

Experimentation Project: Depth Estimation from Disparity through Recursive Filtering

Martijn Houtman

November 4, 2008

Abstract

Depth estimation through iterative filtering uses a filter that refines depth maps from previous iterations. Its basis is using a bilateral filter using two penalty descriptors, the color of a region and the similarity of two disparity-matched regions. This research is focused on finding improvements of this method, mostly using a hierarchical approach. We use previously conducted experiments to compare the improved method with the original method, as well as comparing the new method with results from other methods.

Keywords: Disparity estimation, depth estimation, stereo, stereopsis, filtering, hierarchical

1 Introduction

Disparity estimation is a technique that uses two or more images from the same scene that are taken with slightly shifted camera viewpoints. In the case of two viewpoints it is called stereopsis. Because of the slightly different viewpoints and the different depths of the objects in the scene, objects in one image appear shifted compared to the other image. This is called disparity, or more general parallax. The amount of disparity is determined by the distance of the object from the camera, i.e. the depth of the object.

When the disparity between two points is known, the depth can be calculated by using a simple triangulation. This depth can be recorded for every point in the scene, giving us a dense disparity map. The relation between disparity and depth is linear, therefore the terms disparity map and depth map might be used in the same context.

A rough distinction between algorithms can be made: the ones that use feature-based matching and depth estimation and algorithms that use pixel-to-pixel correspondence. Feature based algorithms work by finding interesting points in the two images, matching them and calculate the disparity. As no image will be completely saturated by feature points, gaps will occur and will need to

be interpolated. Pixel-based algorithms work using scanlines in the images and match each pixel in the first image with a pixel in the second image, working from the left to right, top to bottom. This way, a fully dense depth map can be obtained, which keeps sharp edges in disparity intact, rather than blurring them [1].

1.1 Occlusion

Because of disparity and near and far objects, some parts of objects may be visible in one image and occluded by the the nearer object, as described in [1] and [2].

Occlusion can be quite easily detected by using unmatched feature points or pixels ([1], [3]) and mostly happen around edges or large disparity jumps.

Occluded areas often result in gaps in the depth map. In order to work with depth maps, dense depth maps are often required, so these gaps need to be filled. This is often done by interpolation, neighbour-matching or guessing.

And advantage of using a filter-based method is that these methods usually calculate really dense depth maps, even at occluded areas.

1.2 Image rectification

Stereo vision is based on human vision, which uses two eyes to capture a scene and use disparity to perceive depth. Although the eyes are at a fixed position, they can rotate individually, adjusting the optical centers. This results in images where the second image is slightly rotated around the optical center when compared to the first image.

As pointed out in literature [4], image rectification simplifies the stereo algorithms. When rectified, the relation between disparity and depth becomes a linear equation, as thoroughly described in [2]. The actual depth z of point x can be calculated with

$$d(x) = \frac{lw}{2z(x)} \quad (1)$$

where x is the feature point or pixel coordinate, $d(x)$ is the disparity at point x , l is the distance between the image plane and the optical center of the camera and w is the distance between the optical centers of the two camera's.

Rectification is done by projecting the two images onto a plane parallel to the baseline joining the two optical centers, so that the two images become parallel. In most practical appliances, stereopsis is done by using two camera's that are shifted only horizontally, thus in case of rectified images, they share scanlines. This feature can be easily exploited for example in the case of pixel-to-pixel correspondence.

2 Previous work

The basis of this research is finding optimisations in an iterative matching algorithm, originally developed by Faysal Bourhorbel [5]. This iterative filter is based on the well-known bilateral filtering method [6].

The basis of the approach is to calculate the depth recursively, averaging the computed weight from previous iterations in a spatial neighbourhood. The weights for the bilateral filter combine the color characteristics and matching confidence of the pixels in the left and right image.

The algorithm works as follows. Given two images 1 and 2, the disparity is calculated by weight-averaging the disparity from neighbouring pixels found in a previous iteration. More formal:

$$d_k(i) = \sum_{each\ j} \omega_{ij} d_{k-1}(j) \quad (2)$$

where $d_k(i)$ is the disparity at pixel i at iteration k , j are the neighbouring pixels and ω_{ij} is the weight of the pixel, just as it is done with bilateral filtering. The weight ω_{ij} consists of two penalty factors, and is formed as follows:

$$\omega_{ij} = exp(-\alpha|c_i^1 - c_j^1| - \beta|c_j^1 - c_{j'}^2|) \quad (3)$$

where c_i^1 and c_j^1 are the color values of image 1 at pixels i and j , respectively, and $c_{j'}^2$ is the color value at pixel j' , which is disparity-compensated by $d_{k-1}(j)$.

The term $|c_i^1 - c_j^1|$ penalizes the difference in color between the two pixels in image 1 so that the weights for pixels with a different color from the center pixel are low. This has the effect of averaging the disparity over pixels

that have the same color, i.e. smoothing the disparity in regions with the same color so that objects have a smooth disparity.

The term $|c_j^1 - c_{j'}^2|$ checks how much the region in image 1 matches the region in image 2, given the current disparity. This way, calculated disparities with high confidence (high match value) get a higher weight than disparities with low confidence. This is how the current disparity gets improved, by averaging with neighbouring pixels with high-confidence disparity.

By doing several iterations and doing small random updates to the current depth map each iteration, convergence is ensured. The small random updates ensure that the calculated disparities do not get stuck in a local minimum and makes the algorithm search for the optimal disparity. Too small random updates will get averaged out, too large random updates will only undo the previously improved disparity estimation. A limit to the amount of iterations can be set in order to restrict running time.

The use of a bilateral filter has been done before, as described in [7], and are ranked between the best working methods. Their recursive approach and lack of global optimisation make them excellent candidates for realtime appliances and hardware implementations.

3 Methods

The goal of this paper is to make adjustments to the original method, developed by Faysal Boughorbel, in order to improve the quality of the aquired depth maps, and to eventually compare it with existing results.

The research will be mainly focussed on the improvement by doing a hierarchical approach. As explained, the recursive approach in this algorithm uses the estimated disparity map from the previous iteration as input, do some random updates to it, and recalculate it. Estimated disparities with high confidence will be kept, while estimates with low confidence will be randomly updated and averaged with estimates of high confidence. Our research shows that using a hierarchical approach can seriously speed this up by starting out with downsampled stereo images and upsampling the resulting disparity map each iteration.

Another part of the research is to find out if using scale-spaces is a speeding factor in the hierarchical approach. The idea is to find out if scale-spaces improve the approach of using general structures first, and then improving the resulting depth map by increasing detail in the

input images. This approach can also be combined with using downsampling.

Another approach is to use anisotropic diffusion, rather than the common Gaussian filter, to represent the images at different scales. The idea behind this is to keep edges intact, while the detail inside objects is removed at first and brought back during iterations in order to improve accuracy.

4 Parameters and experimentation

The algorithm has a few parameters that can be set to adjust the outcome of the results. Worst-case, these parameters have to be experimentally determined and adjusted for each dataset. However, if datasets are consistently recorded, one parameter set will work for the whole dataset. The original paper makes no notes on what parameter values are used, and therefore the parameters in in this paper are mostly determined experimentally.

4.1 Dissimilarity penalties

The alpha and beta parameters determine the weight for each pixel in the bilateral filter: the larger the difference, the higher the penalty, the less the value of the surrounding pixel is taken into account. Alpha determines the penalty in color difference of the surrounding pixels, in order to assure smoothing in areas which have the same colors. Beta determines the penalty in difference between color between disparity-matched pixels in the left and right image. For the dataset I used, I determined values $\alpha=0.5$ and $\beta=0.8$ to work best.

4.2 Kernel size

In order to match regions, the algorithm uses a kernel to look at surrounding pixels and average the depth value, according to above mentioned penalties. Obviously, a larger kernel will result in more accurate calculations and more calculations per pixel, i.e. it will take longer to compute. In my experiments, I have found a kernel size with a diameter of 11 pixels to be the best trade-off between accuracy and speed.

4.3 Minimum and maximum disparity

Because of a maximum value of the pixels in the depth map, disparity ranges have to be set, or else the filter will record disparity values outside of the range it can store. Therefore a limit of disparity has to be set, in this case an 8-bit value, meaning 255 levels of depth. In order to

increase accuracy, this range can be divided by 5, leaving a maximum disparity range of around 50 pixels, which is enough for this particular dataset.

4.4 Search area

The size of the filter kernel influences the search area of the kernel of the bilateral filter. Logically, the larger the search area, the more precise the matching will be, but there is a certain limit to it, as the matching criteria are determined by the match of the colors, as to smooth the disparity over areas with similar colors.

Also, the size depends on the dimensions of the images. Experimentation shows that a kernel size between 20 and 40 pixels is enough to get enough detail without the drawback of the algorithm getting too slow.

4.5 Random updates

The paper gives no notes on exactly how much random updates are added during the iterations in order to ensure convergence, just that it the random updates are "small". During our experiments we found out that the updates should be $\pm 10\%$ of the disparity values in order to make at least some improvement without adding noticeable noise to the resulting depth map.

During our research, we noticed that there was some noise in our resulting depth maps, especially near hard edges. This was caused by the random updates that were done each iteration. This is easily solved by linking the random updates to the iterations, making the size of the updates smaller at later iterations, so that in the beginning updates were big, and the updates became smaller and smaller during the next iterations. This resulted in more smooth disparity maps, without interfering with the convergence.

4.6 Padding

Because of our convolution filter approach, padding is involved. There is no mention of padding in the original paper. At first we used padding to infinity, i.e. replicating edge pixels. This however had the drawback of pixels matching outside the image borders, especially if there are big, homogenous areas near the borders.

Therefore we decided to use zero-padding, with added penalty to the weights if a pixel is matched with a pixel outside of the image region. This results in black areas in the occluded areas near the borders, but since these areas are not taken into account in the tests, this does not matter.

5 Results

The dataset used in this experimentation are images taken from the Middlebury Stereo Vision page ([8]). These images are specially recorded images for the purpose of evaluating stereoscopy methods and comparing the results. The images are well-lit, rectified images of tabletop scenes of objects with different colors and textures.

The dataset contains four medium-sized, regularly used images of different scenes under the same circumstances, taken with calibrated cameras. The first one is called Tsukuba, which is a version of the widely-used scene originating from the university of Tsukuba. The second image is called Venus, which is a scene with flat surfaces of posters and newspapers with a lot of texture. The third scene is called Teddy, which is a scene that contains colorful teddy bears. The fourth scene is called Cones, a scene which contains coned-shaped objects and some pottery.

Our experimentation and the results are based on previous comparisons of stereo algorithms, as described in [9]. The results of these tests can be found and compared online at [8]. They also provide the ground truths they used in their evaluation. As in the original experiment, we use the left image as a reference image and calculate the disparity map for this image.

The evaluation of the resulting depth maps, comparing them with the ground truths, as described in the paper above is done on a pixel-to-pixel comparison between the ground truth and the calculated depth map. The following metric is used to compare the results (percentage of bad pixels):

$$B = \frac{1}{N} \sum_{x,y} (|d_C(x,y) - d_T(x,y)| > \sigma_d) \quad (4)$$

where d_C is the calculated disparity value, d_T is the ground truth disparity, N is the total amount of pixels and $\sigma_d = 1$, calculating only the pixels that differ at least one disparity level from the ground truth. This metric thus counts the percentage of non-matching pixels.

Another metric that is mentioned, but not shown in the results of [9], is the Root Mean Squared error (RMS):

$$R = \sqrt{\frac{1}{N} \sum_{x,y} (|d_C(x,y) - d_T(x,y)|^2)} \quad (5)$$

However, the calculated disparity map from our algorithm does not match the ground truth disparity map on an absolute level, but rather on a relative basis, meaning

ranges of the calculated disparity might in the ground truth might be [0..15], whereas our resulting depth map might have ranges [10..41]. With the first two metrics, equations 4 and 5, may give a high response with this case even though the relative disparity values are correct.

This is where the third metric, standard deviation, comes in. The standard deviation metric measures the standard deviation of the absolute differences between the ground truth and the calculated depth maps. Because it only takes the dispersion into account, it is invariant to translation. However, it is still variant to scaling. The metric can be formulated as:

$$S = \sqrt{\frac{1}{N} \left(\sum_{x,y} \delta_{x,y}^2 \right) - \bar{\delta}_{x,y}^2} \quad (6)$$

where $\delta_{x,y} = |d_C(x,y) - d_T(x,y)|$, the absolute difference between the two depth maps.

We give both statistical (1 and visual (5) comparisons of the extensions of the original method. Four versions are being compared: nonhierarchical, nonhierarchical + blur, hierarchical and hierarchical + blur. Nonhierarchical is the method as it was originally described by [5]. Nonhierarchical + blur is the original method with scale-spaces using a Gaussian kernel. Hierarchical is the original method using a hierarchical approach by starting out with scaled-down images and refining the depth map by using images of larger dimensions each iteration. Hierarchical + blur is the same, but it combines scale-spaces with it.

