

Figure 1: Four images and corresponding groundtruth used in [Middlebury Stereo Evaluation](#).

## 1 Part 1

For this part I used the four image set (Fig. 1) on the web site of [\[Scharstein and Szeliski, 2002\]](#). The implementation uses the similarity measure (1) in [\[Fua, 1993\]](#).

### 1.1 Results

For each image of the test set I used the respective disparity range and scale (Fig 2).

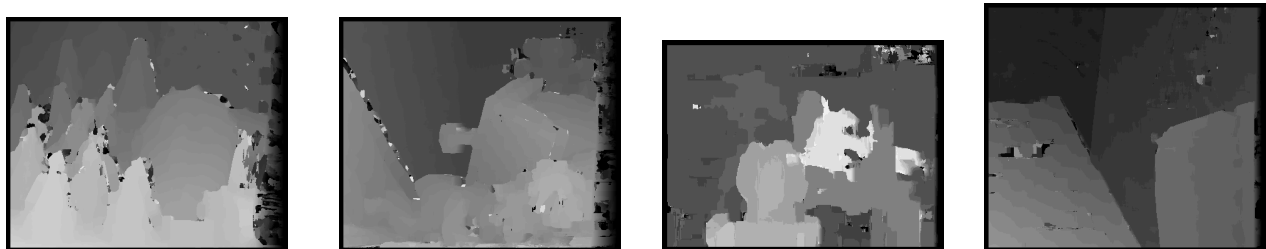


Figure 2: Preliminary results of the implemented method.

## 2 Part 2

For scoring the output I compute the fraction of correct pixels within one disparity value as instructed. However, this score is determined over two different regions for each image. (i) Over all the defined groundtruth data and (ii) Non-occluded area. Doing so allows me to compare my output with state of the art methods listed at [Middlebury Stereo Evaluation](#).

## 2.1 Selecting Window Size

To select the best window size for the similarity test I tried all the odd window sizes within range  $[1, 55]$  and recorded all the scores. This parameter tuning is done on the simple approach of Part 1 as instructed.

