

Multiscale Normalized Cuts Segmentation Toolbox for MATLAB

Authors: Timothee Cour, Florence Benezit, Jianbo Shi

Version: 1.5

Date: 08/04/2008

Download the latest version from <http://www.seas.upenn.edu/~timothee>

This software is made publicly for research use only. It may be modified and redistributed under the terms of the [GNU General Public License](#).

Please cite the paper and source code if you are using it in your work.



Image size: 506 x 344 (Copyright "The First Emperor of China", Ching-chih Chen)

Related publication

Timothee Cour, Florence Benezit, Jianbo Shi. [Spectral Segmentation with Multiscale Graph Decomposition](#). *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

Installation instructions

Make sure you have MATLAB 7.0 or higher and the Image Processing Toolbox.

Running the scripts

1) Unzip the files to some directory mydir

2) In matlab, type:

```
cd mydir
```

```
init
```

```
compileDir //only need to be run once to compile mex-files
```

```
script_ncut_multiscale // script for basic multiscale segmentation
```

```
script_ncut_multiscale_timing // script for computation time
```

Note: if you are using AMD 64 and comileDir brings errors, this might be due to a bug in gcc. The following should fix it: edit the config file for compiling matlab mex files. It is typically located under: `/home/username/.matlab/R14SP3/mexopts.sh`. Go to the section `glnxa64`, and replace the optimization flag `-O` with `-O2`:

```
COPTIMFLAGS='-O2 -DNDEBUG'
```

```
CXXOPTIMFLAGS='-O2 -DNDEBUG'
```

Please address questions / bug reports to: timothee “dot” cour “at” gmail “dot” com

Description

Main Function summary	Description
script_ncut_multiscale	Script that calls main function ncut_multiscale
script_ncut_multiscale_timing	Script that shows running time and segmentation regions for variable input image sizes
ncut_multiscale	[classes,X,lambda,Xr,W,C] = ncut_multiscale(image,nsegs); Computes segmentation eigenvectors (X) and regions (classes) of an image, into nsegs segments
discretisation	compute discrete regions from continuous eigenvectors
computeParametersW	sets parameters for computing multiscale image affinity matrix W
computeParametersLayers	sets parameters for each layer in the multiscale grid
Utility Functions	
init	Adds path to subdirectories of current directory
compileDir	compiles all mex files in specified directory

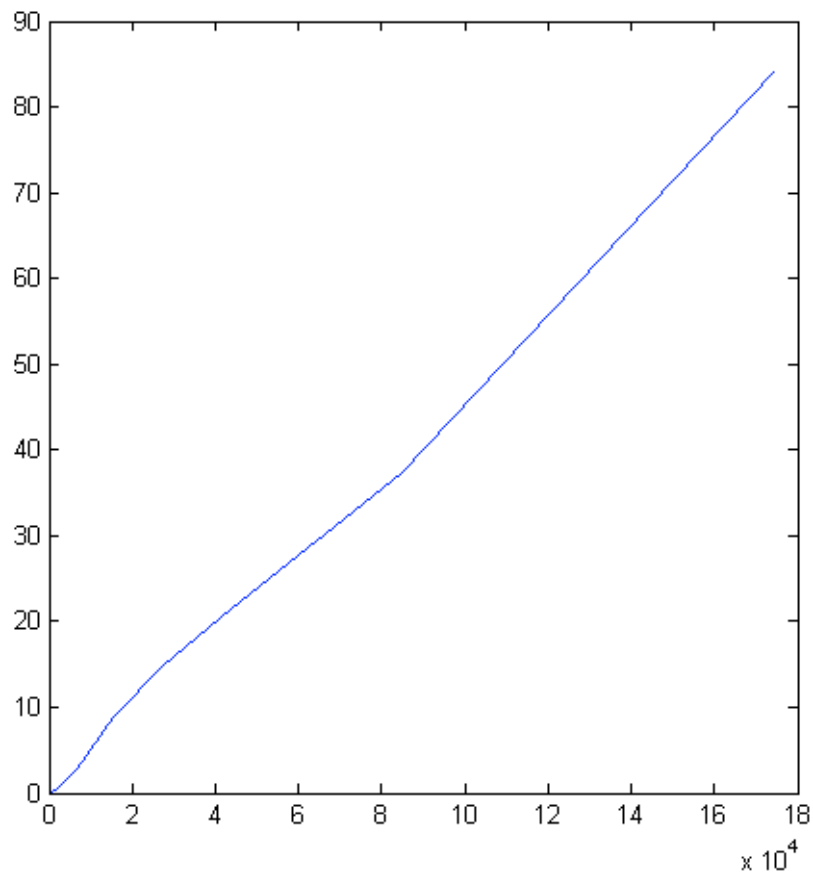
Running Time

We showed in the article and verified in experiments that the running time depends **linearly** on the image size and number of segments requested. Note, the main computation bottleneck is in the eigenvector computation, which comprises two main terms:

- a) the cost of matrix vector multiplications (see article)
 - b) the routine dsaupd from arpack (<http://www.caam.rice.edu/software/ARPACK/>).
- We observe that b) is up to several times slower than a). We hope to reduce the cost of b) in a future release, which is related to the eigensolver used.

For small images (150x100), expect less than 10-15 seconds (< 2 seconds for matrix vector operations)

For large images (>500x500), expect one or more minutes (< 50 seconds for matrix vector operations)



Computation time = $f(\text{number of pixels})$: as the demo shows, the complexity of multiscale segmentation is **linear**. [note: here eigensolver tolerance was set to $1e-2$ instead of $1e-3$, giving slightly less precise segmentations; expect to triple the time for the default eigensolver tolerance)]