“Computer Vision”

Shervin Halat

98131018

Homework 6

Generally, StereoBM algorithm consists of three phases which are as follows:

1. Prefilter phase: prefilter image in order to normalize brightness and enhance texture of the image as the algorithm employs block matching method.
   This phase is controlled by ‘texture_threshold’, ‘prefilter_size’, and ‘prefilter_cap’ parameters.
2. Filter phase: In this phase, stereoBM algorithm tries to determine all correspondence matches based on the pair images using block matching method.
   This phase is controlled by ‘num_disparities’, ‘min_disparities’, and ‘block_size’ parameters.
3. Postfilter phase: this phase controls all correspondence matches by eliminating false matches.
   This phase is controlled by ‘uniqueness_ration’, ‘speckle_size’, and ‘speckle_range’ parameters.

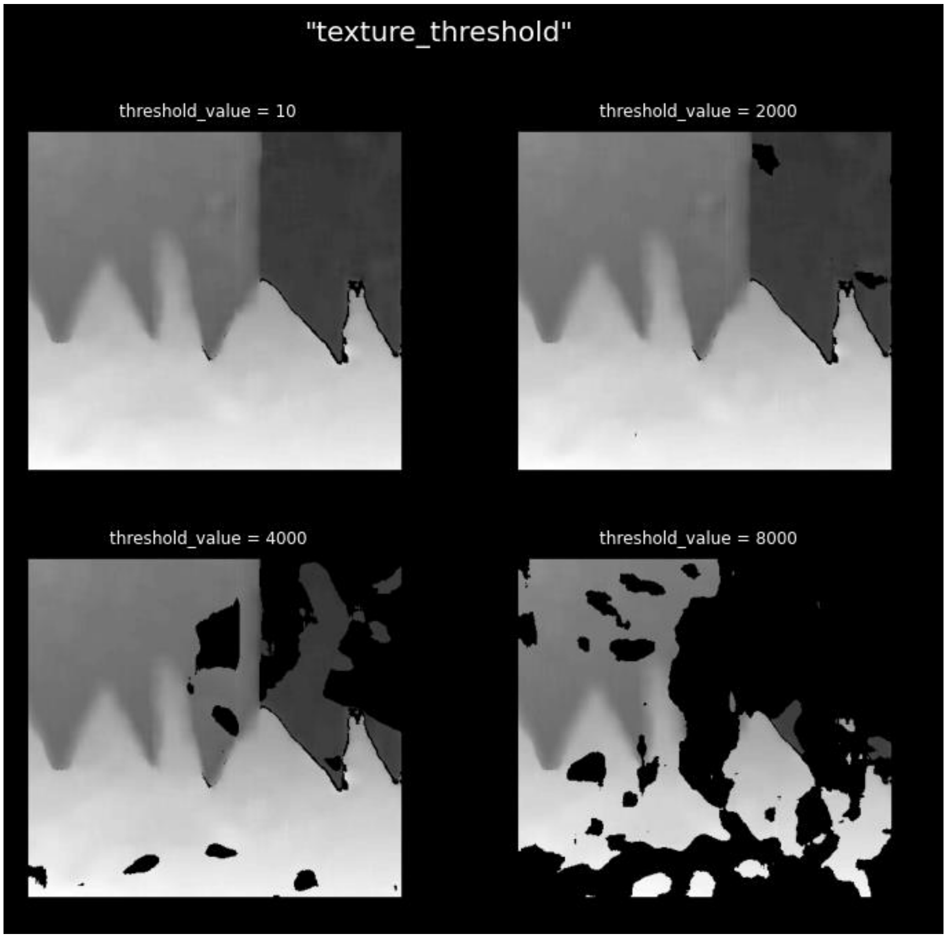
It should be noted that there is no single best values for all image pairs but best parameters for each pair of images should be obtained with trial and error.

For the first six questions, ‘**sawtooth’** dataset is considered in order to evaluate the influence of mentioned parameters above:

A sawtooth dataset sample:

1.

Texture_threshold: filters out areas that don’t have enough texture for reliable mathing.



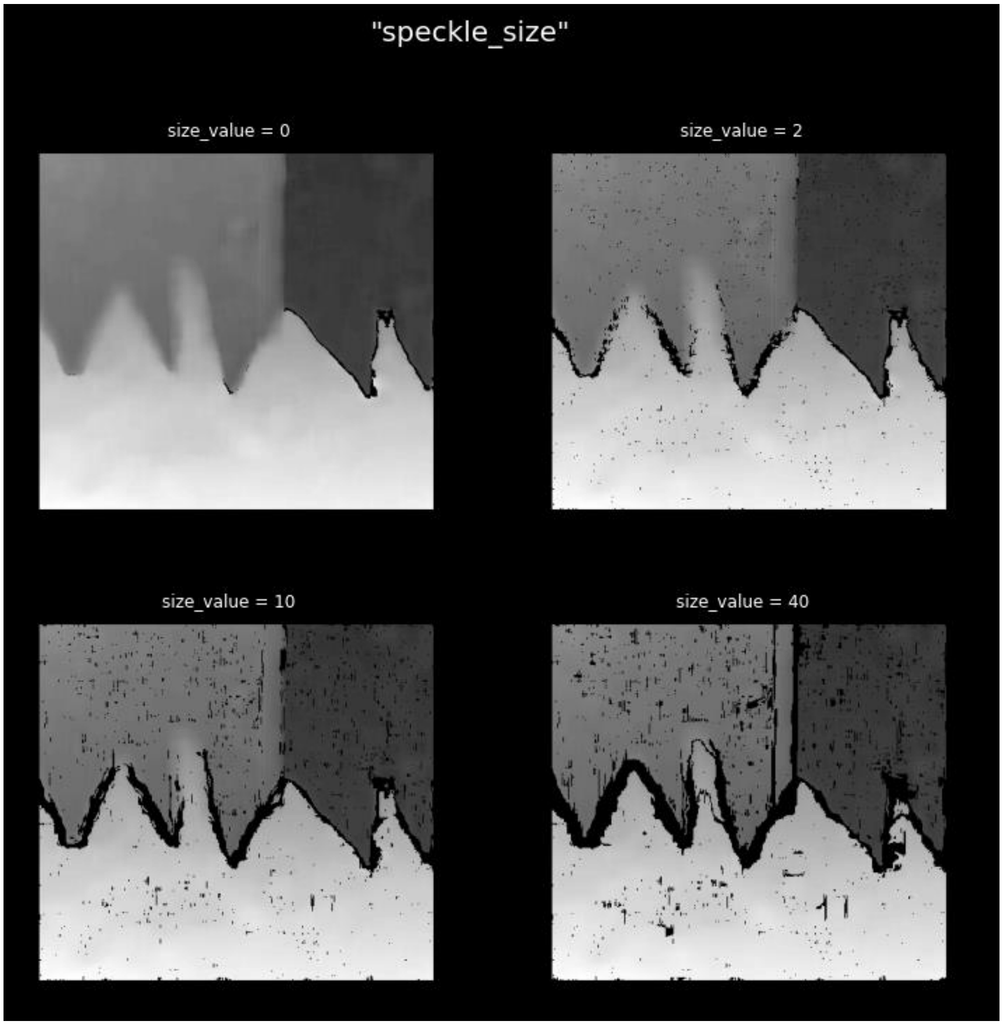
As we can see in the figure above, as the threshold value increases, smoother areas are filtered out (considering sample image above) and considered as invalid disparity value (negative values). As it was expected, more pixels from top-right region is affected since it has more smooth texture compared to two other regions.