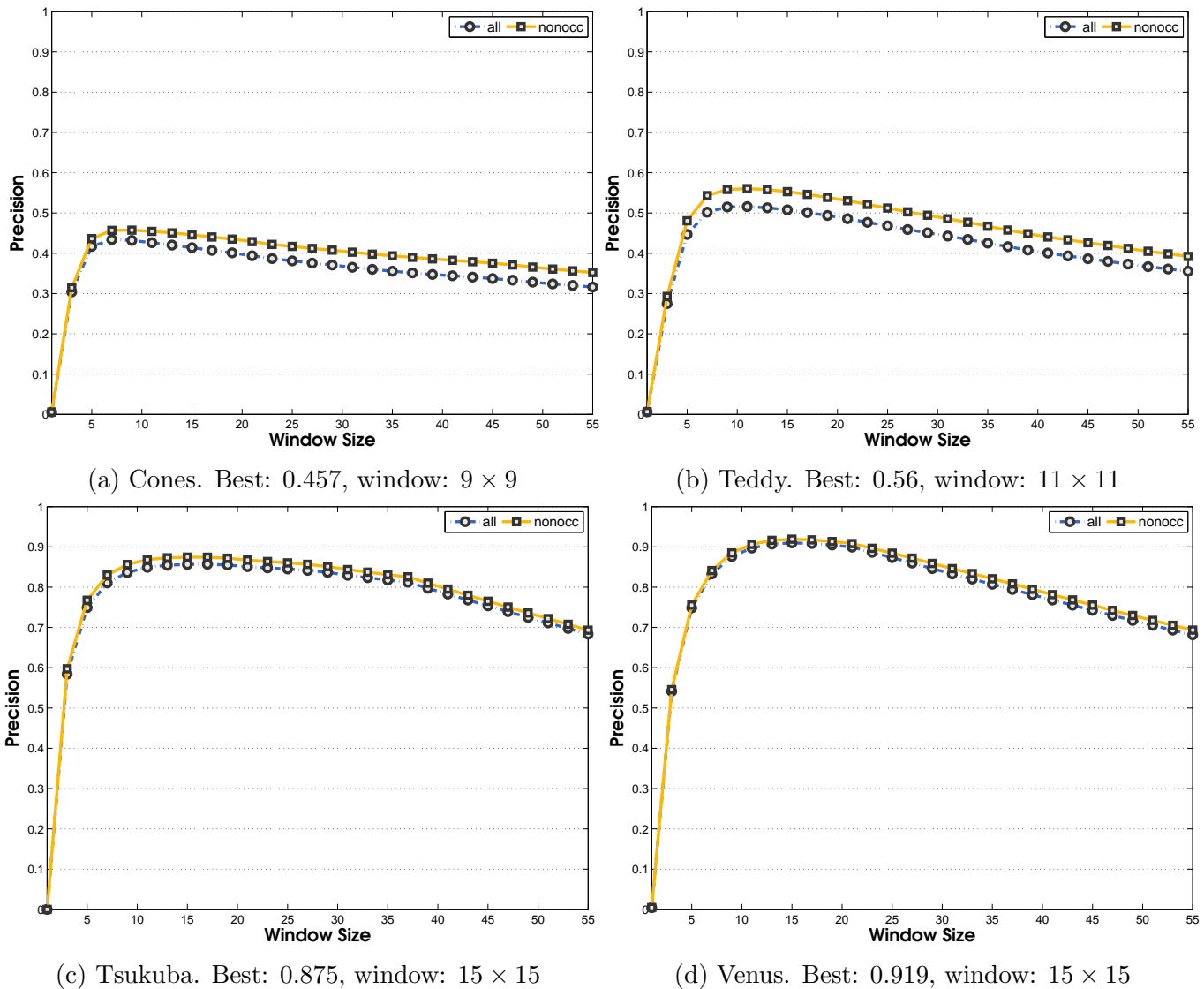


Figure 3: Precision of the implementation as a function of window size.

As you can see in Fig. 3, setting the window size to 15 is a reasonable decision as it approximately maximizes the average precision. (For more images see Fig. 7)

Given that our implementation has a significantly worse performance on 3a and 3b you may wonder what's the explanation for this. To answer this question I cut the regions of the images which were incorrectly labeled. The results are shown in figure 4.

Based on the images of Fig. 4 I can think of two possible reoccurring patterns that appear in the error images.



Figure 4: Left images are the correctly labeled portions and right images are incorrectly labeled portions of the image

- **Non-Distinctive Pattern.** Most of the error regions in Teddy and Cones have a reoccurring pattern that magnifies the weakness of our approach for similarity testing. In such circumstances, we also need to rely on the geometric displacements in neighboring pixels.
- **Boundary Regions.** This type of errors usually appear when we're around object boundaries

and thus unable to correctly identify the matching region.

Furthermore, I analyzed the distance of first best match to second best match to examine the helpfulness of our similarity measure. The following graphs show the ratio  $\frac{1-NCC_{best}}{1-NCC_{secondbest}}$  for when we have found the correct match and for when we are wrong.

