

## README

**Python program for a stereo analysis system involving feature-based, region-based and multi-resolution matching.**

Stereo Matching is done using two methods. Namely, Region based, and feature based. The matching scores available are SSD, SAD and NCC. Either can be selected in the terminal.

### **Steps:**

- At a given level, stereo matching is done, and disparity calculated.
- Validity check is then performed, wherein if the left-to-right match does not correspond to right-to-left match, a zero is placed at that location in the disparity.
- Averaging is performed in the neighborhood to fill these zeroes.
- Disparity is propagated to the next lower (finer) level. This is done by duplicating disparity from 1-pixel to the corresponding 4-pixels in the lower level.
- Using this disparity as starting point of the search, stereo matching is performed, and disparity is updated.
- Harris corners detection is used for feature detection and Harris corner response measure is used as the descriptor value for matching.

In both the files, Images can be changed at:

```
# Change the images here to test
left = cv2.imread('left2.png')
right = cv2.imread('right2.png')
```

Number of levels of the pyramid can be set at: (level 1 is the original image)

```
# Set the desired number of levels for multi-resolution here (level = 1 is the original image)
levels = 3
```

Template size and window can be set at:

```
# Set the template size here
templateSize = 7

# Set the matching window here
window = 100
```

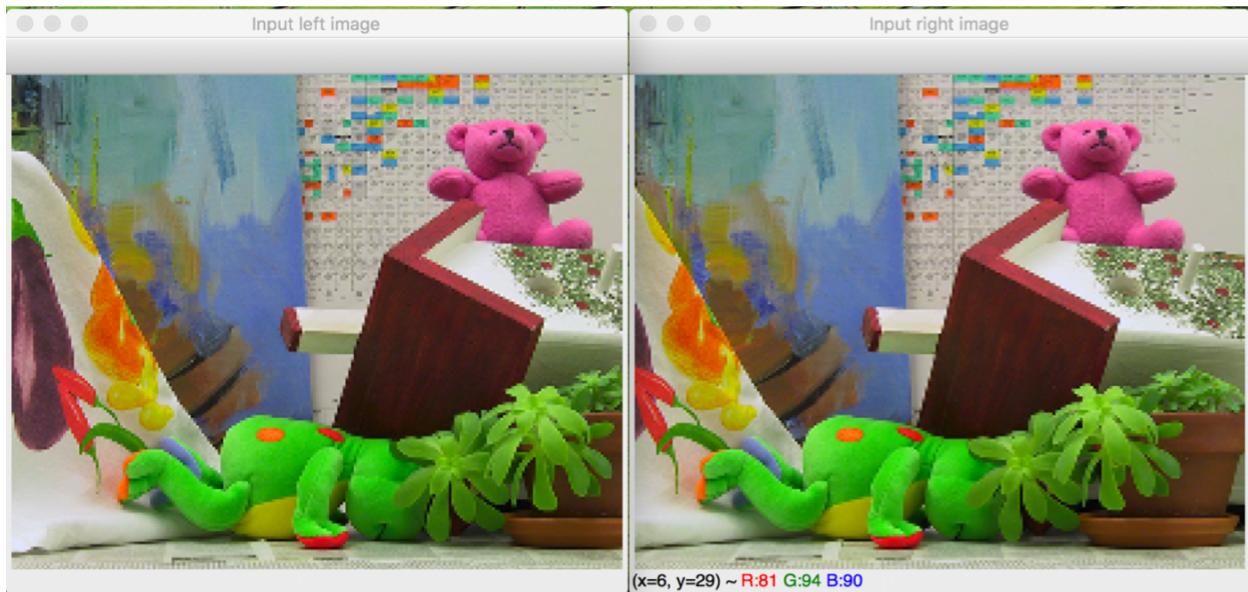
Matching score can be selected in the terminal:

```
Select a matching score: 1.SSD    2.SAD    3.NCC
```

