program Use

Firstly the use of **focxv** is explained. It takes three arguments:

1) a vector file as generated by **dscx**. The file name *must* include extension **vec**.
2) bounding box parameters specified as `top bottom left right`.
3) [optional] an output file name *without* extension, where the selected vectors are written to.

Example: We extract a `40x40` region (height x width) from the upper left:

```
> focxv Desc/img1.vec 10 50 10 50 Focii/foc1
```

The selected descriptors are then saved to directory /**Focii** appending the extension **vef**. If the output file name is not specified, the program will write to the file called **Focus.vef** in the same directory. How the output file is loaded will be explained in the next section.

The program calculates the number of pyramid levels automatically for the selected region. If no descriptors are found in that subspace, then no output file will be saved. More details on its output will follow below.

The program **focxh1** generates a single histogram, which is written to a file with extension **hsf1**. The program requires the bin file with extension **veb** to be present (in the same directory as the **vec** file), as generated by **dscx**.

The program **focxhL** does the same as **focxh1** but for a list of focii specified in a text file, named **BboxFocii.txt**, ie.

```
> focxhL Desc/img1.vec BboxFocii.txt FOCII1
```

in which the bounding boxes are given rowwise. The output is written to file **FOCII1** in the example.

2 Output

The program writes both standard and file output.

The standard output contains the number of levels calculated automatically, `nLevFoc`, and the total number of descriptors, `ntDsc`, for example:

```
nLevFoc 3
ntDsc 27
```

If no descriptors are found, the output returns `ntDsc 0` and no data are written to file.
The vector values are written to the **vef** file. As mentioned already, if no output file is specified, the values will be written to file **Focus.vef** in the same directory. The **vef** is similar to the **vef** file as generated by **dscx**. It is loaded as demonstrated in script `LoadFocVect`, located in directory /**UtilMb**.

The script `runFoc1.m` plots both the vectors of the entire image, and those of the focus (selection). The plotting routines (`PlotCntSpc`, `PlotRsgSpc`, ...) are found in directory /**Plot** of the repository for descriptor extraction (program **dscx**).