We also include a comparison with other methods in tables showing results of Tsukuba 5, Teddy 5, Venus and Cones 6. The method of comparison is derived from the Middlebury Stereo Vision pages ([8]). The results are shown in figure 2.

6 Conclusions

The purpose of this research is to find improvements to the originally proposed method, as implemented by [5]. We researched a few adjustments, such as scale-spaces, a hierarchical approach and combinations thereof. The regular version of scale spaces uses a Gaussian kernel, but we have also looked at using anisotropic diffusion on different levels.

The original method has some pitfalls, as you can see in the third column of table 2, with the biggest pitfall being large, opaque areas, where the algorithm mismatches a

Rank	Method	nonocc	all	disc	Rank	Method	nonocc	all	disc
1	CoopRegion	0.87	1.16	4.61	1	SubPixDoubleBP	9.18	10.67	20.78
2	DoubleBP	0.88	1.29	4.76	2	2OP+occ	9.48	10.85	21.44
3	OutlierConf	0.88	1.43	4.74	3	CostAggr+occ	9.87	11.04	21.85
4	EnhancedBP	0.94	1.74	5.05	...				
5	SymBP+occ	0.97	1.75	5.09	34	GC	14.13	19.36	30.77
...					35	SemiGlob	14.17	15.21	30.89
43	SO	5.08	7.22	12.18	36	PROPOSED	14.21	14.4	26.69
44	6g	5.08	7.22	12.18	37	TreeDP	14.79	17.93	31.38
45	SymBP+occ	5.23	7.07	24.14	38	2OP+occ	14.93	20.12	33.51
46	STICA	7.7	9.63	27.83	...				
47	Infection	7.95	9.54	28.88	46	STICA	19.73	21.79	38.38
48	PROPOSED	38.6	39.17	43.69	47	Infection	20.69	22.54	41.01
					48	SymBP+occ	21.61	24.51	47.49

Table 1: Tsukuba data set using the bad pixels metric (left) and the standard deviation metric (right), both less means better. Note the high score (low ranking) of the bad pixels metric for our proposed method, and the more fair score with standard deviation.

Rank	Method	nonocc	all	disc	Rank	Method	nonocc	all	disc
1	SubPixDoubleBP	3.45	8.38	10.01	1	DoubleBP	2.43	6.1	4.54
2	DoubleBP	3.53	8.3	9.63	2	SubPixDoubleBP	2.48	6.19	4.58
3	19	3.55	8.71	9.7	3	AdaptingBP	2.51	3.26	4.7
4	AdaptingBP	4.22	7.06	11.79	...				
5	Segm+visib	5	6.54	12.33	32	Infection	10.49	12.02	16.12
...					33	STICA	10.53	23.19	13.88
44	Infection	17.7	25.09	44.41	34	PROPOSED	10.69	22.25	16.19
45	RegionalSup	18.31	26.72	32.09	35	Layered	11.55	22.07	13.85
46	SO	19.95	28.17	26.26	36	VariableCross	11.96	14.61	13.74
47	PhaseDiff	19.98	28	29	...				
48	PROPOSED	67.82	69.99	59.26	46	PhaseBased	16.82	33.05	19.9
					47	SO	16.97	33.48	22.08
					48	PhaseDiff	20.57	30.29	17.93

Table 2: Teddy data set using the bad pixels metric (left) and the standard deviation metric (right). Again, the standard deviation gives a more fair score.

