Measuring Shape: Rectangularity

student: Jan Ignatowicz
supervisor: dr Krzysztof Misztal
Institute of Computer Science and Mathematics,
Jagiellonian University

project carried out as part of the subject ‘Individual project’;
extends: https://github.com/kmisztal/py-shape-descriptors

Abstract

This paper presents the results of experiments made on the implemented rectangularity measurement methods. There are five methods presented. The methods are implemented in Python3 with OpenCV framework. The methods are taken from papers provided in the bibliography. Examples for testing are taken from MPEG7 dataset.

1. Introduction

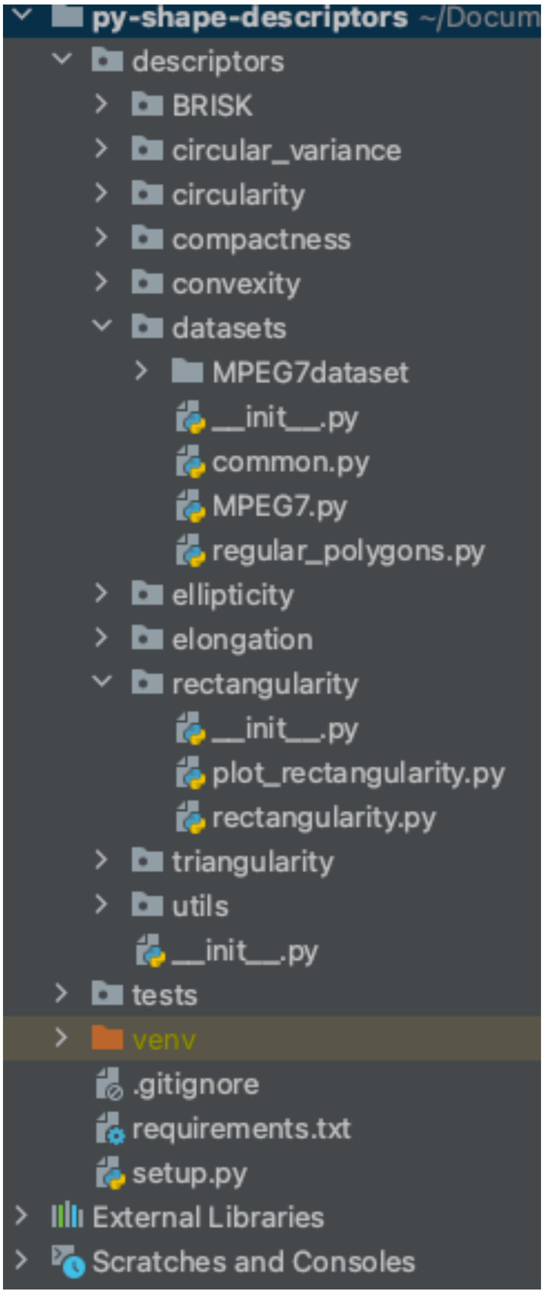
Image processing deals with many problems, including those related to the classification and finding shapes in an image. Shapes can have plenty of features such as size, pattern, colour, the degree of perturbation or blur. These features can be mapped as numbers to objects named feature vectors. Based on those vectors the comparisons between shapes or whole images may be processed.

Experiments presented in this paper focus only on the shape feature of figure.
In order to search for shapes, there is a possibility to use their descriptors. The usage of normalised descriptors can help in well determining the shape of an image.

The methods and algorithms are discovered to measure the rectangularity level of found shape. Most of these approaches are based on comparisons and ratios of calculated areas. Some of them may use the shapes moments or sides.

A direct example of using a descriptor for rectangularity is the pattern recognition in images. Different objects appear on images from a various of domains such as medicine, industry, biology or astronomy. These objects may be classified, identified oraz recognised.

2. Project structure



Project is an extension to actual existing project https://github.com/kmisztal/py-shape-descriptors.

The whole project contains the implementation of several shape descriptors.