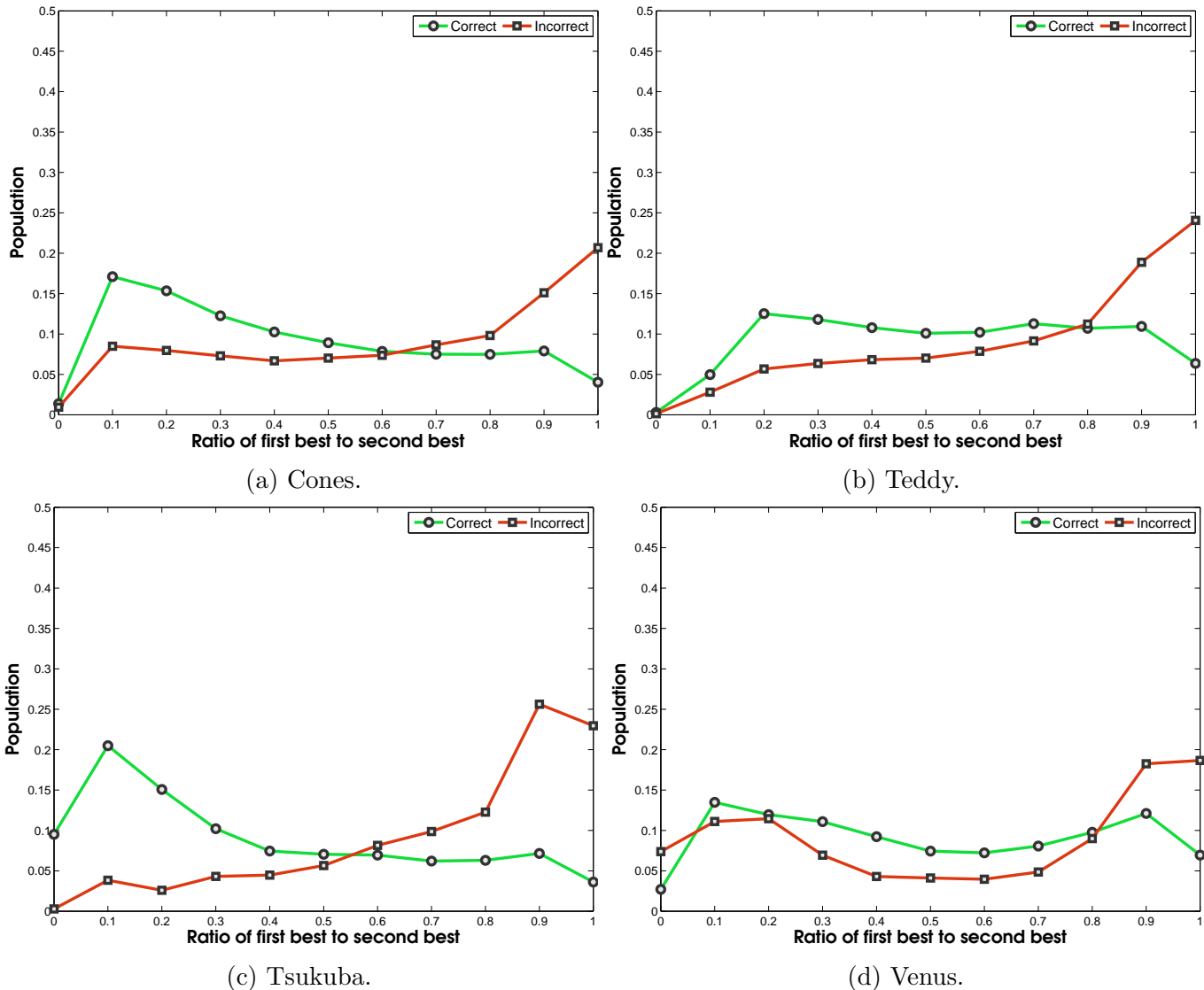


Figure 5: The ratio of distances of first match and second match from the best possible match ( $NCC = 1$ ).

Based on the Fig. 6, we can perform validation check by marking estimations with ratios over 0.8 as bad estimations (similar to double peek test discussed in the class). It must be noted that it is equally important to correctly fill the marked regions, or the final outcome could be even worse as we're also marking a small portion of the correctly estimated region as wrong.

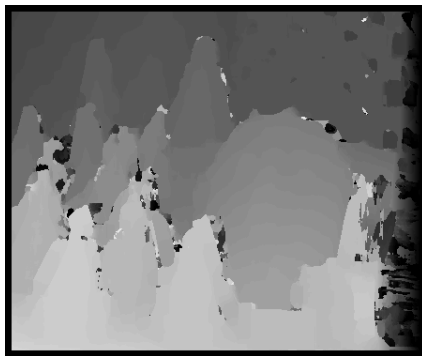
### 3 Part 3

For this part, I implemented the validity check described before and tried two methods to fill the marked region.

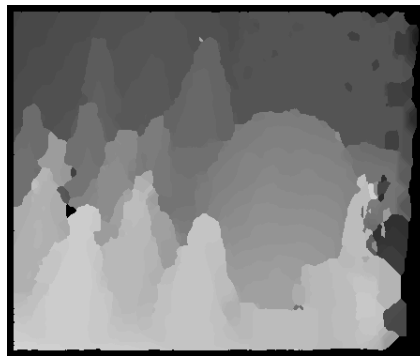
- **Nearest Valid.** In this approach each marked region is filled with the value of nearest valid region.
- **Linear Row Interpolation.** After identifying the maximal length marked region in each row of the image, that area is linearly interpolated.