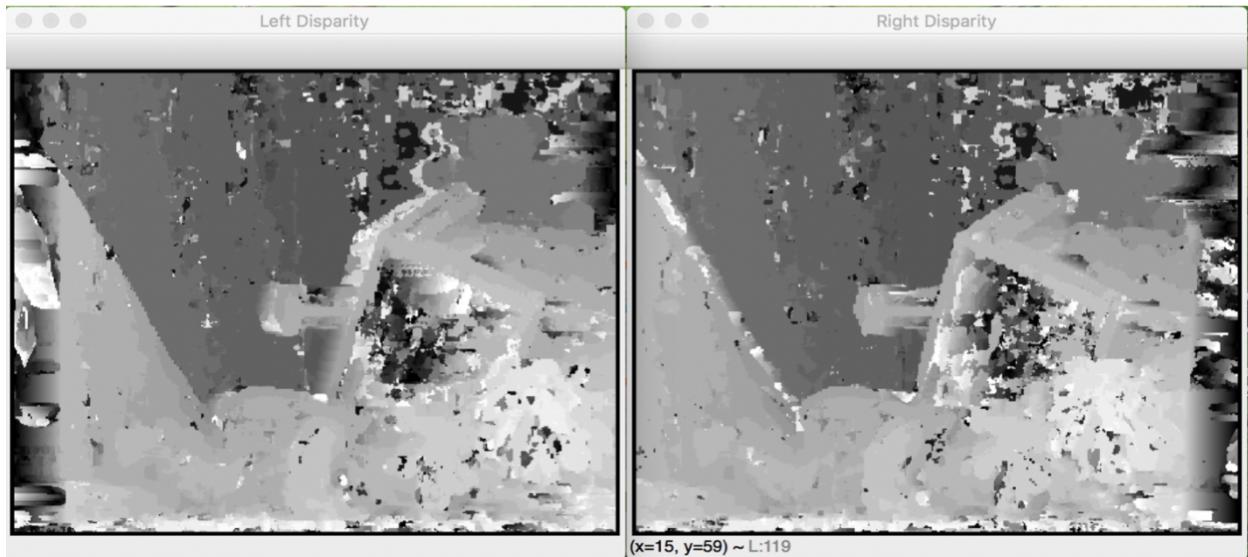
## Some Examples:

### 1. Region Based:

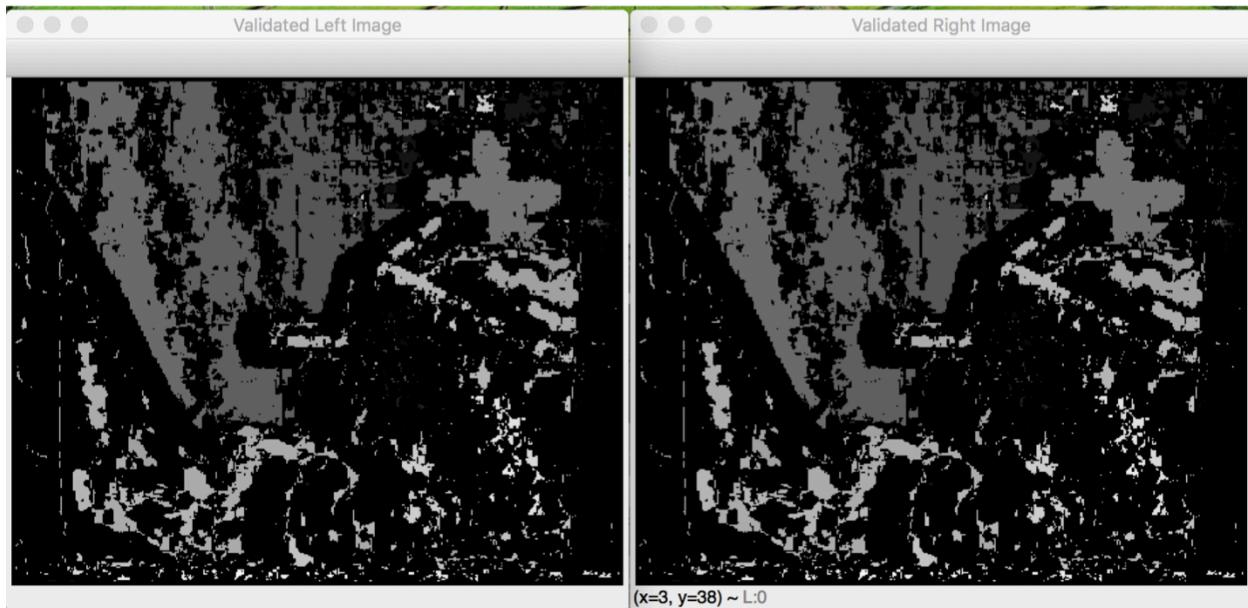
Eg 1: Levels = 2 Template size = 7 Window = 100 Matching score = SSD Input images:



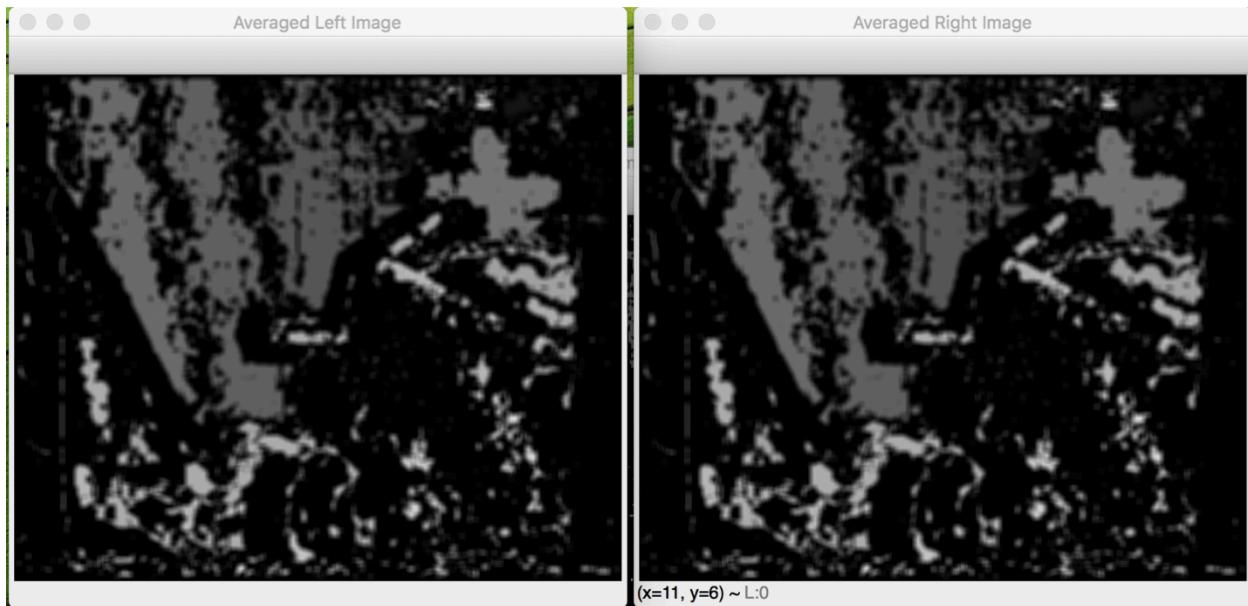
### Disparities:



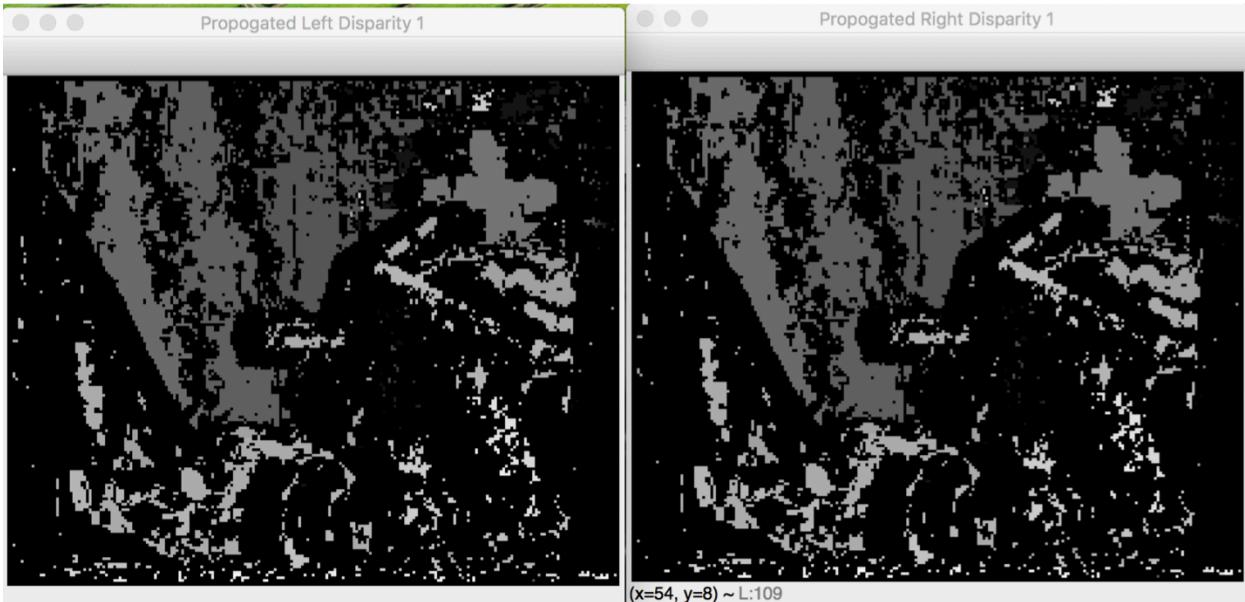
**Validation:**



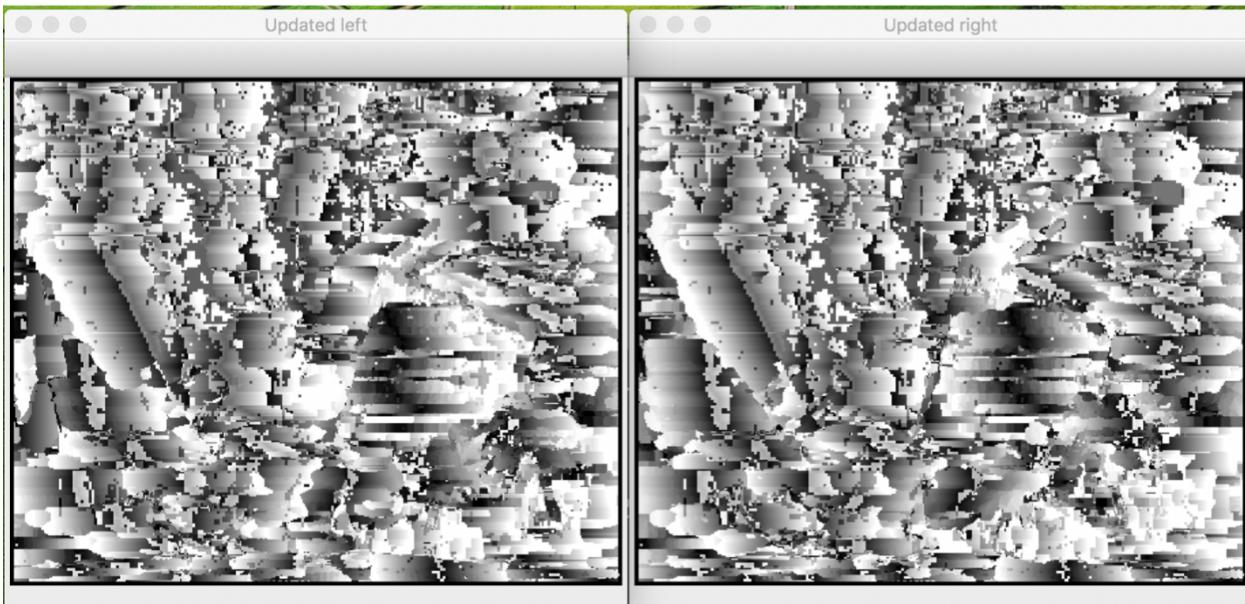
**Averaging:**



Disparity Propagated to lower level:

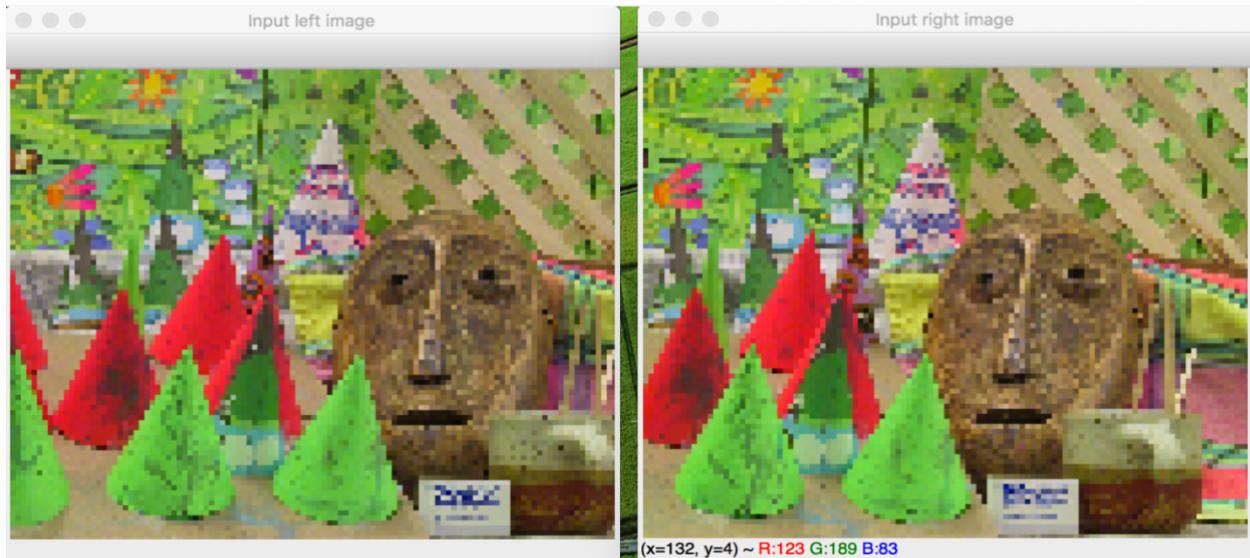


Updated disparity:

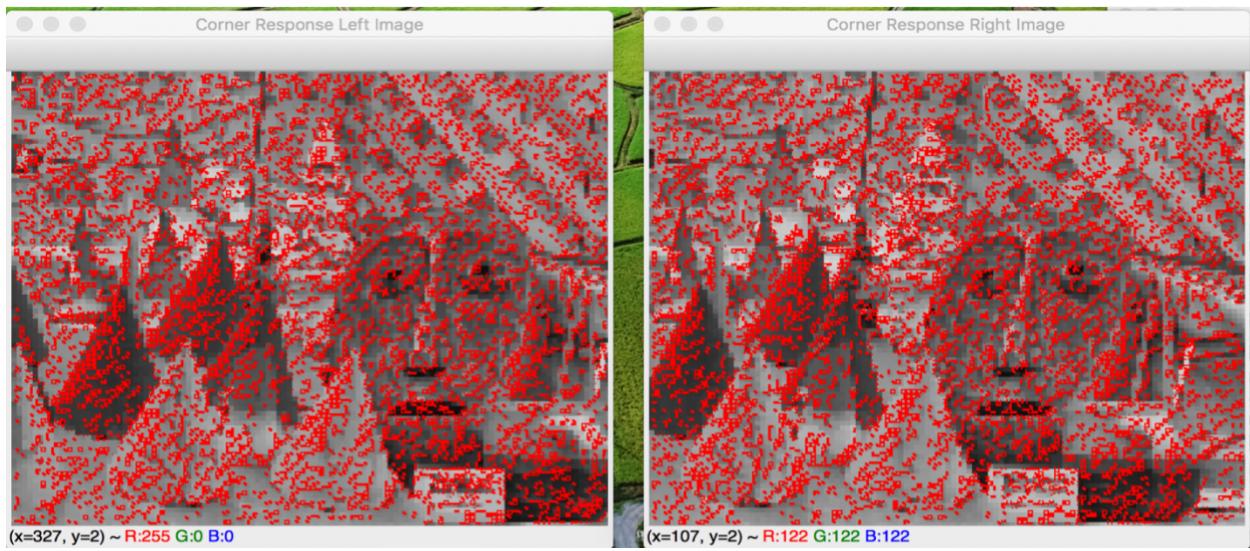


## 2. Feature Based:

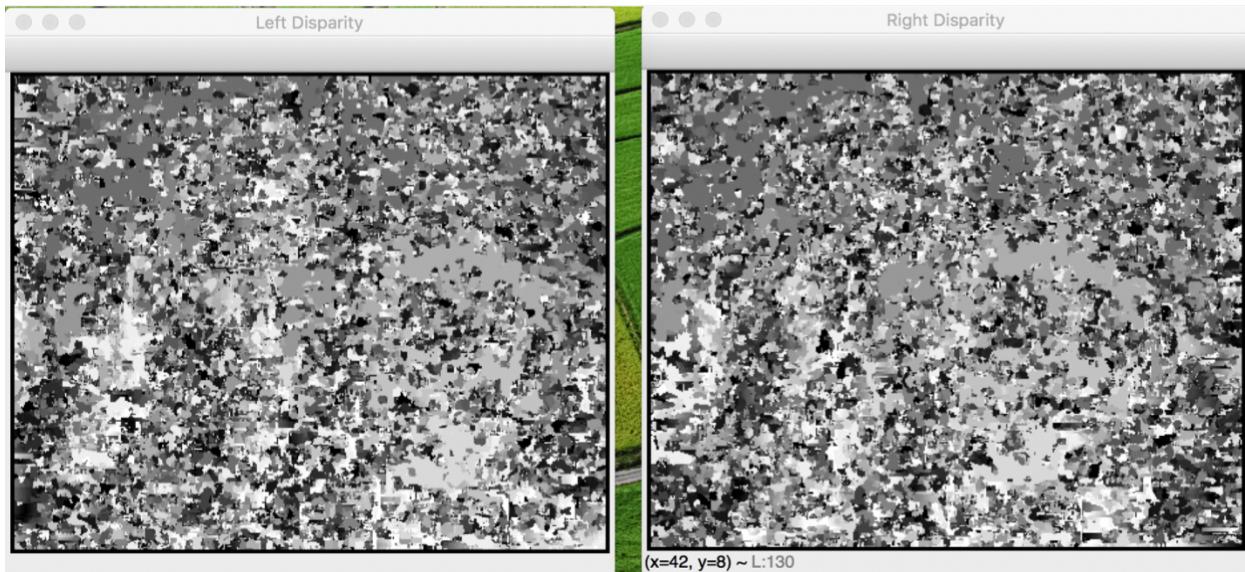
Eg 2: Levels = 3 Template size = 7 Window = 100 Matching score = NCC Input images:



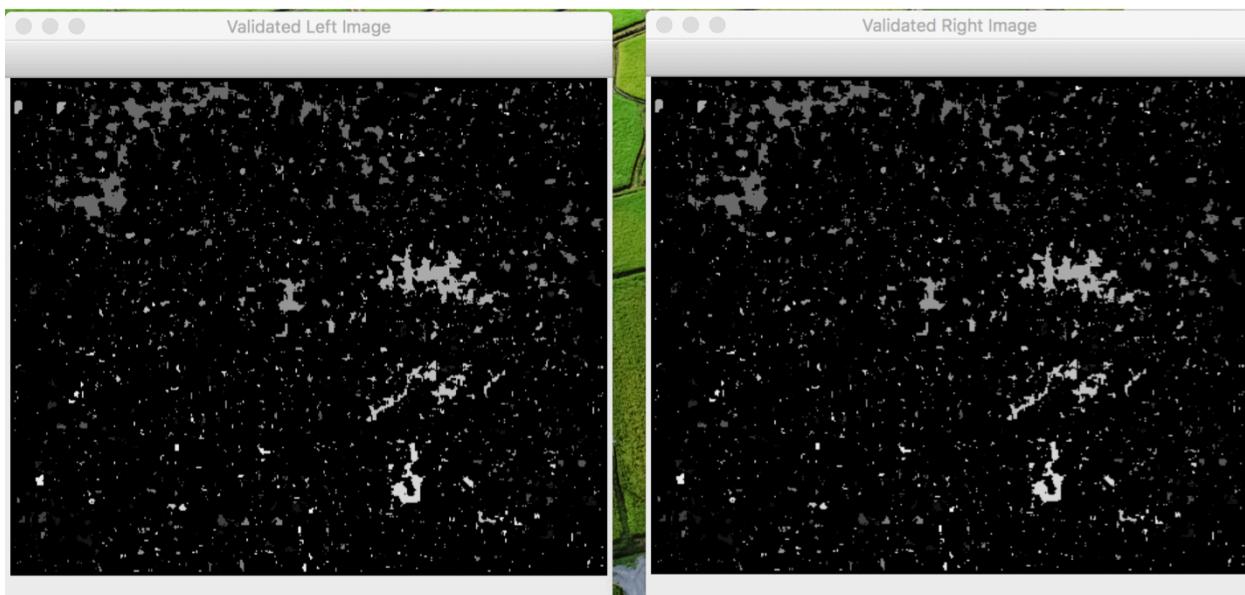
Images after feature detection:



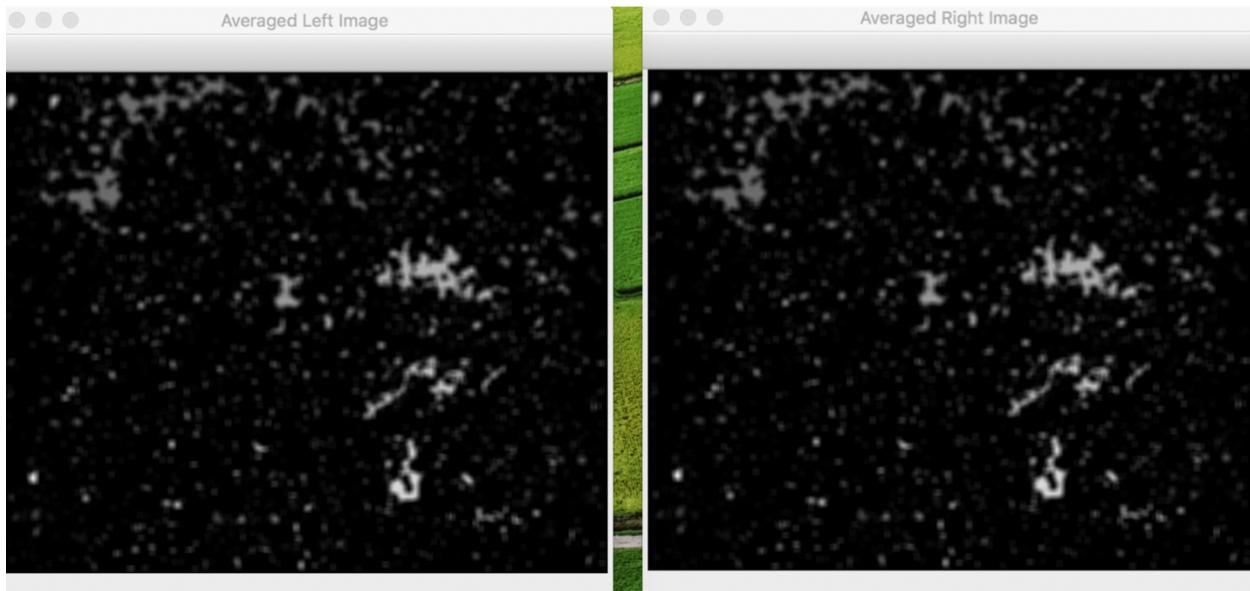
Disparities:



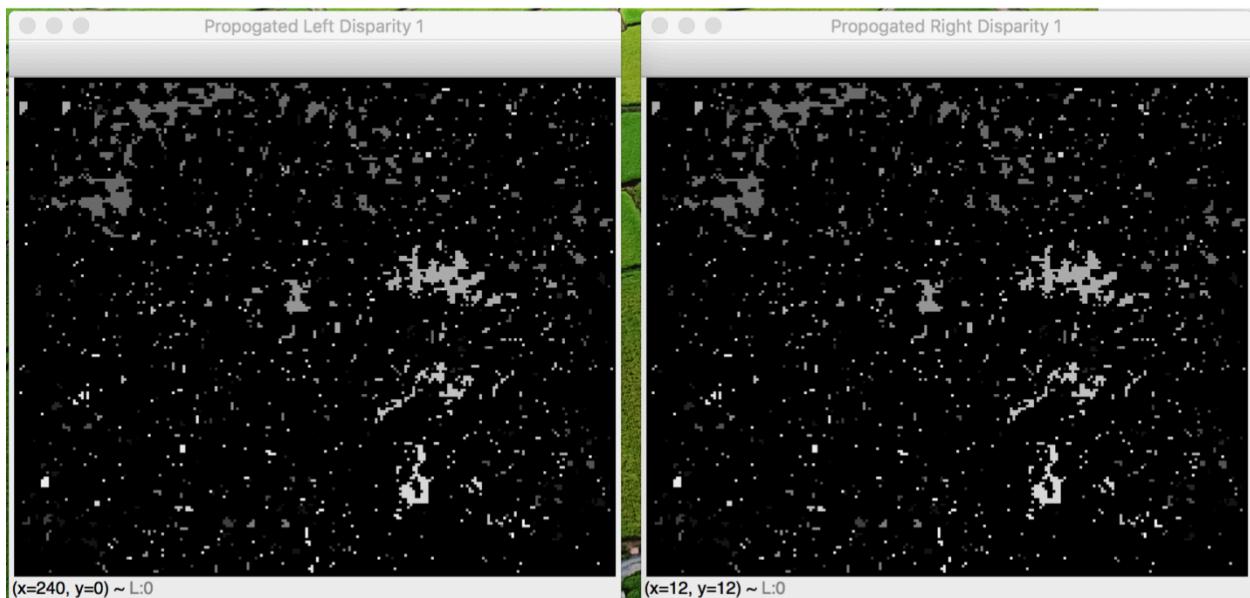
Validity check:

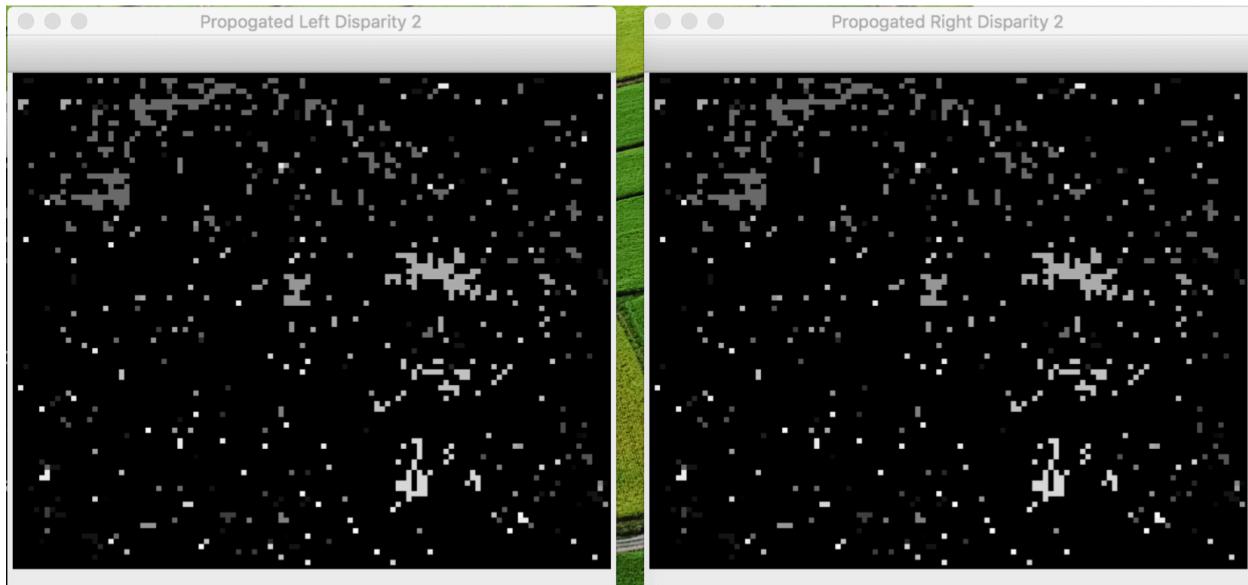


Averaging:



Disparities propagated to lower levels:





Updated disparities:

