Quantification of Various Properties of Objects (MTAT.03.260)

Hannes Tamme

2011

Abstract

In this paper we describe a method (program) for extracting various objects properties from picture. We describe a tool-set for image manipulation, object recognition and properties extraction.

Keywords: object recognition , object properties

Introduction

In order to process the large quantity of images we need to provide labels for details in the picture. To achieve this goal we perform various image manipulation and pattern, shape recognition tasks.

Installation

Installation from source is recommended. If we need thirty second solution for 32bit linux then run:

source INSTALL

This call sets environmental variable LD_LIBRARY_PATH pointing to lib directory and PATH variable to bin. Change directory to *bin* and run helper script

sh ts_user_run

Also call man page with

man ./ts

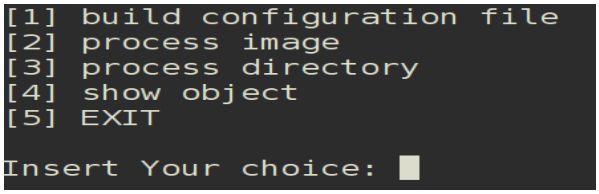
from *doc* directory for further information.

Check opencv [1] , boost [2] , Qt Creator [3] and xdotool [4] documentation.

Image classifier

We need describe, define what is exactly object and background. Because object can be basically any shape we can not use (easily) any known machine learning algorithms. Only absolute known properties of object is its sharp division line between object and background.

In order to further increase contrast between object and background we manipulate image regions so that the background is WHITE and the object color is something else. All performed steps are recorded to file. We can use that file later to perform exactly the same manipulation for all given picture in the set.
Communication between the user and the program is performed by helper script ts_user_run (black window) or directly by giving command line parameters to program.



For image manipulation (configuration file building) we have graphical tool. Invoking this tool ts in image configuration mode is performed as follows:

ts -i image-file

for that we can also use helper script ts_user_run.
User interface for this tool:

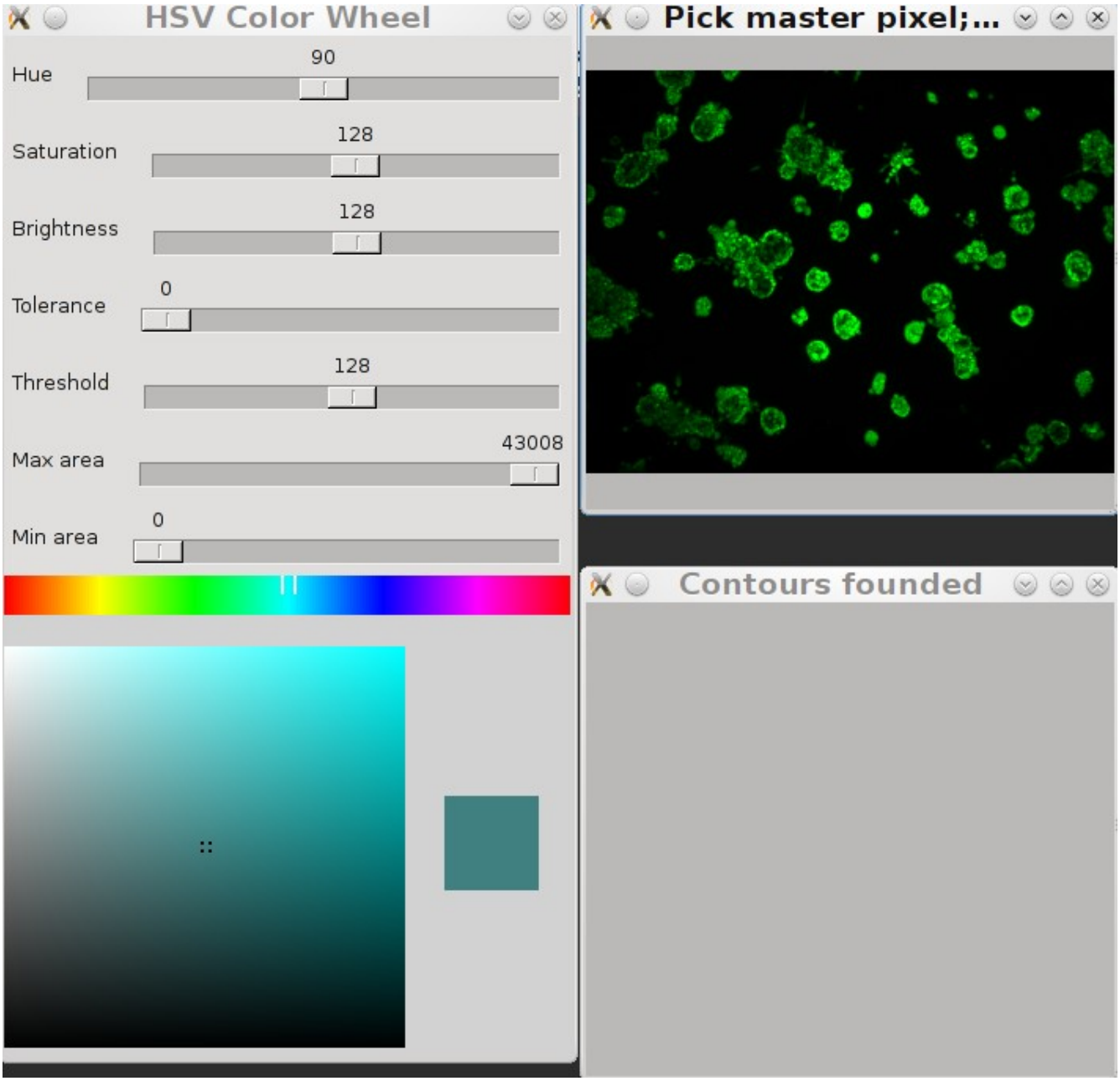


Image manipulation is carried out as follows:

1) left mouse click on image region that we want to adjust
2) adjust rulers on color wheel window until we achieve WHITE background
3) repeat step one and two on some other image region as long as needed
4) if modification done hit key “s” as save and “Esc” to quit

The result of this operation is configuration file named conf.csv. We can find it under the folder results. This file is used if we process image (viewing an image and extracting various objects properties from picture). If we have unrealistic quantity of founded objects then first thing is to check configuration file content or existence.

Window, named HSV color wheel, is used for adjusting pixels on our training picture. Lets go through all adjustments rulers properties.

Hue: It is basically pixel color. When we select some pixel from our training picture this ruler is set automatically to appropriate value.

Saturation: It shows how clean is color. If We want white color then we adjust this ruler towards to minimum and Brightness ruler to maximum.

Brightness: It is opposite darkness.

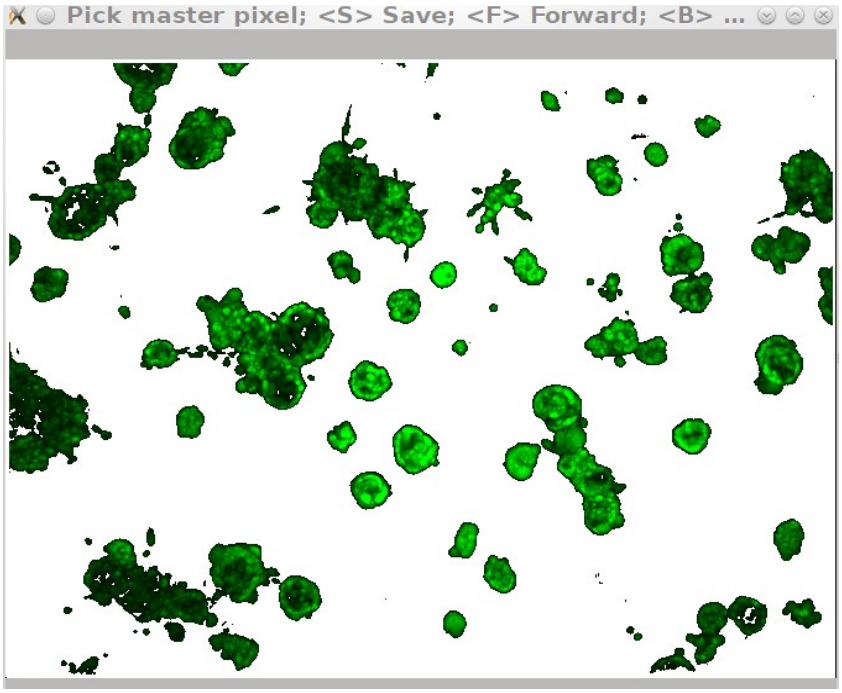
Tolerance: Here we set our limits. How much pixel value can differ from selected pixel. All pixels within limits are subject for manipulation.

Threshold : Before we can detect objects we alter all pixels below threshold value white and all pixels above threshold to black. Appropriate level varies from picture to picture. This value is guidance for adaptive threshold inside the program.

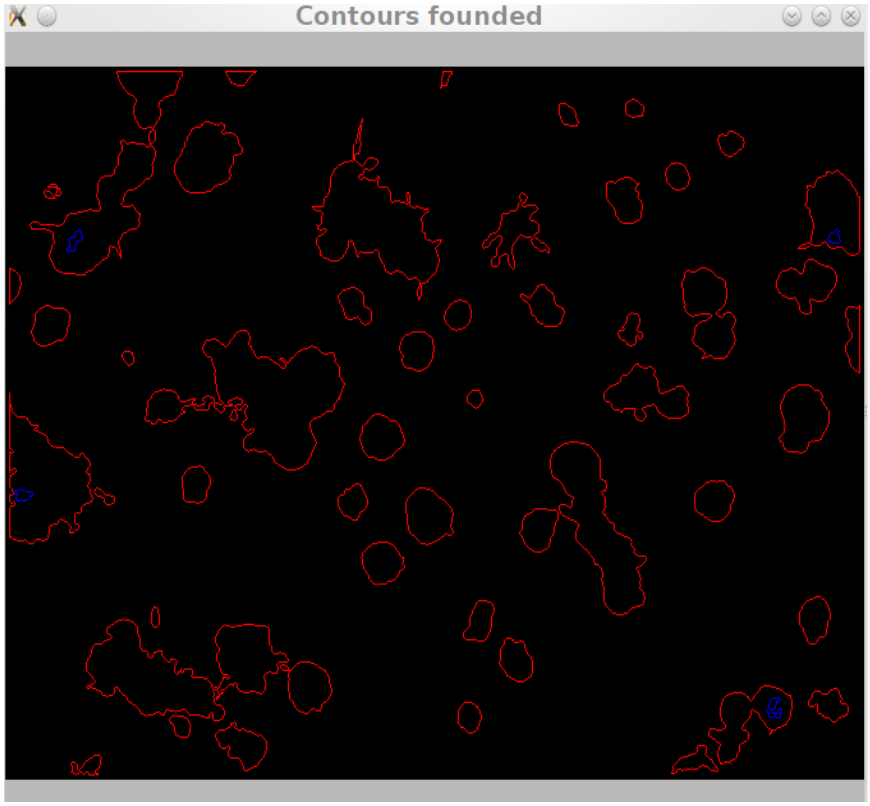
Max area: It is the maximum area of objects in pixels.

Min area: It is the minimum area of objects in pixel. This value is used mostly to remove small objects from output. For example if we do not want 2 pixel size objects properties. These small objects tend to be false alarm or bad configuration file indication.

If our adjustments are not visually good we can reset our picture pressing key “b” as back. If all result on Contours founded window is acceptable then press key “s” as save. This also automatically resets our image. After that you can also see on Contours founded window how bad is output without various methods combination. Next picture shows how picture shod view after transformation:



Objects founded on picture is displayed on Objects founded window. If you move from Image window or Objects founded window then window is re-sized. We trigger resizing feature (except left hand side) with our mouse. Next picture shows acceptable output:



At the end of our process we have filled our configuration file with our wishes. The complexity of this file depends about of our strategy.

