

## Defocus Stacking - Bokeh without aperture f/1.2

Herbert Kopp

The silver bullet for taking photos with a beautiful blurred background are lenses with a wide aperture. Unfortunately these lenses are expensive. Software generated bokeh does not come up to the bokeh produced by an optical lens and by now there were few attempts to find a way out.

In 2016 Thilo Gockel [2] introduced the term *Defocus Stacking* for a method which superposes a sharp image of the motif and a defocused image of the scene. It is a manual method which uses an exact mask of the motif to blend both images requiring additional manual corrections.

### The Defocus Stacking algorithm

Below a new defocus stacking method is presented, which is based on a picture stack (①). The stack includes a sharp picture of the motif at the top layer and additional pictures beneath which are increasingly defocused. When shooting the picture stack we could start putting the focus plane on the motif and then shift the focus plane nearer and nearer towards the camera. In figure 1 *level 0* is the top level of the stack and *level 3* is the lowest level

The algorithm proceeds as follows:

- For the sharp picture at *level 0* a mask ② is generated isolating the motif from the background. The mask must not be very exact, therefore automatic segmentation algorithms can be used.
- At first the mask is extended and blurred and then used for blending the pictures on *level 0* and *level 1* of the stack ③.
- The picture obtained by this blending operation is shown as a purple symbol ( ) in step ④ and used as the new 'foreground' picture in the next step.
- The mask is extended and blurred once more and used for blending with the picture on *level 2* of the stack ④.
- We repeat this for all following levels of the stack. In *figure 1* there is only one further step necessary for *level 3* ⑤.
- In a final step we blend the sharp image from level 0 with the result obtained from the preceding step using the original mask. ⑥

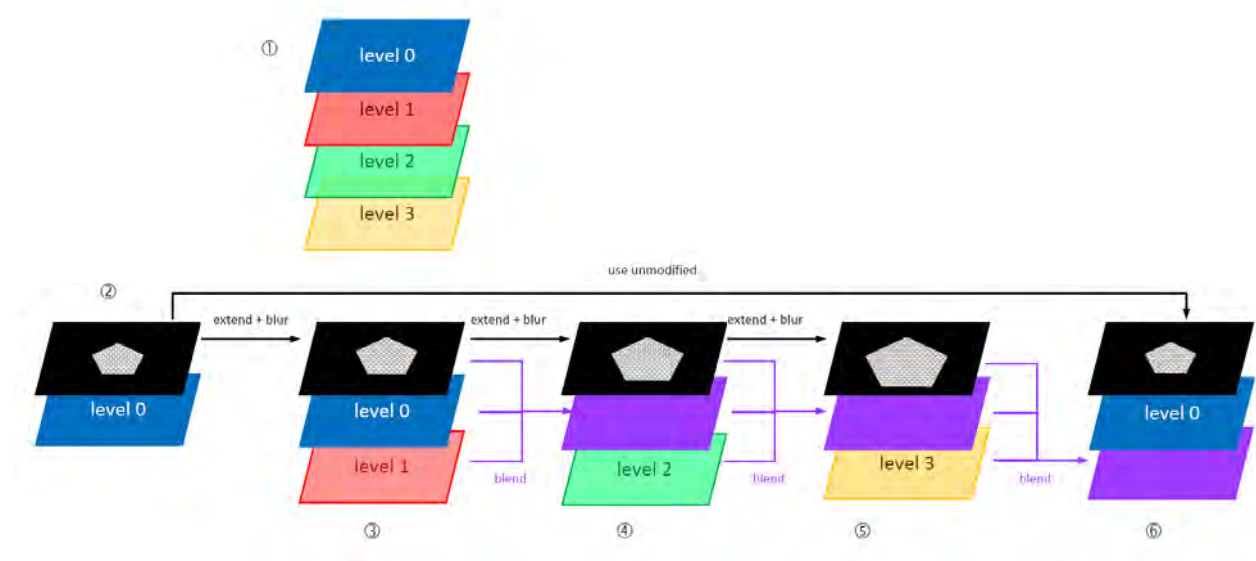


Figure 1: Schematic workflow of the defocus stacking process.

The process outlined above yields good results if the generated initial mask approximately isolates the motif and if the defocus steps between the levels of the stack are not too large. Since the mask is extended and blurred from level to level we get a smooth transition from the sharp motif to the fully blurred background of the lowest stack level even if the initial mask is not very precise.

### *The segmentation algorithms*

My implementation of the algorithm outlined above uses two automatic segmentation algorithms:

#### *GrabCut*

is the newer and more powerful algorithm. It is based on a concept from graph theory and was published by Rother, Kolmogorov and Blake [3]. All pixels of an image are arranged in a graph, where adjacent pixels are connected by edges. The edges are weighted by the similarity of the pixels at their vertices. The segmentation between foreground and background pixels is achieved by a graph theoretical method - a 'minimal cut': It decomposes the graph into two parts - foreground and background - by cutting its edges so that the weight sum of the cutted edges is minimal.

#### *Connected Components*

is based on the identification of blobs, i.e. connected subareas in the picture with similar attributes (channels of their color space, brightness, ...) [4]. The *DefocusStacker* uses the algorithm with only one attribute. By default it selects the largest blob as background. The assignment of the blobs to the foreground/background can be altered.

Figures 2 and 3 show sample results of the Defocus Stacking algorithm.



*sharp picture of the motif*



*GrabCut mask for the first level*



*stack with four images*

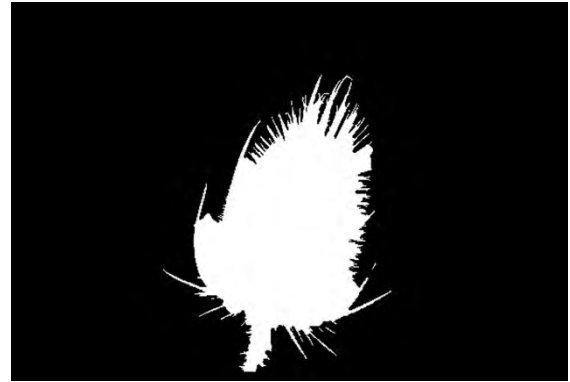
*result:*



*Figure 2: defocus stacking for a motif with distinct edges*



*sharp picture of the motif*



*GrabCut mask for the first level*



*stack with four images*



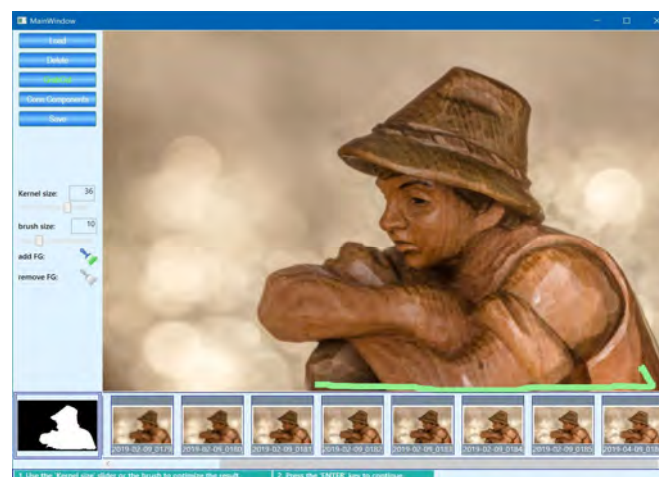
*result*

*Figure 3:* defocus stacking process for a motif with complex edges

### *The DefocusStacker implementation*

*DefocusStacker* is an implementation of the algorithm demonstrating the feasibility of the method. There are two versions [5], [6]:

- an implemetation for Windows 10 (Version 1909) shown in figure 4 and
- an implementation for Photoshop, tested with CS5 and CC2020. (an action plus a jsx- script).



*Figure 4:* user interface of the *DefocusStacker* application

#### 4. References

- [1] Gulbins, Gulbins: Photographic Multishot techniques, Rocky Nook-Verlag, 2009
- [2] Gockel: Bokeh, dpunkt-Verlag, 2016
- [3] Rother, Kolmogorov and Blake:  
GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts  
<https://cvg.ethz.ch/teaching/cvl/2012/grabcut-siggraph04.pdf>
- [4] AI Shack Tutorials: Connected Component Labelling  
<http://aishack.in/tutorials/pixel-neighbourhoods-connectedness/>
- [5] DefocusStacker (for Windows10)  
[https://www.dropbox.com/s/ucgvpf5yh33e1f2/DFS\\_Setup.zip?dl=0](https://www.dropbox.com/s/ucgvpf5yh33e1f2/DFS_Setup.zip?dl=0)
- [6] DefocusStacker(PS) (for Photoshop)  
<https://www.dropbox.com/s/4h896z36wimsfcr/DefocusStacker%28PS%29.zip?dl=0>