Implemented methods for rectangularity can be found in file:

descriptors/rectangularity/rectangularity.py

Experiments were made with usage of the file:

descriptors/rectangularity/plot_rectangularity.py

Under directory:

descriptors/datasets

there can be found files responsible for:
- downloading MPEG7 dataset from website https://dabi.temple.edu/external/shape/MPEG7/MPEG7dataset.zip
- loading images for experiments

Under directory:

descriptors/utils

There are useful functions for all shape descriptors. In rectangularity case there is only one common function in usage. This function is used to calculate moments of the shape and can be found in the file:

descriptors/utils/moments/moments.py

3. Implemented rectangularity Measures

1. **Minimum Bounding Rectangle (MBR) - R_B**
   The rectangularity descriptor is implemented as simple ratio of the region area and minimum bounding rectangle.

$$R_B = \frac{Area(S)}{Area(MBR(S))}$$

2. **Rectangular Dicrepancy - R_D**
   Rectangularity is measured as the normalised discrepancies between areas of the whole region and the bounding rectangle. Bounding rectangle is calculated with usage of the OpenCV framework.

$$R_D = 1 - \frac{R + D}{B}$$

   R - difference between rectangle and clipped region
   D - difference between the whole region and clipped region
   B - rectangle area

   According to information given in papers the descriptor is the result of comparison two descriptors. One is calculated for the given image and the second is calculated for the rotated given image with 45 degrees rotation. The major value is returned as the rectangularity descriptor.

   Due to some preprocessing of image errors the condition B != 0 is provided. Also, for better measures, the absolute of descriptors provided as the result. This addition is made according to the results obtained while testing the method.

3. **Robust MBR - R_R**
   This method is quite similar to the Rectangular Discrepancy with one modification;
   the rectangle area in denominator is replaced with 'I' the intersection of rectangle area and the region.

$$R_R = 1 - \frac{R + D}{I}$$

4. **Agreement method - R_A**
   In this method the idea is to assume that the given image is truly a rectangle and to calculate the rectangle sides in two different ways. The both calculated sides should agree, in that case.

$$R = \frac{|a_1 - a_2|}{a_1 + a_2} + \frac{|a_1 - a_2|}{a_1 + a_2}$$

   The sides are calculated as follows:

$$a_1 = \sqrt{3}\alpha = \sqrt{\frac{6[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]}{\mu_{00}}}$$

$$b_1 = \sqrt{3}\beta = \sqrt{\frac{6[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}]}{\mu_{00}}}.$$

$$a_2 = \frac{P \pm \sqrt{P^2 - 16A}}{4}$$
$$b_2 = \frac{A}{a_2}$$

   The resulted descriptor must be processed as follows in order to make it comparable with other descriptors:

$$R_A = 1 - \frac{R}{2}$$

5. **Moments method - R_M**
   The last method calculates the rectangularity descriptor based on shape moments.

$$R_M = \begin{cases} R & \text{if } R \leq 1 \\ \frac{1}{R} & \text{otherwise} \end{cases}.$$

$$R = 144\frac{mu_{22}}{m_{00}^3}$$

4. Experiments

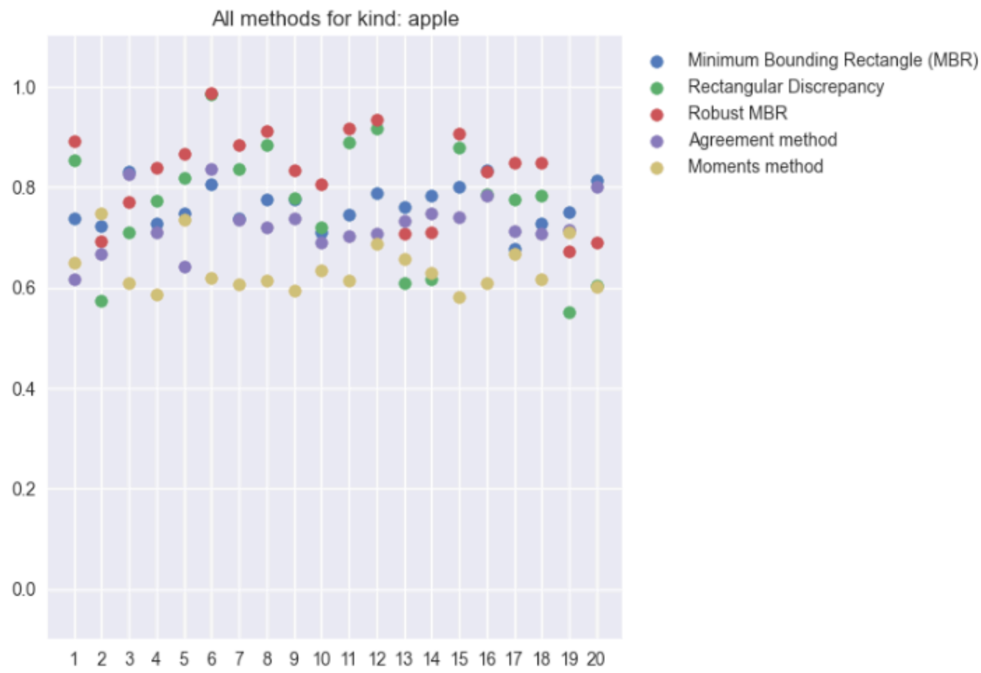
The MPEG7 dataset contains various kinds of shapes. For each kind there are provided exactly 20 images. All images are black and white.

