# Contents

# 1 Introduction

this is introduction

# 2 Installation

This section covers compiling LSBPT from source.

LSBPT is a Qt Widgets application developed with qt4, it has been tested under Linux ( fedora 23) and Mac OSX. It depends on several external libraries, listed below with version I used, other version libraries may also work.

- opencv 2.4.12.2
- gdal 2.0.2
- OpenMP (gcc-4.9.2) Optimal. if not enabled, there may be some warnings " ignoring #pragma omp ..."

Edit the "LSBPT.pro" in src folder to configure the library dependencies.

## 2.1 Linux and Mac OS X

Once you solve the dependencies , follow the instructions to compile.

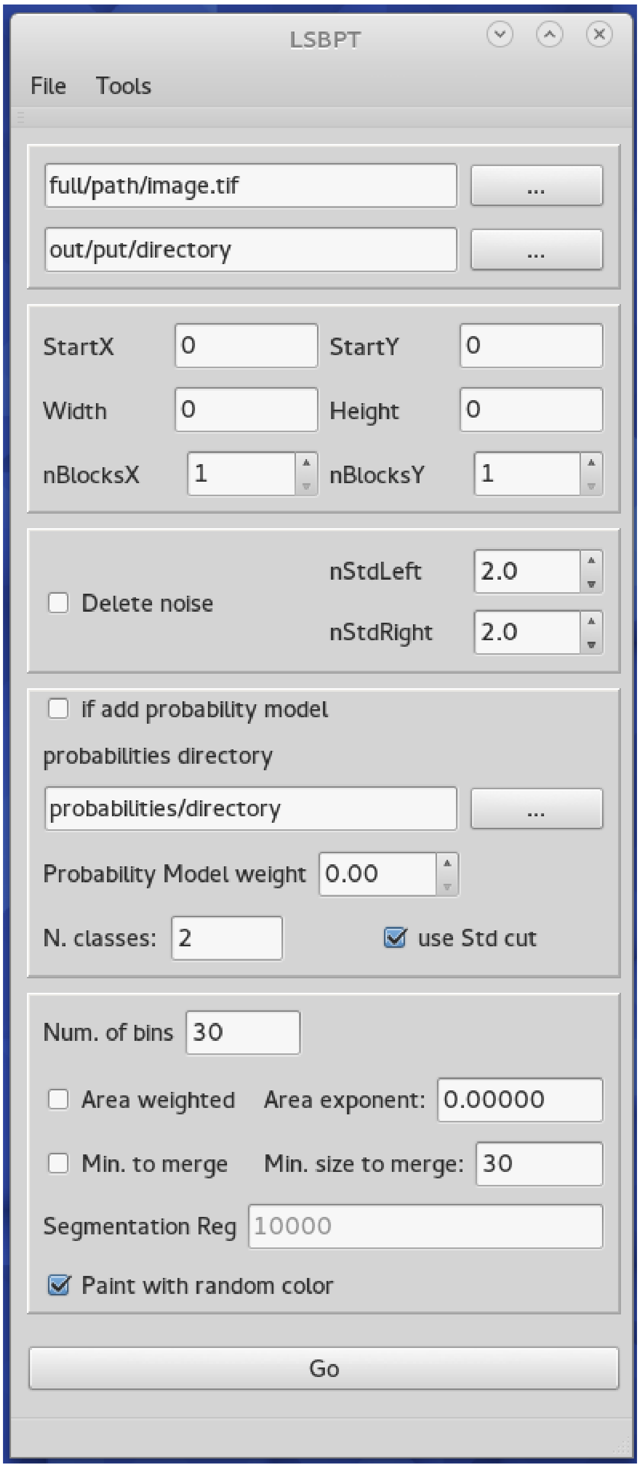
- git clone
- cd to the directory ∼/lsbpt
- *mkdir build* Create a build directory
- *cd build* Cd to build
- *qmake ../src* Use qmake to create makefile
- *make* Compile
- *chmox +x apps/LSBPT* Make the file executable

After compiling, you'll find the executable file "LSBPT" in /lsbpt/build/apps, simply type " *path/to/LSBPT* " (i.e, *app/LSBPT* if you are in build directory) in bash to launch the application.

If you are using Qt creator, simply use "Open project" and select the "LSBPT.pro" under " *∼/lsbpt/src* ", then follow the instructions to configure and compile the project.