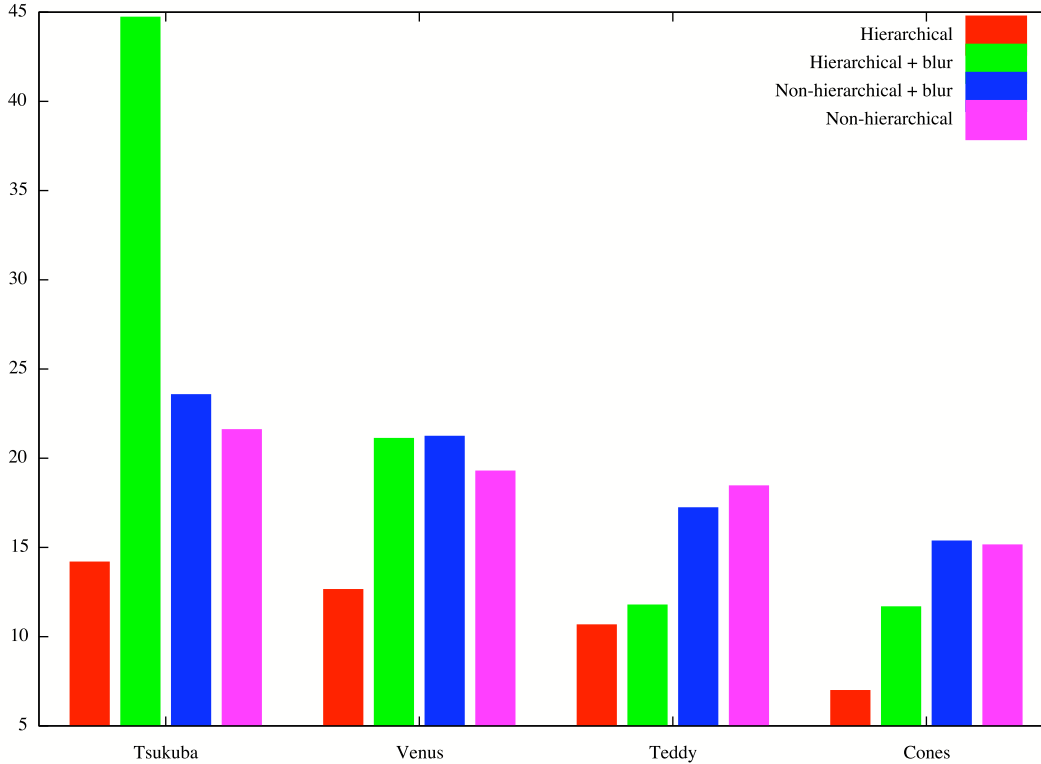


Figure 1: Comparison of the new methods using the standard deviation metric (less is better). As you can see, the differences between the non-hierarchical approaches and the non-hierarchical approaches are significant for all three datasets, and in general the hierarchical approach, with our without blurring, has a better score, except for the Tsukuba dataset.

point in this area with another point in the area, which does not have the exact same location. Adding small random updates in this case does not help, because any random update will match the pixel with another pixel in this opaque area, where the colors are the same, and therefore the penalty will be small, so any random update will be as good as the previous one.

Another big pitfall is that areas which share the same depth, but have different colors in it, get assigned different disparities. For a good example, look at the second image, third column. The poster on the left has a green background and several small rectangular images on it. These small images have a different color than the poster's background, and are therefore treated as separate objects. Because the algorithm only takes pixels within the same object into account, it might be possible for the disparity to stick in a local minimum, where even the small random updates get cancelled out.

This pitfall could be addressed using scale-spaces, where details get filtered out in lower levels and being brought

back in at higher levels to obtain a more detailed disparity map. As you can see in the results, this is the case, but still the poster does not have an opaque disparity, as it should have.

The hierarchical approach starts out with a global disparity map and increases the detail by scaling up the previous disparity map to refine it in the next iteration. This has the advantage that objects will start out small and get bigger and therefore more detailed in the next iterations, such that the disparity map can be more refined later on. This makes objects have a more opaque disparity, as can be visually confirmed, and also has the advantage that the algorithm takes much less time to compute.

As can be seen in graph 1, for each image from the dataset, the hierarchical approach gives better results, usually between 30% and 50% better. Generally, the hierarchical approach with blurring also gives better results, apart from one peak in the Tsukuba image. This peak can be visually confirmed in 5: the image is much darker

