

Stereo Software

This document summarizes the MATLAB software that we will use to create and display stereograms, run the Marr-Poggio cooperative stereo algorithm, and display the results of this algorithm. This software is available in M-Files stored in the following directory on the CS file server: /home/cs332/download/stereo.

Creating and displaying stereograms

makeStereogram.m

```
[left right anaglyph] = makeStereogram(dmap, percent)
```

`makeStereogram` creates a random-dot stereogram in the output `left` and `right` images with the percentage of white dots given by the input `percent`, which should be a number between 0.0 and 1.0. The disparities for the stereogram should be specified in the input `dmap` matrix. `makeStereogram` also creates a color anaglyph from the left and right images of the stereogram, which is stored in the output `anaglyph` image. All of the output images are the same size as the `dmap` matrix.

showStereogram.m

```
showStereogram(left, right)
```

`showStereogram` displays the left and right halves of a stereogram side-by-side in a single figure window, using `imshow`.

magicEye.m

```
image = magicEye(dmap)
```

`magicEye` creates a magic-eye stereogram with random colors, from the disparity map stored in the input `dmap` matrix. The output image is the same size as the `dmap` matrix.

Displaying a color anaglyph

You can display a color anaglyph in a new figure window using `imshow`, with the name of the matrix storing the anaglyph given as input:

```
>> figure, imshow(anaglyph);
```

makeCake.m

```
dmap = makeCake(step)
```

`makeCake` creates and returns a 200x200 matrix with a disparity pattern that consists of concentric circles of four different disparities (0, `step`, $2*\text{step}$ and $3*\text{step}$).

makeDoubleStereogram.m

```
[left right] = makeDoubleStereogram
```

`makeDoubleStereogram` creates a 10% random dot stereogram that violates the uniqueness constraint. Each white dot in the left image is copied twice in the right image, at disparities of 0 and 4 pixels.

Running the Marr-Poggio cooperative stereo algorithm and displaying results

coopStereo.m

```
[init_net final_net] = coopStereo(left,right,minD,maxD,itters)
```

coopStereo implements the Marr-Poggio cooperative stereo algorithm. The `left` and `right` input images should contain a stereogram. The inputs `minD` and `maxD` are the minimum and maximum disparities in the stereogram. The input `itters` specifies the number of iterations to be executed. The `coopStereo` function returns the initial and final state of the network stored in 3D matrices of double type values. The intermediate states of the network are displayed in separate figure windows as the algorithm progresses, using the `showDispPlanes` function described below.

showDispPlanes.m

```
showDispPlanes(network)
```

`showDispPlanes` displays a state of the input 3-D network for the cooperative stereo algorithm as individual disparity planes in a single figure window. The disparity planes are labeled using the coordinate of the third dimension of the input 3-D network for the particular disparity plane (starting at plane 1).

showNetwork.m

```
showNetwork(network)
```

`showNetwork` displays a state of the input 3-D network for the cooperative stereo algorithm as a gray-level image, where each location shows the average of the active disparities at that location (i.e. the average of the disparities yielding a value of 1 in the network).

stereoZeros.m

```
stereoZC = stereoZeros(conv)
```

`stereoZeros` computes the zero-crossings of the input convolution, for use in stereo processing. Only those zero-crossings detected in the horizontal direction are stored in the output matrix. A 1 is stored at the locations of positive zero-crossings, and -1 is stored at the locations of negative zero-crossings.

Special script for Assignment 2

For the examples used in Assignment 3, the following script file contains commands to create and display stereograms, run the Marr-Poggio cooperative stereo algorithm and display the results of the algorithm. Many of the commands are initially commented out:

```
stereoScript.m
```