R	G	B	hue	saturation		brightness		tolerance		threshold	max_area	min_area
0	1	0	60	0	255	2	128	43008	0			
0	11	0	60	0	255	5	102	43008	47			

Image (directory of images) processing

For image processing ia. object labeling we have command line tool that can invoked like that:

ts -c conf.csv -i image-file -o output-format -v no

ts -c conf.csv -d image-directory -o output-format -v no

Or using helper script ts_user_run . These commands generates file that contains objects descriptors. These files are locating under results directory.

Program output format is selectable. Format **standard** gives human readable object descriptors (area, circumference, location, direction, color). Format **raw** gives object x,y coordinates on cartesian coordinate system. Coordinate zero point is upper left corner.

If we select image with small objects (4-5 pixels) the output may be massive. If yo have approximately 5000 tiny objects on output please consider build a new classifier. Your text editor can't handle it.

Standard output in file shod look something like this:

file_name	(obj_nr)	area	circumference	orientation (deg)	center_x		center_y		Obj_wavelength (nm)
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(1)	186	71	22	63	500	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(2)	326	94	1	538	505	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(3)	772	116	22	183	486	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(4)	183	49	57	136	472	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(5)	300	59	90	359	467	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(6)	86	55	69	600	459	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(7)	445	89	14	643	458	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(8)	1711	234	40	585	474	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(9)	974	125	48	230	445	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(10)	585	86	69	397	425	505		
/home/h/mnt/pilditootlus/projekt/pic/position_1.tif	(11)	5805	553	6	142	434	505		

On standard output we can see object index, location, area, circumference, orientation, center, and average color in nm.

Raw output in file looks like this:

/home/h/mnt/pilditootlus/projekt/pic/position_1.tif

x1,y1	x2,y2	x3,y3	x4,y4	x5,y5	x6,y6	x7,y7	x8,y8	x9,y9	x10,y10	x11,y11
69,494	542,487	179,470	131,465	359,456	603,453	648,446	593,444	230,427	392,410	92,396
73,494	547,487	180,471	132,466	364,456	600,453	650,446	598,444	238,427	395,410	93,397
75,496	548,488	180,473	133,465	365,457	598,455	652,448	599,445	239,428	396,411	97,397
74,497	548,492	179,474	134,466	366,457	598,456	652,450	602,445	241,428	397,411	98,398
74,501	547,493	180,475	139,466	368,459	597,457	653,451	603,446	242,429	397,412	99,398
73,502	551,497	181,475	140,467	369,459	597,460	653,452	605,446	243,429	398,413	103,402
73,503	552,497	182,476	141,467	369,460	595,462	654,453	606,447	245,431	401,413	103,407
72,504	553,498	184,476	143,469	371,462	597,464	654,454	607,447	246,431	402,412	104,408

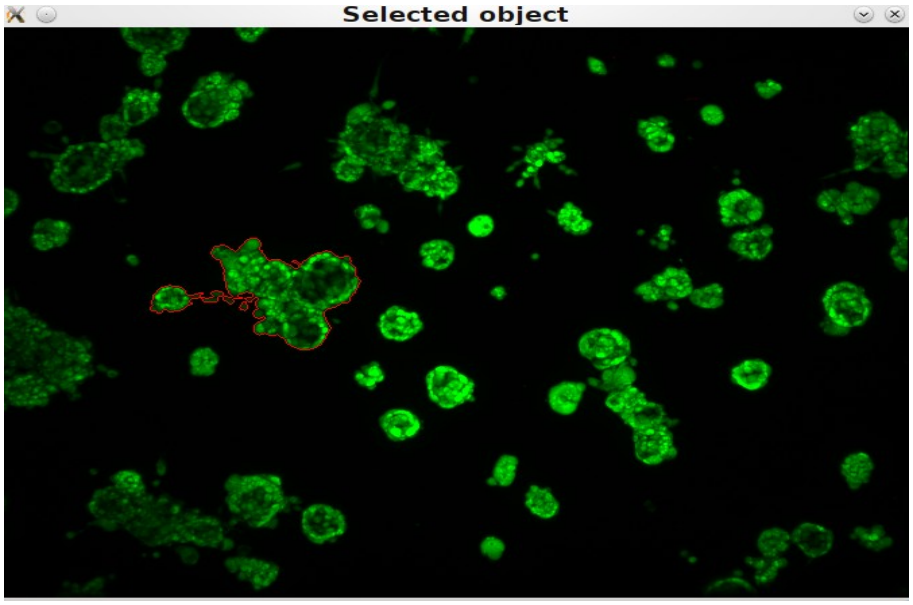
We can use raw output for rebuilding object of interest.

Object of interest

If we want to watch one particular object we can feed our program with following parameters:

ts -c conf.csv -i image-file -s object-number -***v no***

Or using helper script ts_user_run. Desired object is surrounded by narrow red line.



Program internal logic

Brief research shows that current task is not easily resolvable using some machine learning algorithm.

The problem is mainly unmanageable object shape, location, color variation. Only fully qualified property is object and background gradient. Before shape detection we tray to amplify this difference. Our goal is maximize differences between object and background.

For that we select one average background pixel and transform it to something that looks white. Simple trackbar moving includes several internal tasks (histogram equalization, smoothing with different kernels, thresholding and so on). We must use several methods founded by heuristics for acceptable final results. Image is scanned and transformed pixel by pixel. If you select verbose output then processing time roughly doubles.

Good enough result is if background is white and object color is something else. After that we can use

object detection alghoritm (cvFindContours). Input shoud be an 8-bit single channel image. Non-zero pixels are treated as 1's, zero pixels remain 0's - the image is treated as binary. To get such a binary image from grayscale, one we use T*hreshold*, *AdaptiveThreshold* and our own heuristic methods for image manipulation. The function retrieves contours from the binary image and returns the number of retrieved contours. The pointer first_contour is filled by the function. It will contain a pointer to the first outermost contour or NULL if no contours are detected (if the image is completely black).

Result (objects cordinates) is stored onto list.

References

[1] http://opencv.willowgarage.com/wiki/Welcome

[2] http://www.boost.org/

[3] http://qt.nokia.com/products/developer-tools

[4] http://www.linuxcertif.com/man/1/xdotool/

Appendix

ts(7) **User Manual** ts(7)

NAME

ts - quantification various properties of objects

SYNOPSIS

ts [-cidho] [-c config-file] [-i image-file] [-d image-directory] [-o output-format(row/standard)]

DESCRIPTION

ts quantificate various properties of objects. Quantification is performed in accordance with config-file. config-file is created by user using only -i switch. Program output is selectable - row (only coordinates (Cartesian)) or standard (various objects attributes).

OPTIONS

-i process image config-file performing varios image transformation, processing, maching and segmentation routines . Produce config-file or row, standard image processing output.

-c config file where is described all image modification actions. Use absoute path for you config-file.

-h program help.

-o specify output format row or standard

-d process image-directory

EXAMPLES

ts -i image-file

fabricate config-file

ts -c conf.csv **-i** image-file **-o** output-format

process image file.

ts -c conf.csv **-d** image-directory **-o** output-format

process all images inside directory.

ts -h

show help.

FILES

conf.csv

The configuration file. See -h for further details.

DIAGNOSTICS

The following diagnostics may be issued on stderr:

File does not exist.

The input file does not look like config-file or folder path contains no images.

ts can only handle appropriate type formats created by ts -i

Various output containing information about image processing

BUGS

under some circumstances we can build unrealistic configuration file

TODO

more features , easy installation ...

AUTHOR

Hannes Tamme <htamme at ut dot ee>

SEE ALSO

opencv, boost, qt