2.

Speckle_size and Speckle_range:

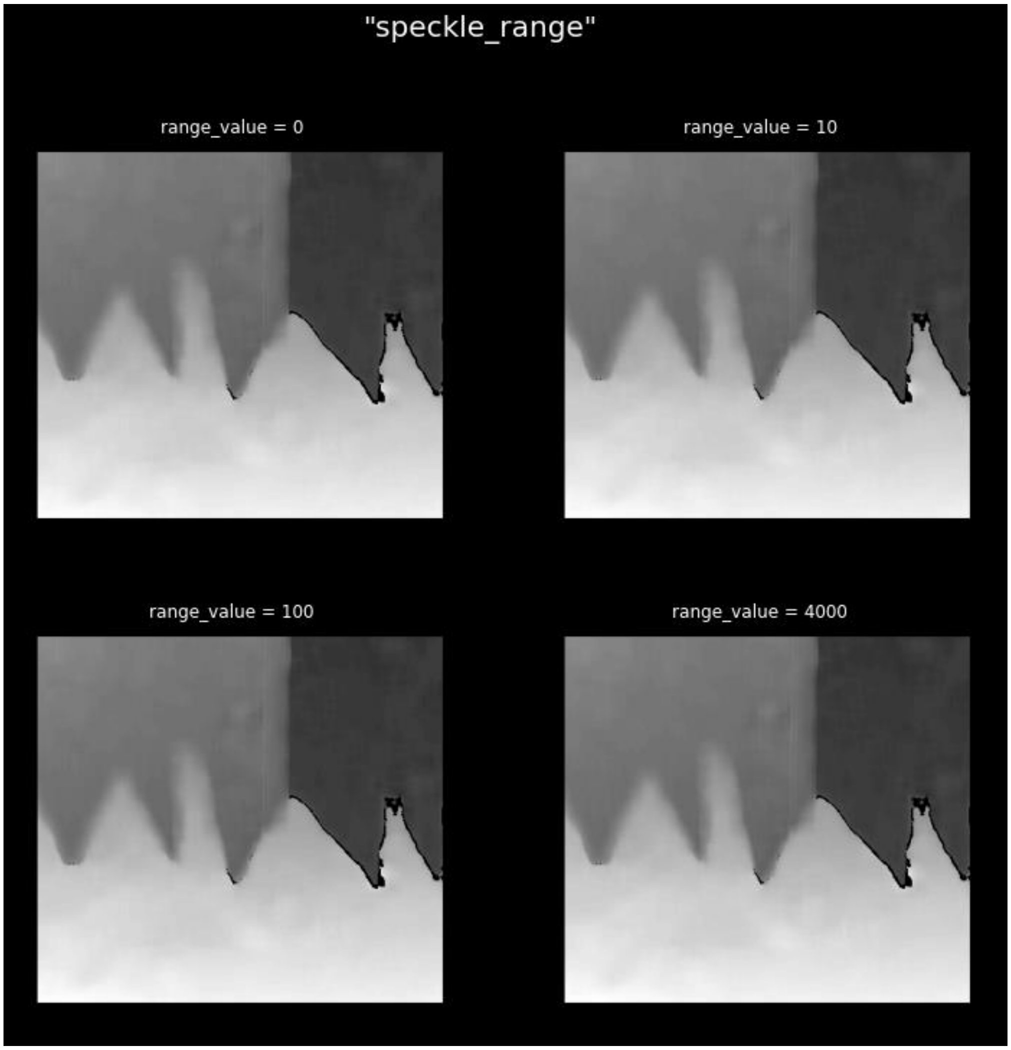
These two parameters replace blobs of similar disparities (the difference of two adjacent values does not exceed speckle_range) whose size is less or equal speckle_size (the number of pixels forming the blob) by the invalid disparity value (negative value).

Speckle_size:



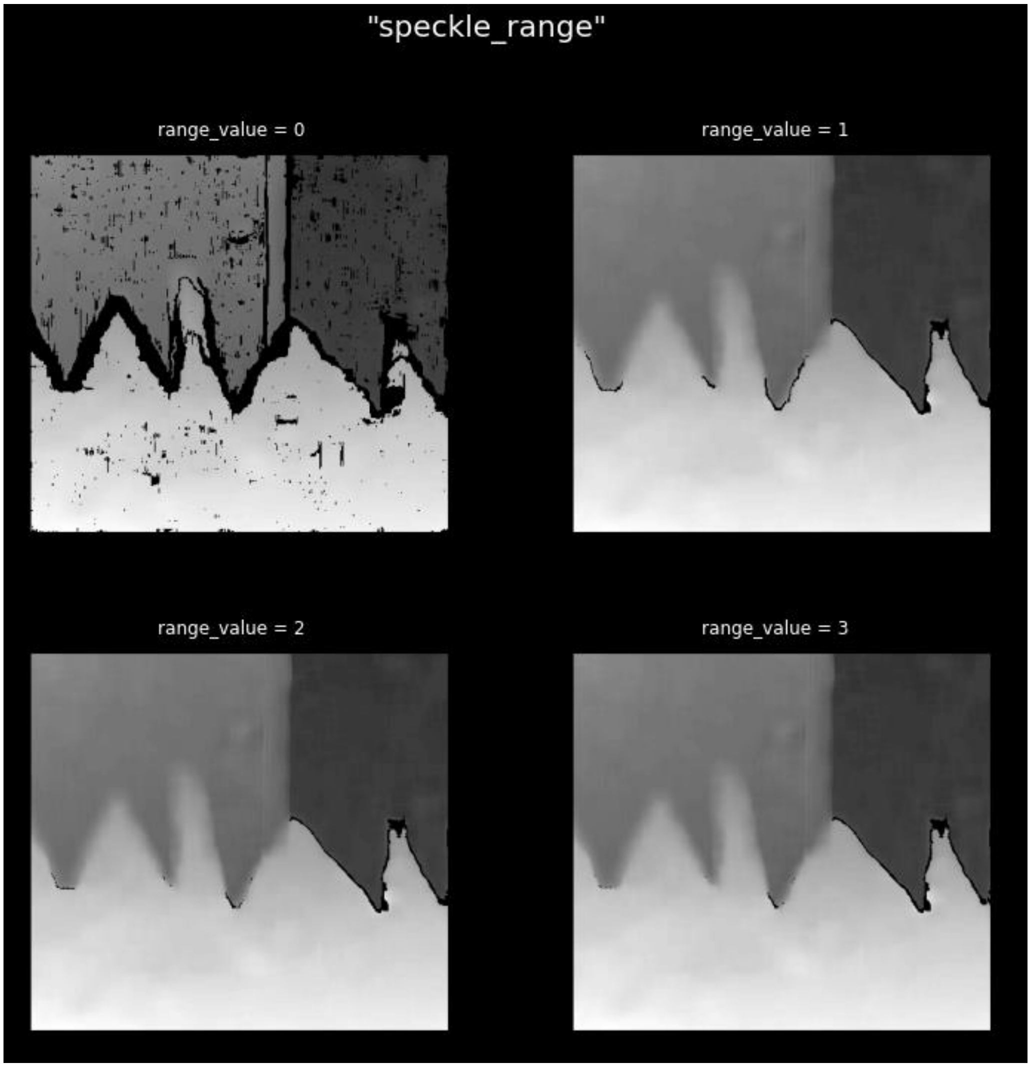
Considering figure above, increasing speckle_size results in increase in number of invalid disparities (negative values) since greater blobs in size would be considered as speckles.

