**ALGORITHM TO RECOGNIZE LINES AND CURVES.**

The algorithm uses several computer vision routines, plus computational geometry constructs to detect lines and curves.

The algorithm uses the following steps:

1. It preprocesses the image with a bilateral filter. The documentation for the filter and the parameters can be found in <http://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html>. The default parameters used for the filter are: diameter: 5, sigmaColor: 50, sigmaSpace: 50.
2. It detects edges using Canny edge detection. The documentation for the Canny edge detection can be found in <http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html>. The default parameters used for the filter are: lowthreshold: 20, ratio: 6. These parameters work well because of the bilateral filter, even when they would not be recommended with a normal noisy image.
3. It creates a relative neighborhood graph. The relative neighborhood graph is created using the Delaunay triangulation and an algorithm that could be improved in speed, as it checks all the edges tied to this. The relative neighborhood graph shows a feature independent shape of the object. However, as it contains the euclidean minimum spanning tree, it will be connected, while the lines of CADs are not usually connected. A documentation of the relative neighborhood graph can be found here: <https://en.wikipedia.org/wiki/Relative_neighborhood_graph>.
4. It filters edges when the two nodes of the edges contain more than 2 edges each, and this edge is longer than the two smaller edges. That breaks the minimum spanning tree connectedness property so that it can better represent a CAD.
5. It separates the lines at either a point with more than two edges, or the end of a line.
6. It tries to fit windows of lines, if it finds a fit for that window with a straight line y = c1\*x + c0, it tries to search back and forward until it doesn’t fit anymore.
7. It tries to do the same but with a curve of the form y = c3\*x^3 + c2\*x^2 + c2\*x +c0. If the fit is much better than with the straight line, as determined by the **line-ratio** parameter, it then moves the window a third, holding part of the previous window, because the curve can move faster, and the same equation would not work for the new points. It does this until it does not fit anymore. The parameters for the line curve fit are, the **line-ratio**=2, the **line-window**=40 and the **line-maxfit**=2.

**Installation:**

The following are the steps to install in Ubuntu.

# Install required ubuntu packages.

sudo apt-get install build-essential cmake

git pkg-config

sudo apt-get install libjpeg8-dev libtiff4-dev libjasper-dev libpng12-dev

sudo apt-get install libgtk2.0-dev

sudo apt-get install libatlas-base-dev gfortran ç

#Create an opencv dir and download the latest version.

mkdir opencv\_src cd opencv\_src/

git clone https://github.com/Itseez/opencv.git

cd opencv/

# Cmake it.

mkdir release

cd release/

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local ..

make

# Install it in env/local/lib

sudo make install

# Link it to site-packages

cd env/lib/python2.7/site-packages/ ln -s /usr/local/lib/python2.7/dist-packages/cv2.so .

# Finally create the virutalenv, and install numpy and Pillow

virtualenv env

source env/bin/activate

pip install numpy Pillow

**Usage:**

usage: detect\_lines.py [-h] [--no-bilateral]

[--bilateral-diameter BILATERAL\_DIAMETER]

[--bilateral-sspace BILATERAL\_SSPACE]

[--bilateral-scolor BILATERAL\_SCOLOR] [--no-normal]

[--normal-cut NORMAL\_CUT] [--no-full]

[--full-cut FULL\_CUT]

[--canny-minthresh CANNY\_MINTHRESH]

[--canny-ratio CANNY\_RATIO] [--line-window LINE\_WINDOW]

[--line-ratio LINE\_RATIO] [--line-maxfit LINE\_MAXFIT]

[--debug]

input output

detect\_lines: It finds a set of lines and curves in a picture of an 3D printed

object and adds them to the image. The output of the command is a JSON in the

form {'curve': [], 'straight': [], 'other': []} All of them are sets of points

representing either the straight line, the curve, or the unclassified line.

positional arguments:

input input image file

output output image file

optional arguments:

-h, --help show this help message and exit

--no-bilateral Don't run the bilateral filter

--bilateral-diameter BILATERAL\_DIAMETER

The diameter of the bilateral filter

--bilateral-sspace BILATERAL\_SSPACE

The sigma color of the bilateral filter

--bilateral-scolor BILATERAL\_SCOLOR

The sigma space of the bilateral filter

--no-normal Don't remove edges with the normal distribution filter

--normal-cut NORMAL\_CUT

The cut point to remove edges by size

--no-full Don't remove edges with the full edge filter

--full-cut FULL\_CUT The cut ratio between the sum of the smaller edges and

the larger edge

--canny-minthresh CANNY\_MINTHRESH

Lower threshold of the Canny filter

--canny-ratio CANNY\_RATIO

Ratio between the lower and upper threshold of the

canny filter

--line-window LINE\_WINDOW

Window to search for the line to fit with curve or

straight line

--line-ratio LINE\_RATIO

Number of times curve fit should be better than

straight line fit

--line-maxfit LINE\_MAXFIT

Maxmimum least squares average to declare a fit

--debug Opens a window at each step showing what's happening

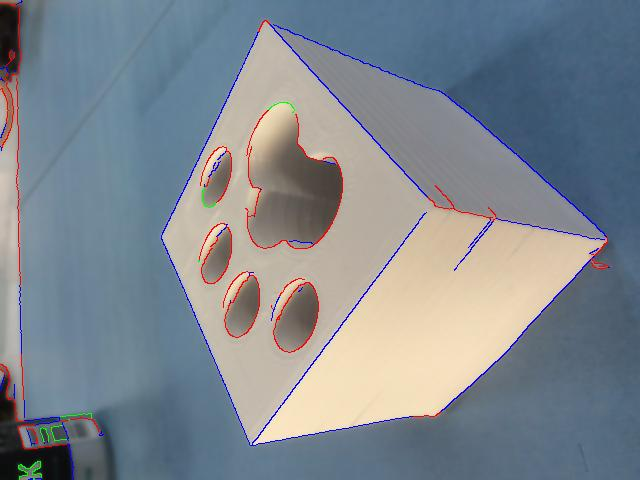
with the algorithm

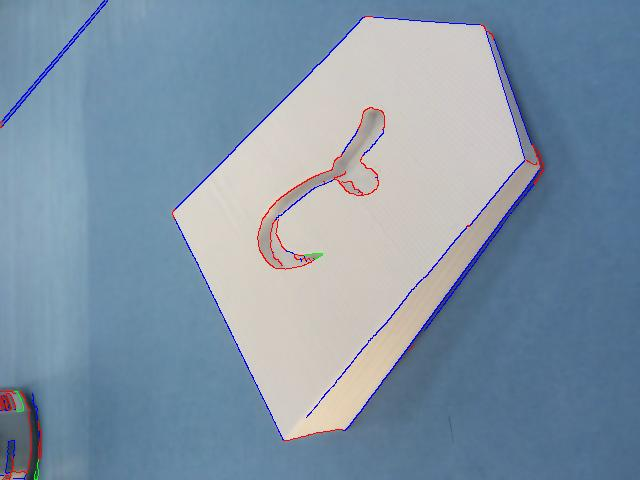
The program outputs the lines, classified in straights, curves, and other in JSON format. The JSON format printed is like this:

{ “curve” : [[[x1,y1], [x2,y2], [x3,y3]], [] …], “straight” : [[[x1,y1], [x2,y2], [x3,y3]], [] …], “other” : [[[x1,y1], [x2,y2], [x3,y3]], [] …]}

Redirect to file to improve performance and reduce output.

**Example Images:** Blue straight, Read curve, Green non-classified**.**







The program does not seem to work that well with the paper objects. Some of the reasons are that some edges in the paper are folds, which generates a lot of noise. As the main reason for the application is CAD outputs, that don’t have folds, I think it was fair to target the efforts to the CAD models.