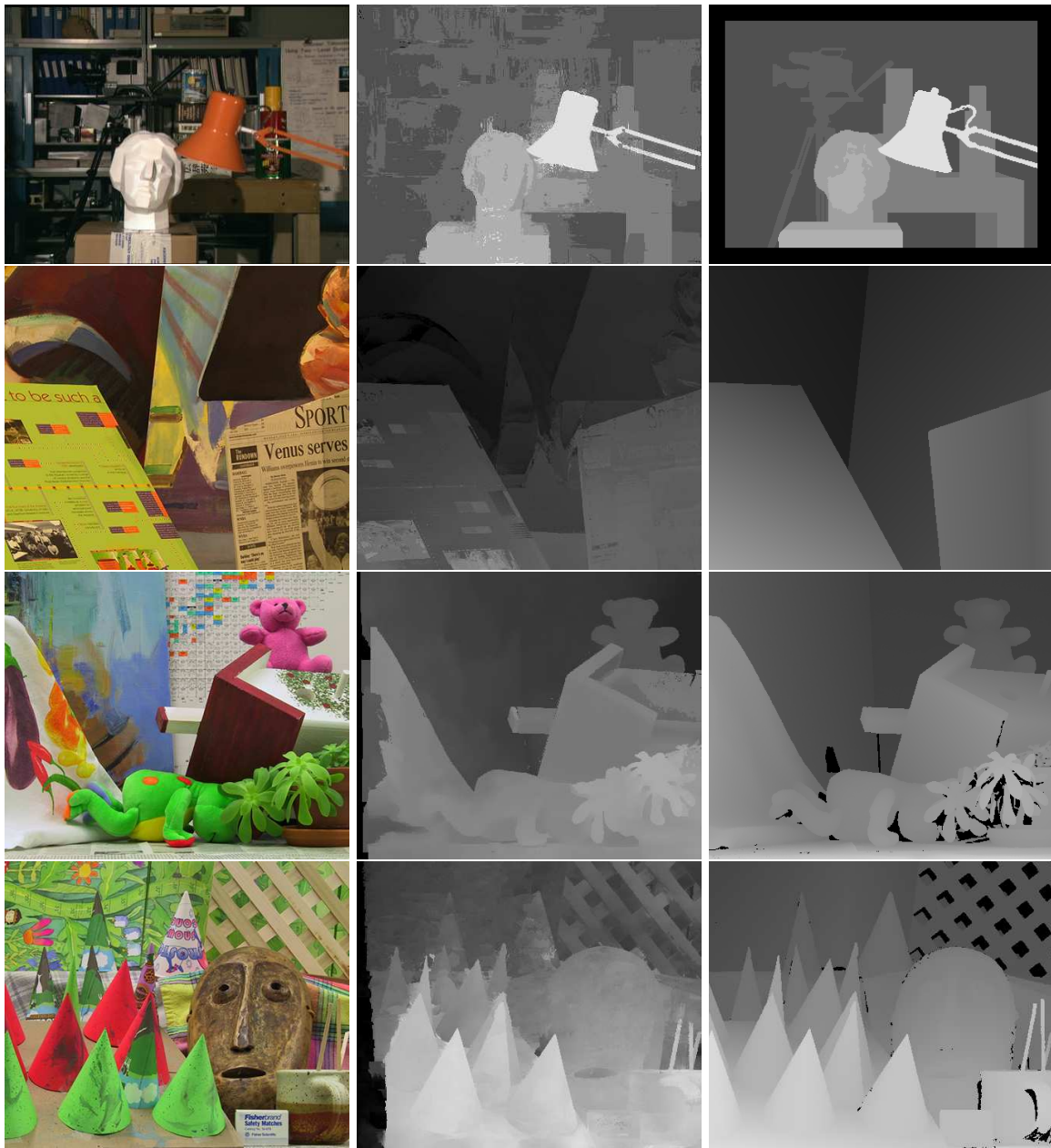


Figure 2: Results with the image in the left column, the result in the center and the ground truth on the right



Figure 3: Comparison of our result versus SymBP+occ and Infection. Even though the latter two have better scores, the visual result of our proposed method is much more accurate when it comes to object edges.

and only finds the head as an object. The results of the hierarchical + blur approach also performs less than the plain hierarchical approach.

We give a comparison between the results measured with metric of equation 4 (bad pixels) and 6 (standard deviation). Standard deviation gives us a far more fair rank than the percentage of bad pixels does (see table 5). This is visually supported by comparing the resulting disparity maps, as seen in figure 3: our method (left) gives much more visually appealing results than the other two (center and right), but the percentage of bad pixels metric ranks us below the other two methods, however, the standard deviation ranks us above them.

We also show a graphical comparison in figure 4. This graph shows that the peak gets evened out in the case of the standard deviation metric, while preserving the general shape of the graph of the bad pixels metric.

Even though the other two algorithms have a better score, when compared visually, our result looks much more detailed and more precise, especially at the edges of object, even though the absolute disparity values differ more from the ground truth than with the other two depth maps. This is also confirmed by looking at the results on a Philips 3D screen ([10]), where our results look better than the other two examples, especially because of the more accurate edges and the lack of "holes" around occluded areas. Ofcourse, it all depends on the application of the depth maps, whether you need more accuracy at region edges or want accurate more accurate absolute values.

With proper disparity range scaling and some post-processing, like subpixel refinement by polynomial fitting [11], we would probably have been able to get more accurate results, but this paper is about the quality of the algorithm itself, not about the quality of the refinement algorithm.

References

- [1] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. 1997.
- [2] Peter N. Belhumeur. A bayesian approach to binocular stereopsis. 1994.
- [3] Stephen T. Barnard and Martin A. Fischler. Computational stereo. 1982.
- [4] David Forsyth and Jean Ponce. *Computer Vision - A modern approach*. 2003.
- [5] Faysal Boughorbel. Adaptive filters for depth from stereo and occlusion detection. 2008.
- [6] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision*, pages 839–846, 1998.
- [7] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewénius, and David Nisér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.
- [8] Middlebury stereo vision page. <http://vision.middlebury.edu/stereo/>.
- [9] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. 2002.
- [10] Philips lenticular autostereoscopic 3d display. <http://www.philips.com/3dsolutions/>.
- [11] Q. Yang, J. Davis R. Yang, and D. Nistér. Spatial-depth super resolution for range images. 2007.

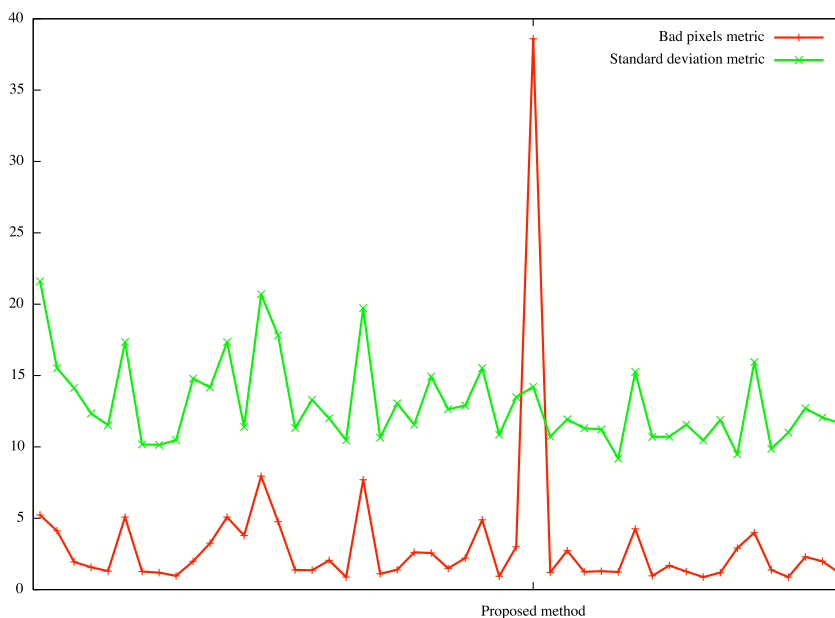


Figure 4: Graphical representation of the two metrics used in this paper. Note that the proposed standard deviation metric keeps the general shape intact, but the spike at 30 (our method) is being reduced to a more normal level. Please note that these numbers are about relative values, not absolute values.

Rank	Method	nonocc	all	disc	Rank	Method	nonocc	all	disc
1	SubPixDoubleBP	0.1	0.26	1.35	1	ImproveSubPix	0.53	2.25	1.61
2	2OP+occ	0.13	0.18	1.76	2	19	0.58	4.64	1.78
3	AdaptOvrSegBP	0.09	0.11	1.24	3	DoubleBP	0.58	4.19	1.78
...					...				
37	STICA	1.95	2.73	13.31	14	SegmentSupport	0.84	3.7	2.39
38	Infection	1.86	2.44	17.71	15	SO+border	0.85	3.13	2.44
39	GC+occ	1	1.38	5.49	16	Segm+visib	0.92	2.89	2.6
40	PROPOSED	3.71	4.08	8.77	17	PROPOSED	0.95	7.03	2.63
41	PhaseBased	3.11	4.03	15.99	18	AdaptOvrSegBP	0.97	2.35	2.77
42	SymBP+occ	3.05	3.94	9.91	19	SymBP+occ	0.99	2.85	2.85
43	RegionalSup	3.66	4.89	21.76	20	CostAggr+occ	1.09	2.09	3.13
...					...				
46	SO	4	5.18	12.97	46	Infection	3.84	5.75	10.96
47	6	4	5.18	12.97	47	SO	3.92	12.11	7.11
48	DP	5.46	6.28	13.8	48	PhaseDiff	7.14	13.46	9.7

Table 3: Venus dataset using the bad pixels metric (left) and Cones dataset using the standard deviation metric (right).

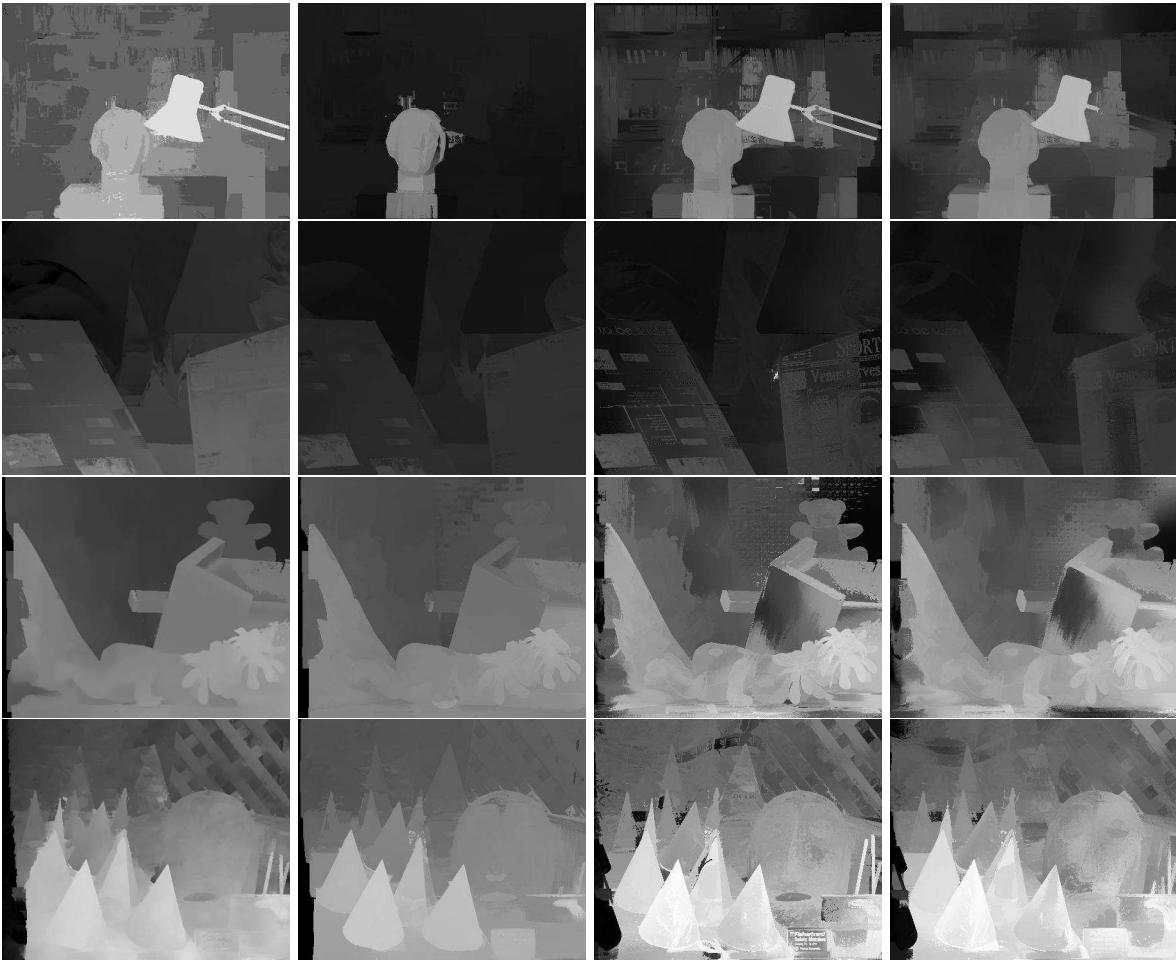


Figure 5: From left to right: Hierarchical, Hierarchical + blur, Non-hierarchical, Non-hierarchical + blur