Images from the series: ‘apple’, ‘bottle’, ‘cellular_phone’, ‘Bone’, ‘camel’ were taken for the rectangularity descriptor measuring experiments.

Both ‘bottle’ and ‘cellular_phone’ have more or less rectangle shape.
The shape of ‘apple’ is more likely to circular.
The ‘Bone’ examples mostly looks like the rectangle with round appendices at the ends. In datasets these shapes are also rotated.
The ‘camel’ examples are less solid than the previous. They contain many deformations due to plain polygons.

Kinds:

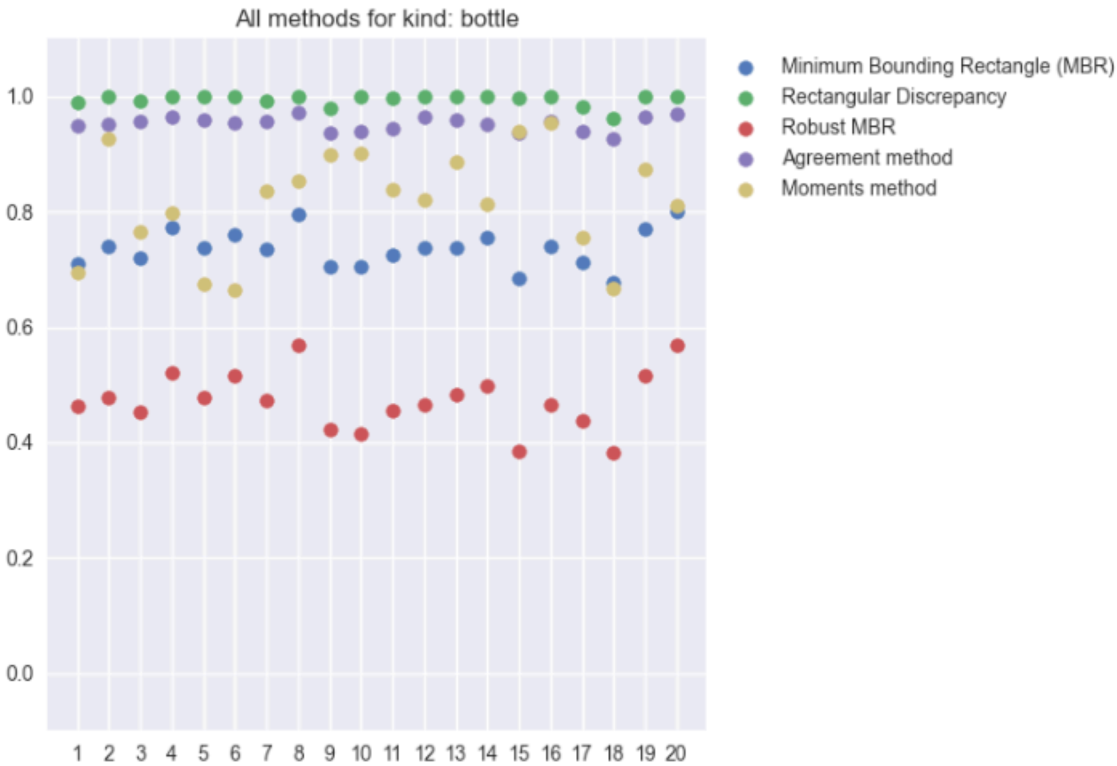
1. The rectangularity descriptor results for kind ‘apple’:

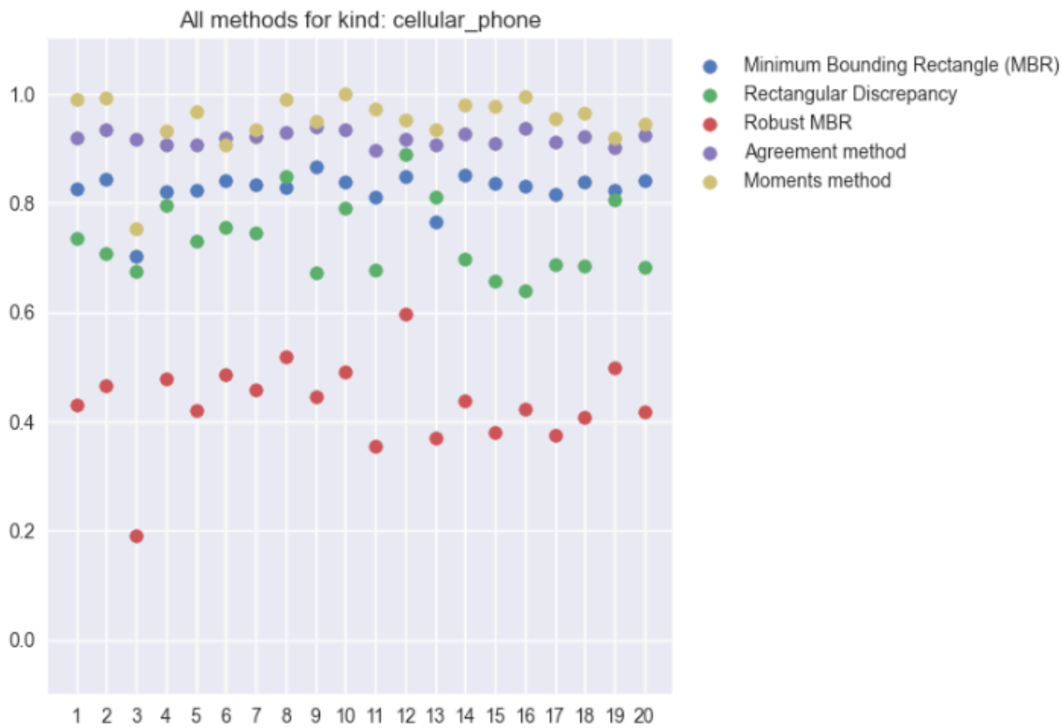


   For circular shape the results of all methods hesitate between 60% to 90% (most: up to 80%). This is quite a big percentage. However, all methods operates on the rectangles that should cover the majority of the given shape.

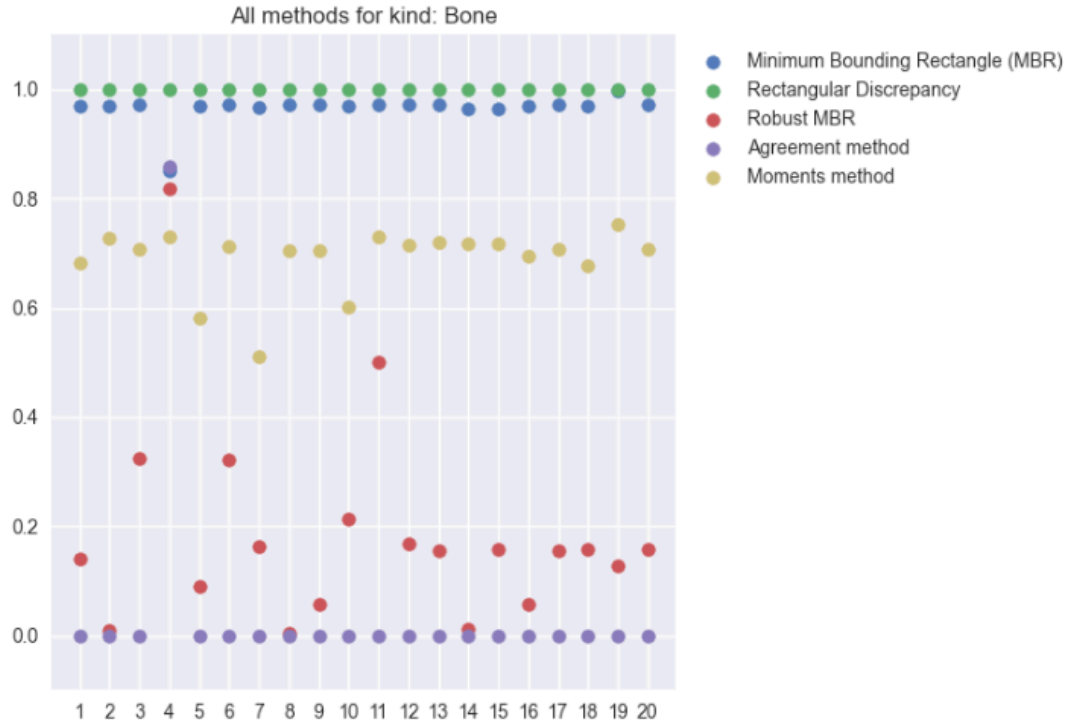
2. The rectangularity descriptor results for kind ‘cellular_phone’ and ‘bottle’:
   Both shapes are similar to rectangle, but the images from ‘cellular_phone’ in most examples have some protruding part of the rectangle and the images from ‘bottle’ end with emaciation of the neck on the upper edge.
   For ‘bottle’ shapes the R_B and R_A methods give the result near to 100%.
   For shape like ‘cellular_phone’ the methods R_M and R_A catch the shape closely to a rectangle

   For both, the discriminator R_R denies to existence of rectangle, giving only 40% up to 50%. In general it seem to be wrong, but the R_R method operates on intersections of the areas, so here all the protruding elements are omitted.

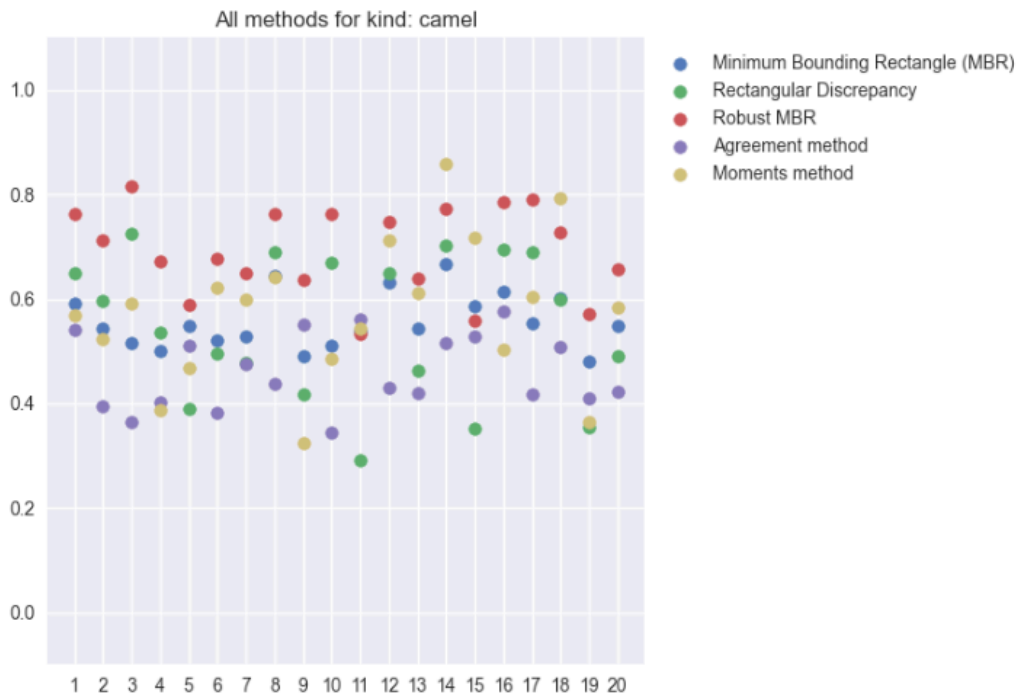

All methods for kind: bottle
1.0
0.8
0.6
0.4
0.2
0.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Minimum Bounding Rectangle (MBR)
Rectangular Discrepancy
Robust MBR
Agreement method
Moments method


All methods for kind: cellular_phone
1.0
0.8
0.6
0.4
0.2
0.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Minimum Bounding Rectangle (MBR)
Rectangular Discrepancy
Robust MBR
Agreement method
Moments method

3. ‘Bones’ are a specific examples. These images can distinguish the methods to be invariant and to not be invariant. The implemented agreement method cannot handle with rotated bone to diagonal.
Despite some protruding elements the M_B and M_D methods diagnose the shape to be a rectangular.



4. ‘Camels’ are specific due to theirs misshapen shapes. The protruding head and humps definitely deny the classification of the shape as rectangular. All the methods mostly hesitate with their results from 50% up to 70%.



5. Conclusions

In this paper the tests results of the extensive list of implemented algorithms for measuring rectangularity is provided.

Invariance to rotation:
This conclusion is mostly made by testing the ‘Bones’ images.
For five presented methods only one seem to not be invariant to rotations, this is an Agreement method.

Invariance to scaling was not tested during the experiments.

In papers from research (provided in reference section) there is claim that Minimum Bounding Rectangle and Rectangular Discrepancy methods are the best classifiers for rectangular.
In case of correcting equations to be more resistant for rotations there can be added an Agreement method as a good classifier as well.

6. References

1. https://github.com/kmisztal/py-shape-descriptors
2. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.160.506&rep=rep1&type=pdf
3. https://www.researchgate.net/profile/Paul_Rosin/publication/227273599_Measuring_rectangularity/links/555e1cef08ae6f4dcc8dd1a9/Measuring-rectangularity.pdf
4. https://www.researchgate.net/publication/263909418_Shape_rectangularity_measures
5. https://opencv.org/
6. https://dabi.temple.edu/external/shape/MPEG7/MPEG7dataset.zip