# FlakeAutoFind

Charles Yang

Last updates: October 31, 2020

## Contents

**1 General Algorithm** **1**

**2 Flattening** **1**

## 1 General Algorithm

The image is flattened using an infered background level. After flattening, morphological transformations are applied. Opening reduces the amount of small, white pixels, eroding reduces the size of bigger clumps of white pixels, and closing makes the flakes more solid and bold.

## 2 Flattening

The symmetry of the background is assumed to that of an ellipse. Similar to a circle, where the radius is defined

$$s^2 = (x - x_c)^2 + (y - y_c)^2$$

we define a distance coordinate

$$s^2 = (x - x_c)^2 + r^2(y - y_c)^2$$

where $r$ is the aspect ratio of the ellipse, given

$$r = \frac{\text{width}}{\text{height}}$$

In this way, we consider the background to be a function of the elliptical distance. To maintain smoothness, the background is modeled as the sum of even terms:

$$B_N(s) = \sum_{n=0}^{N} a_n s^{2n}$$

Clearly, the first $n$ coefficients can be solved for using a system of linear equations generated by $n$ samples of the background. This yields the equation

$$B = SA$$

where $B$ is a vector of background values, $A$ is a vector of the coefficients, and $S$ is the Vandermonde matrix of the values $s_i^2$.

Once the coefficients are determined, the equation can be determined recursively as:

$$f_0 = a_N$$

$$f_n = s^2 f_{n-1} + a_{N-n}$$

with

$$B_N(s) = f_N$$

For simplicity, we consider only the lowest two orders,

$$B_2(s) = a_0 + a_1 s^2$$

Taking two samples,

$$B_2(s_1) = a_0 + a_1 s_1^2$$

$$B_2(s_2) = a_0 + a_1 s_2^2$$

$$B_2(s_2) - B_2(s_1) = a_1(s_2^2 - s_1^2)$$

$$a_1 = \frac{B_2(s_2) - B_2(s_1)}{s_2^2 - s_1^2}$$

$$a_0 = B_2(s_1) - a_1 s_1^2 = B_2(s_2) - a_1 s_2^2$$