# 3 Run a segmentation

Luanch the application simply by double click or with command line *path/to/the/application/LSBPT*, an interface is showed in figure 1. Click the first "..." button to select a image you want to segment and click second "..." to select a directory where you want to save the results, and then click button "Go", a segmentation is running. When it finishes, a dialogue like figure 2 pops up reminding you the segmentation has been done.



full/path/image.tif

out/put/directory

StartX 0 StartY 0

Width 0 Height 0

nBlocksX 1 nBlocksY 1

☐ Delete noise

nStdLeft 2.0

nStdRight 2.0

☐ if add probability model

probabilities directory

probabilities/directory

Probability Model weight 0.00

N. classes: 2

☑ use Std cut

Num. of bins 30

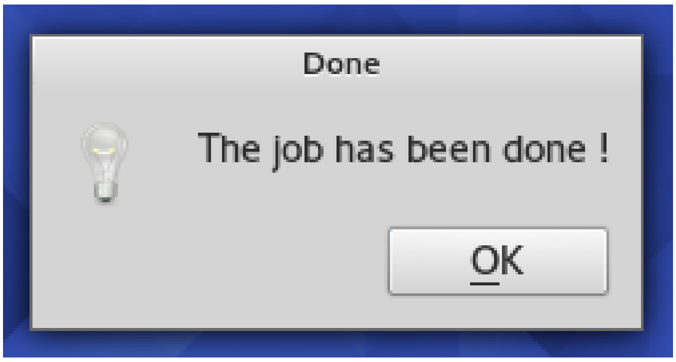
☐ Area weighted Area exponent: 0.00000

☐ Min. to merge Min. size to merge: 30

Segmentation Reg 10000

☑ Paint with random color

Figure 1: Software interface



The job has been done !

Figure 2: Dialogue of finishing

### 3.1 Processing a part of the image

It is possible to process any rectangular part of the image. The input "StartX" and "StartY" indicate the coordinates of left-up corner pixel of the rectangle part of horizontal and vertical direction respectively. Coordinates are set in up-down, left-right manner ( (0,0) at the left-up corner pixel of the image). "Width" and "Height" indicate the size of the rectangle you want to process. **We use 0 to indicate the rest part of the image start from "StartX" and "StartY"**.

### 3.2 Processing with tiling strategy

"nBlocksX" and "nBlocksY" indicates the number of tiles to split the image into in X and Y direction respectively. The image is split evenly, except that tiles in the last column and last row will may little bit larger if the size of the image can't be divided by the number of the tiles, please reference *blockSizeCalculator_evenSplite* function in "image" class for details.

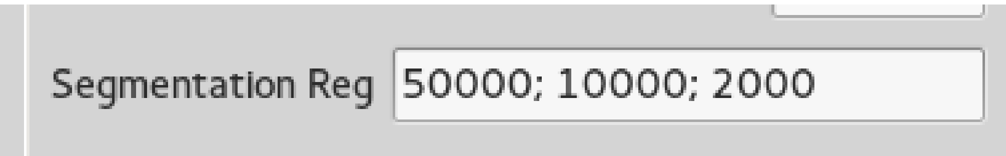
### 3.3 Delete outliers

If you want to delete some outliers, check the check box "Delete noise", "nStdLeft" and "nStdRight" indicates the number of standard deviation on each side. In this case, for each channel, pixel intensities which are larger than "mean + nStdLeft * sigma" will be set to "mean + nStdRight * sigma " and pixel intensities which are smaller than "mean - nStdLeft * sigma" will be set to "mean - nStdLeft * sigma", here "mean" is the mean value of the pixel intensities of the channel , "sigma" is the standard deviation.

### 3.4 Parameters for constructing BPTs and segmentation

- Num. of bins Indicates the number of bins of the histogram in histogram model

- Area weighted If use area when compute the dissimilarities between regions, check *getOrder()* function in "compoundModel" class. This term is for balancing the tree.



Segmentation Reg 50000; 10000; 2000

Figure 3: Input regularization term for multi-scale segmentation

- Area exponent The strength of using area to balance the tree
- Min. to merge (check box) If use minimum size to merge strategy
- Min. to merge If use minimum size to merge strategy, then regions which are smaller than this number will merge first. This strategy is also for balancing the tree, and it's also effective to eliminate some small segments
- Segmentation Reg Segmentation ( Classification ) regularization term.

### 3.5 Multi-scale segmentation and classification

LSBPT can extract multi-level segmentation or classification at the same time. The level is controlled by the regularization term ( **Segmentation Reg** ). Several regularization term can be used at the same time, separated with ";" an example showed in 3. An descending order is recommended though not mandatory. Results will be save with names "segmentation+level.tif". "level" is set starting from 0. For example, if the regularization term is set to be "10000; 50000; 2000", then "segmentation0.tif" corresponding to 50000, and "segmentation1.tif corresponding to 10000, and "segmentation2.tif" corresponding to 2000.

### 3.6 Add probability model

When probabilities are available, we can add probability model. Double floats are used for probabilities, and they are written into binary file row-wise. Some example of the probability files are showed in "test/input/probabilities", check loadProbabilities() function in "imageblock" class for more details.. For tiling strategy, each tiles has a probability file, and it is named as "i+indexX+j+indexY.dat" ( for example for a tile in the second column and third row, the name would be "i1j2.dat").

To add probability model, check the check box " if add probability model ", and select the directory where the probability files are saved. The other parameters are :

- Probability Model weight This parameter indicates the importance of the probability model
- N. classes Number of classes