Table 1: Results for different filling methods. The window size used for evaluation is  $15 \times 15$  and the error in non-occluding region is reported.

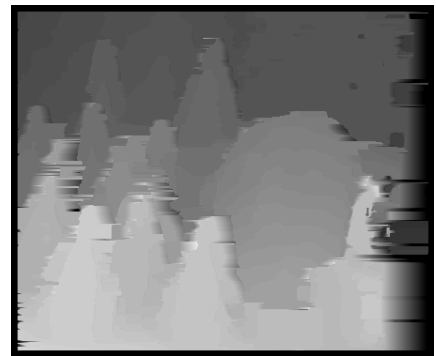
Filling Method	Image	Without	With	Change
Nearest Valid	Cones	0.446	<b>0.453</b>	+0.007
	Teddy	0.553	<b>0.56</b>	+0.007
	Tsukuba	0.875	<b>0.895</b>	+0.02
	Venus	0.919	<b>0.924</b>	+0.005
Linear Row Interpolation	Cones	0.446	<b>0.477</b>	+0.031
	Teddy	0.553	<b>0.577</b>	+0.024
	Tsukuba	0.875	<b>0.891</b>	+0.016
	Venus	0.919	<b>0.923</b>	+0.004



(a) Cones without validity check.



(b) Cones with nearest valid.



(c) Cones with lin. interpolation.

Figure 6: Output after filling the identified invalid region with different methods.

Interestingly the linearly interpolated image, though less visually appealing, gives a better estimation compared to the smooth looking output of the nearest valid method.

The overall performance improvement of methods based on the proposed validity check is not expected to be significant, mainly because the correct and incorrect region in Fig. 6, except for Tsukuba, can't be well distinguished.

## 4 More Images

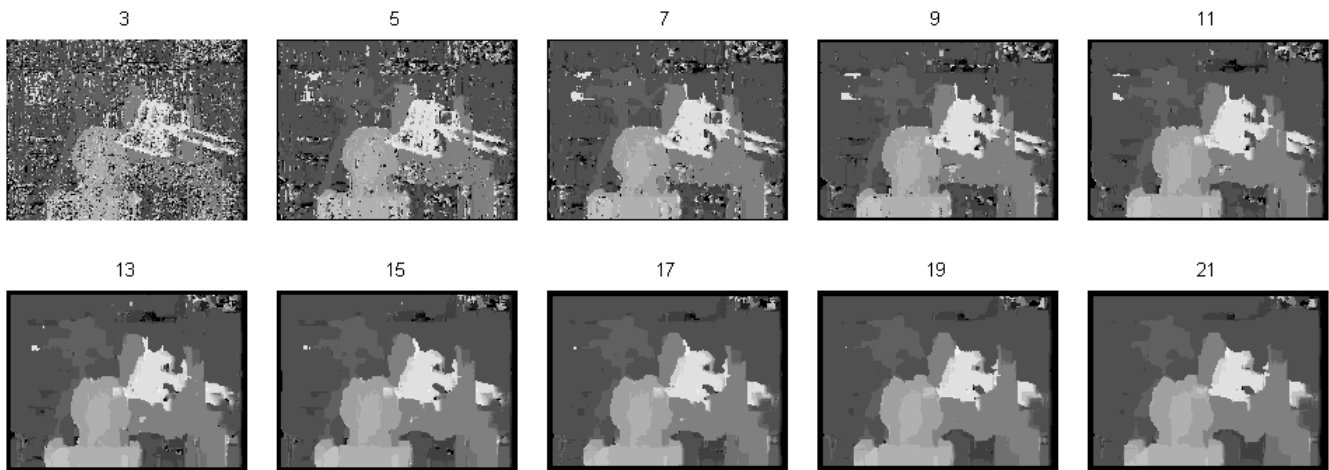


Figure 7: Tsukuba image with different window sizes.

## References

- [Fua, 1993] Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine vision and applications*, 6(1):35–49. [1](#)
- [Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42. [1](#)