Speckle_range:



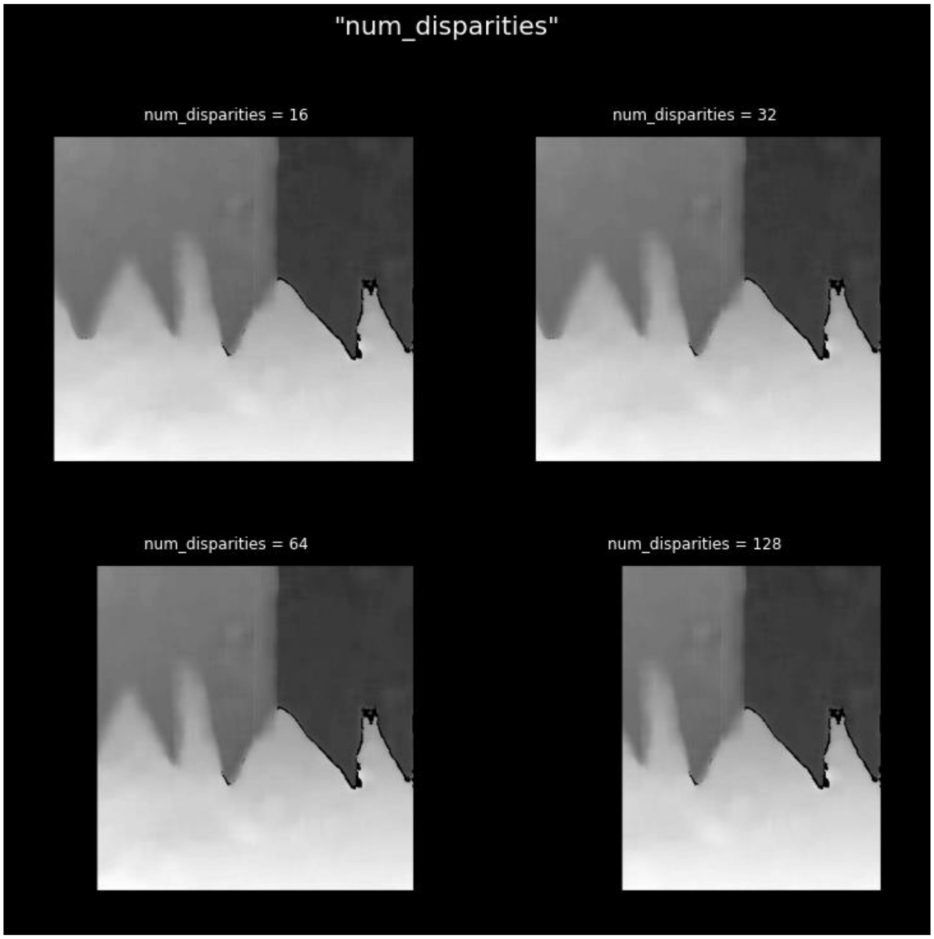
As we can see in the image above, changes in speckle_range doesn’t have any effect on the disparity maps since the window size is set to zero by default.

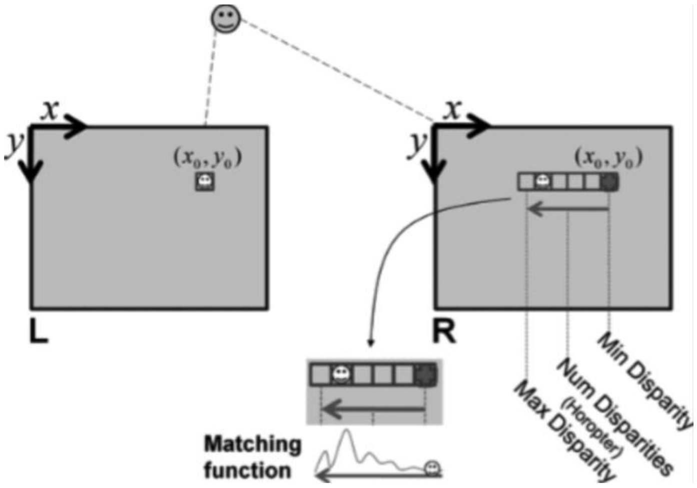
But considering a window size of 50 and changing speckle_range from 0 to 3, we can see that by increasing the speckle_range, invalid disparity values (negative pixels) decreases since the disparity blob sizes would be increased; therefore, higher chances would be for each blob to be greater than the window size (50 in our case) and lastly not to be considered as a speckle.



3.

Num_disparities:



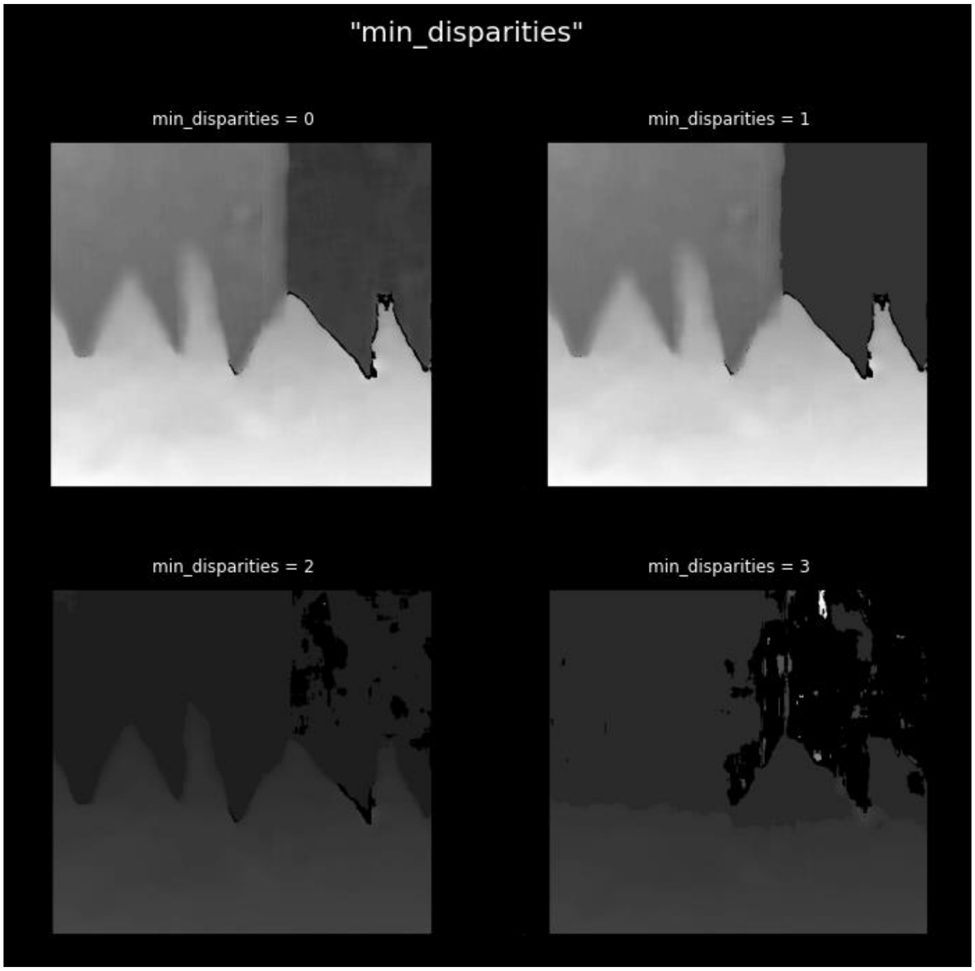


Num_disparity defines number of pixels that the matching window may slide over it, also, this number of pixels are considered only left-hand side of the intended pixel. Therefore, there are no sufficient number of pixels for the first columns of the image to be matched according to the StereoBM algorithm. Therefore, as we can see in the first figure above, as the num_disparity increases, more invalid disparities will be observerd (number of columns are equal to the num_disparity parameter).

4.

Min_disparity: this parameter defines minimum possible disparity value whit the following relation with num_disparity:

(min_disparity + num_disparity = max_disparity)



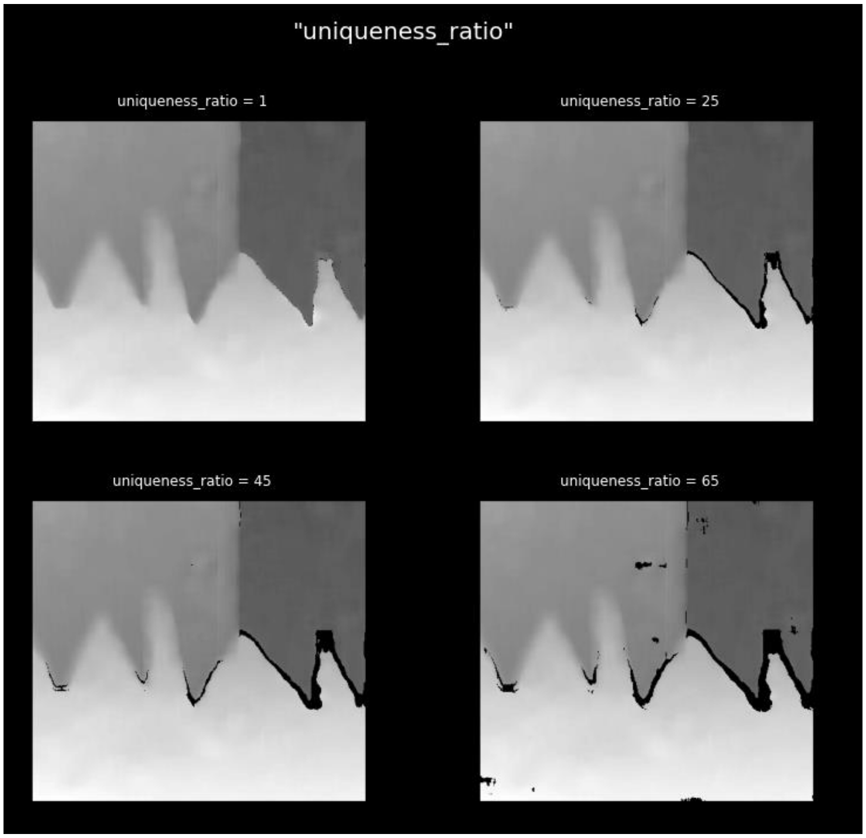
Considering figure above, by increasing min_disparity from to 1, each region of disparity map slightly enhanced in terms of smoothness. But increasing further, resulted in poor disparity maps. Therefore, in our case, it seems that min_disparity of 1 is more sensible.

5.

Uniqueness_ratio:

This parameter filters out pixels with matches which has not enough margin (which is specified with uniqueness_ratio parameter) over the next best match in the corresponding search range. (Margin is percentage of best match cost function and normally should be within 5-15 range)

Therefore, we expect invalid disparity values (dead pixels) to be increased as we increase this parameter.



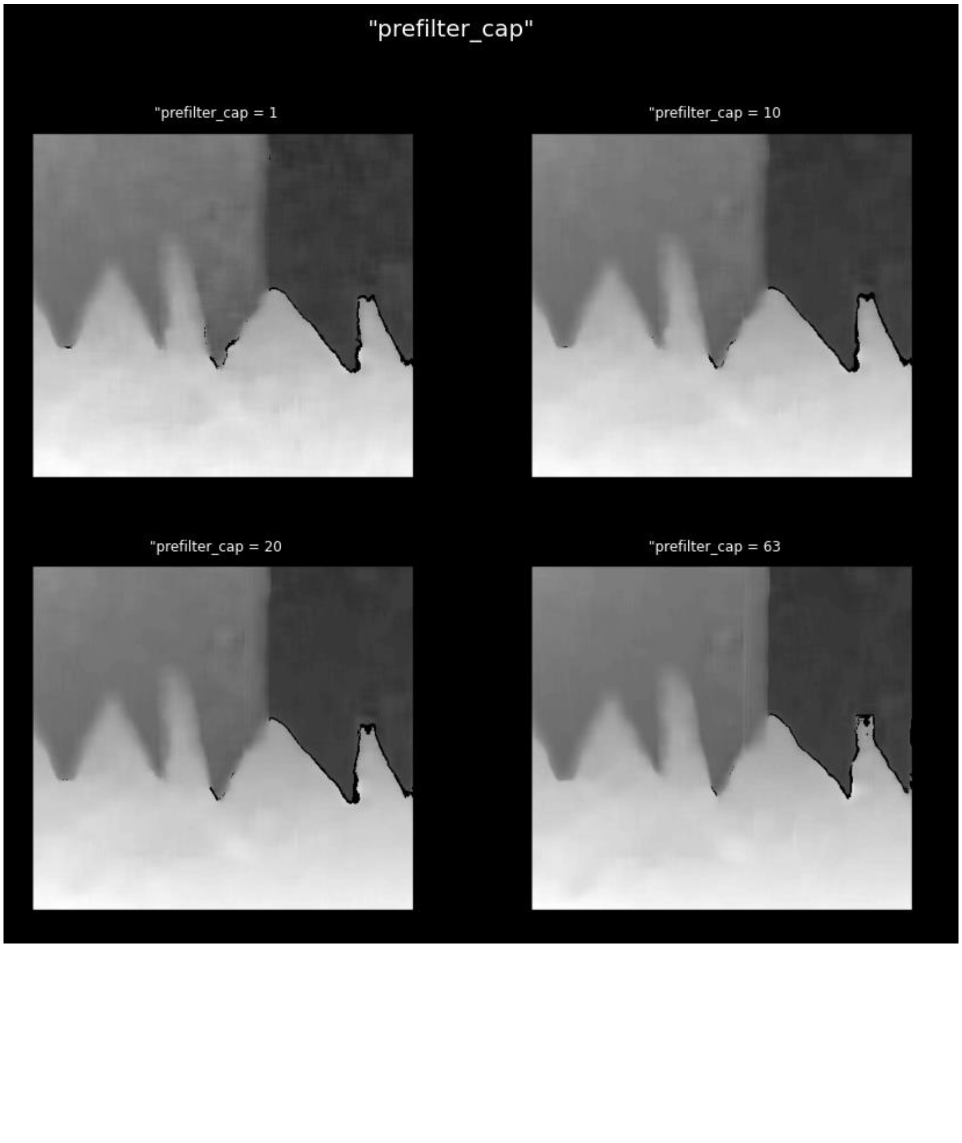
As we expected, dead pixels increased by increasing the uniqueness_ratio parameter.

6.

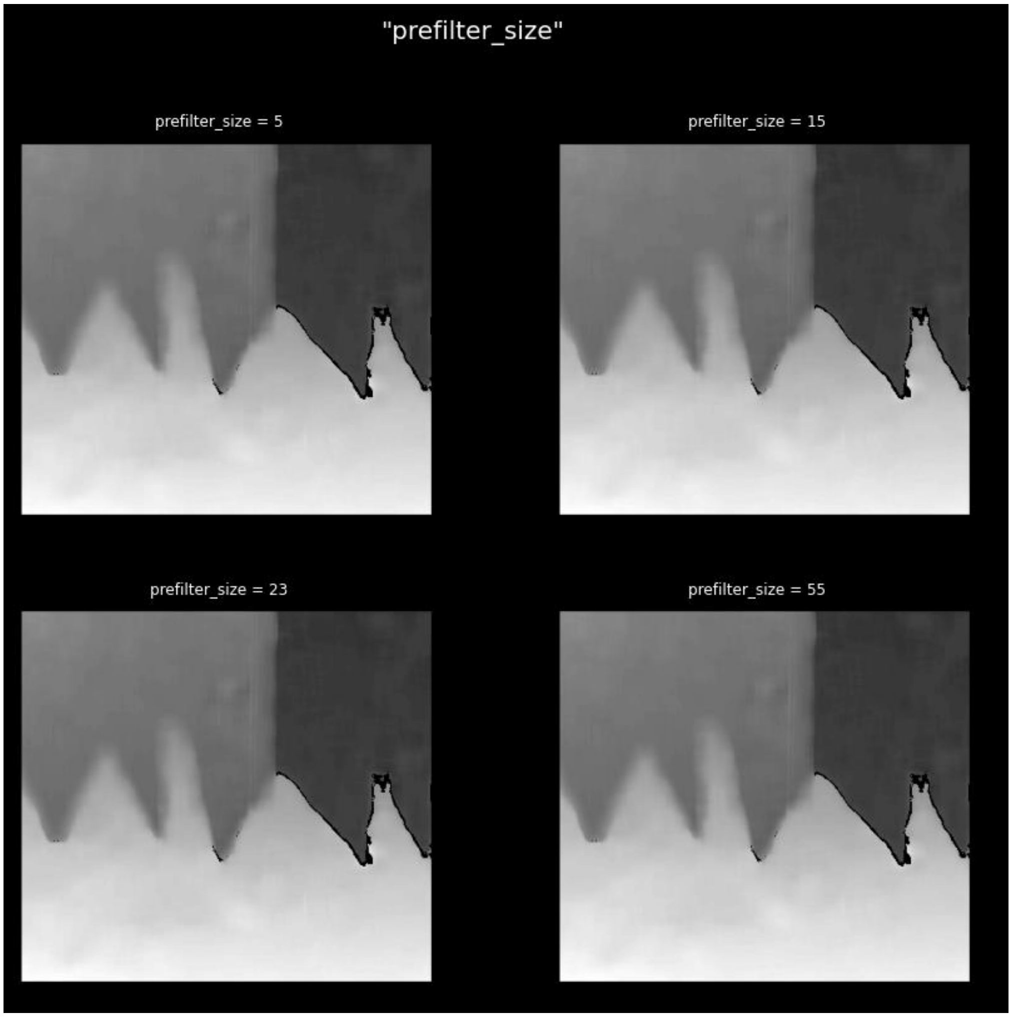
Prefilter_cap and Prefilter_size:

These two parameters cooperate in prefilter phase in order to normalize brightness and enhance texture. The second one (prefilter_size) determines the size of the window by which the images should be filtered according to the value defined by prefilter_cap parameter.

By experiment, it seems that increasing prefilter_cap parameter results in smoother disparity map and less invalid disparity values (dead pixels) as can be seen in figure below:



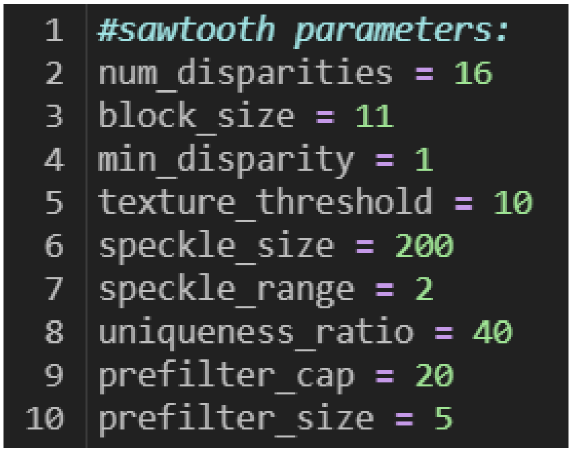
It should be noted that no sensible change could be observed by changing in prefilter_size parameter in none of the datasets:



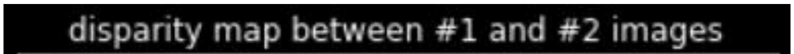
7.

Sawtooth dataset best obtained model:

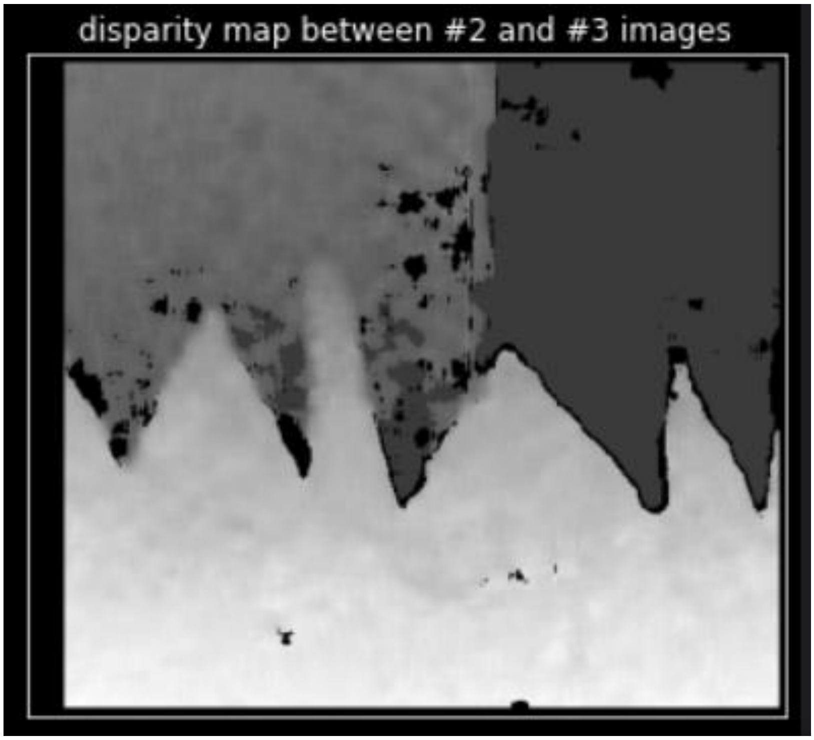
Results are obtained by following parameters:



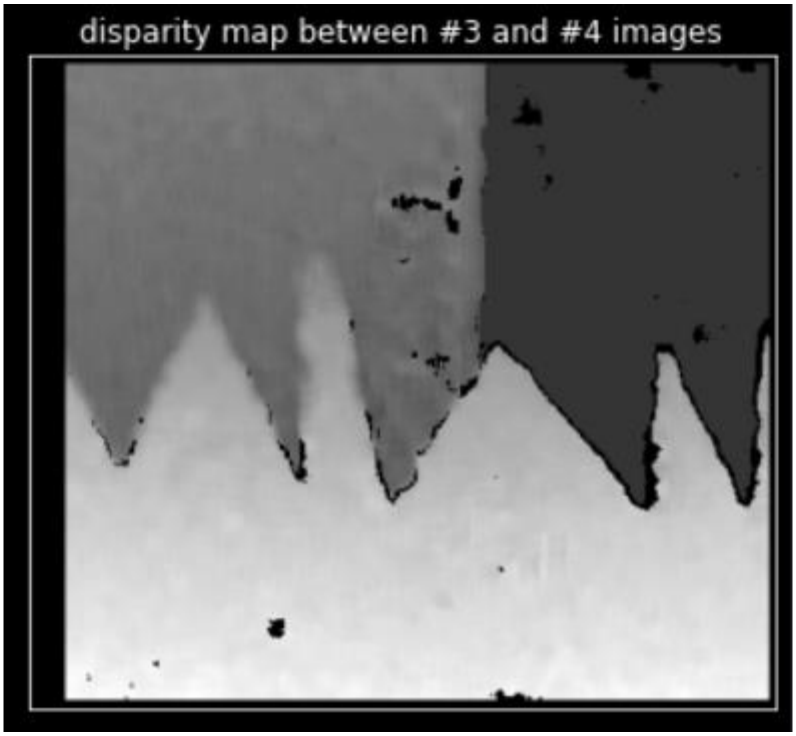
```
#sawtooth parameters:
num_disparities = 16
block_size = 11
min_disparity = 1
texture_threshold = 10
speckle_size = 200
speckle_range = 2
uniqueness_ratio = 40
prefilter_cap = 20
prefilter_size = 5
```



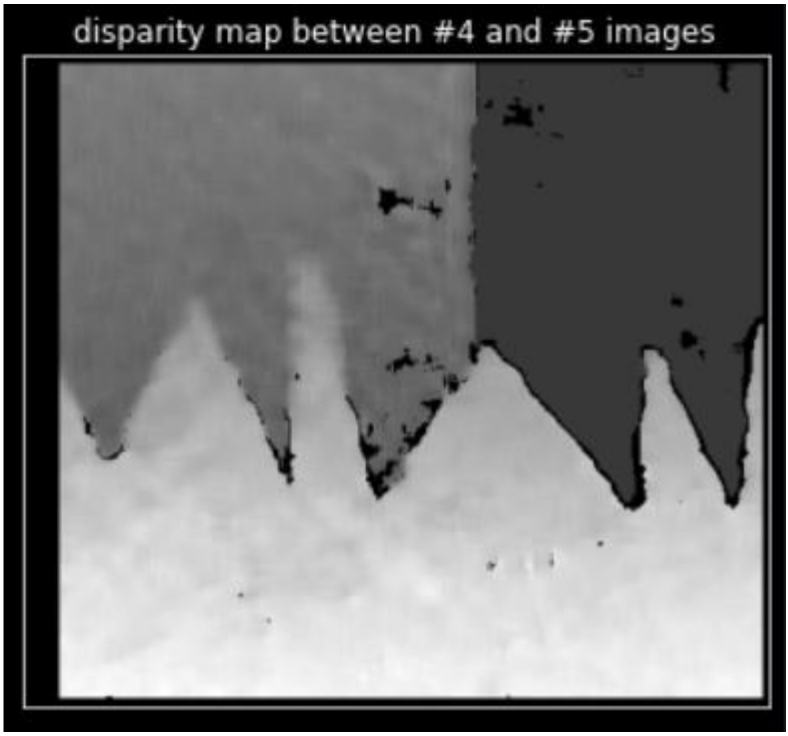
disparity map between #1 and #2 images



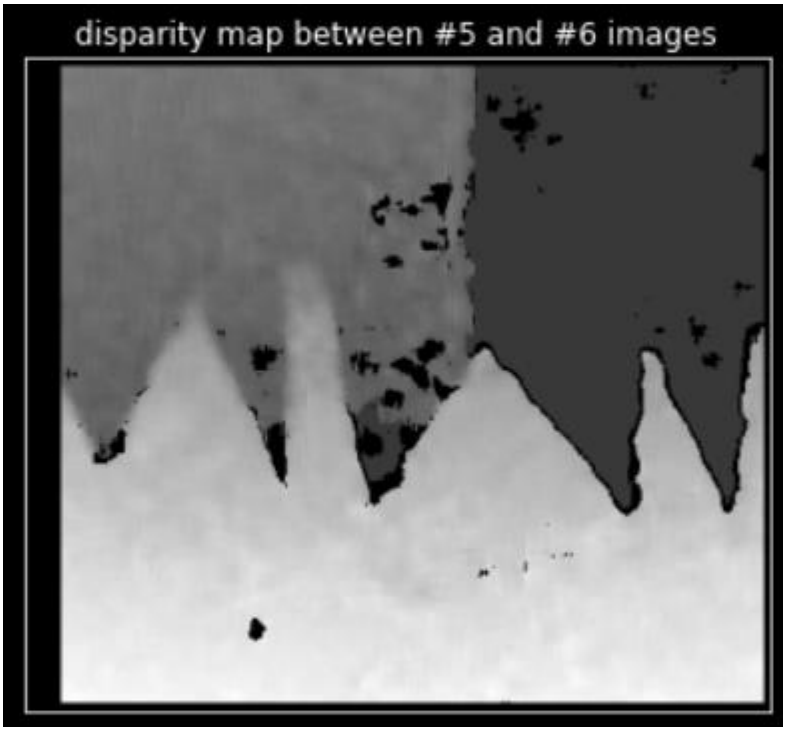
disparity map between #2 and #3 images



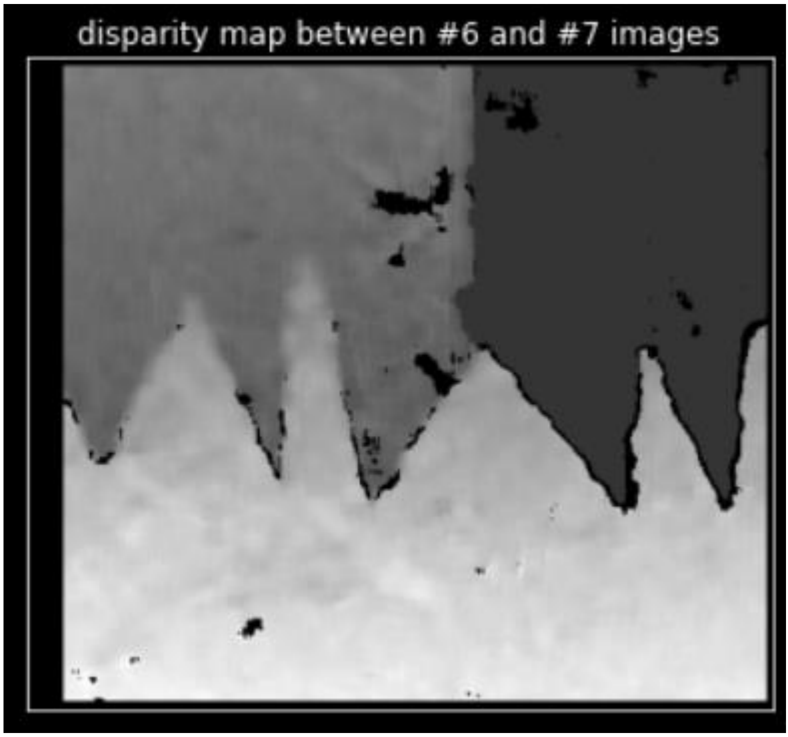
disparity map between #3 and #4 images



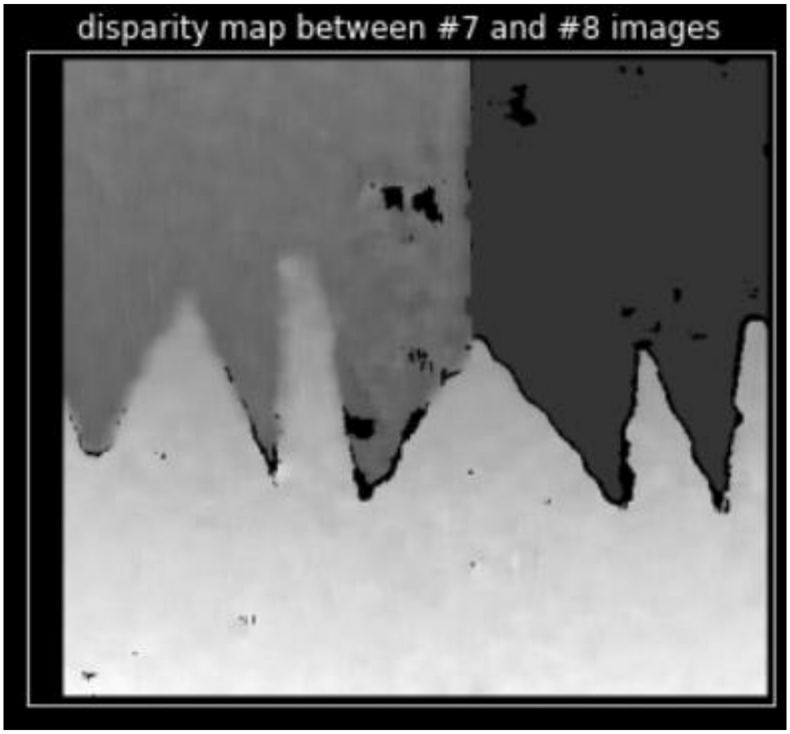
disparity map between #4 and #5 images

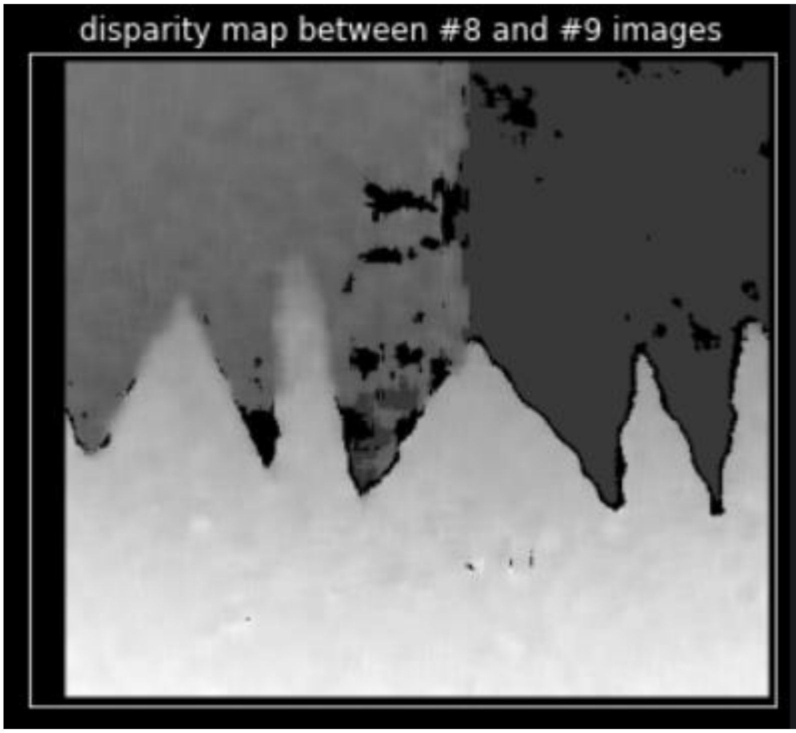


disparity map between #5 and #6 images



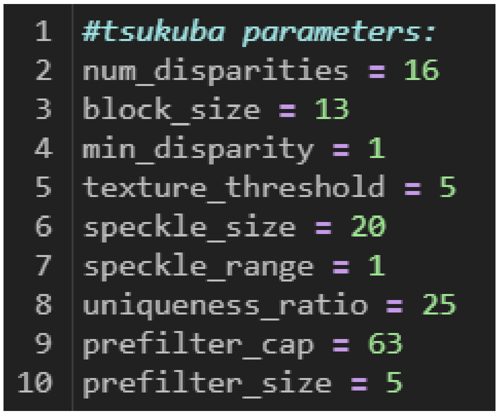
disparity map between #6 and #7 images



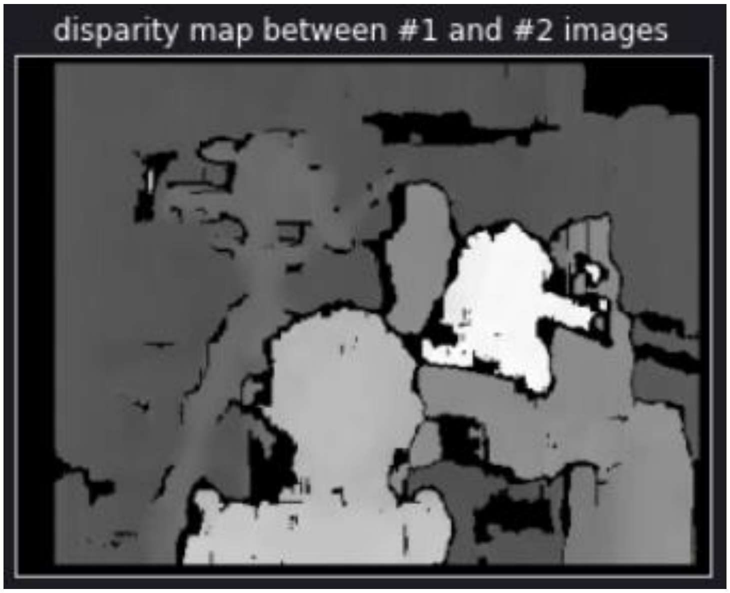


Tsukuba dataset best obtained model:

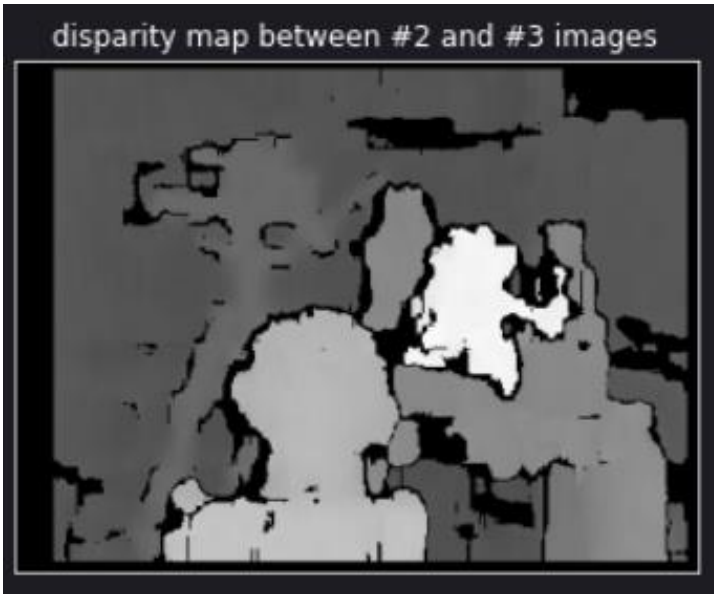
Results are obtained by following parameters:



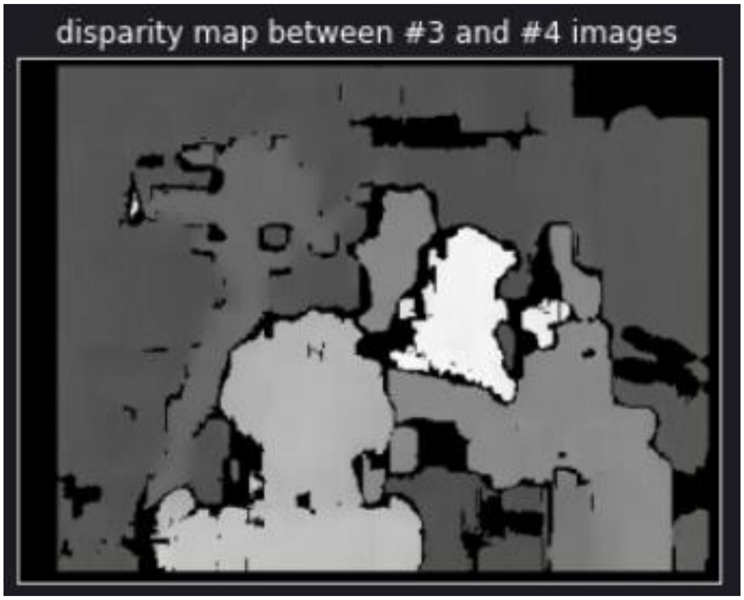
```
#tsukuba parameters:
num_disparities = 16
block_size = 13
min_disparity = 1
texture_threshold = 5
speckle_size = 20
speckle_range = 1
uniqueness_ratio = 25
prefilter_cap = 63
prefilter_size = 5
```



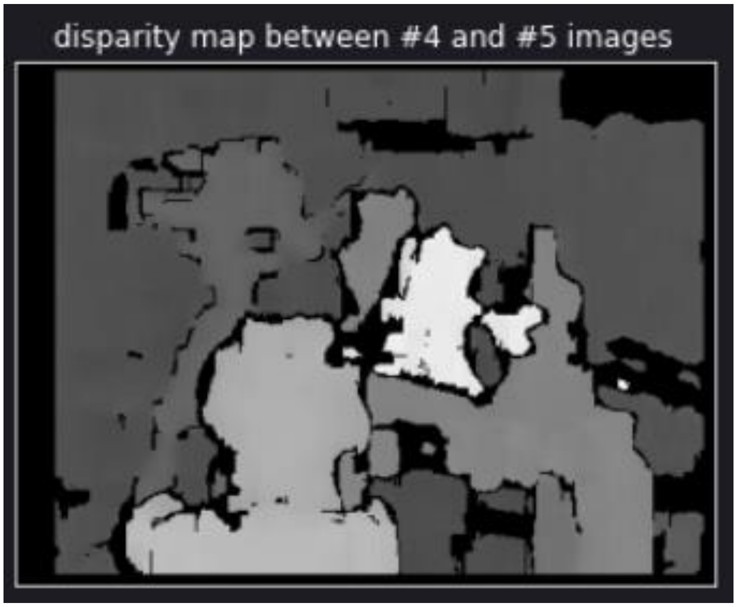
disparity map between #1 and #2 images



disparity map between #2 and #3 images



disparity map between #3 and #4 images



disparity map between #4 and #5 images