AN EVALUATION OF STEREO MATCHING METHODS FOR VIEW INTERPOLATION

G. Führ1, G. P. Fickel1, L. P. Dal’Aqua1, C. R. Jung1, T. Malzbender2, R. Samadani2

1 Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
2 Multimedia Communications and Networking, HP Labs Palo Alto

ABSTRACT
Stereo matching has a long history in image processing and computer vision. In fact, there are innumerable approaches reported in the literature, and quantitative evaluation is usually performed by comparing the obtained disparity maps with ground truth data (using the MSE, for instance). One important application of stereo matching is view interpolation, where it is desired to produce a new synthetic view from (at least) a pair of images and the corresponding disparity maps. In view interpolation, evaluation is mostly qualitative (visual quality of the synthesized image), and quantitative approaches compute objective similarity metrics between the synthesized image and the actual image at the same position (e.g. PSNR). The main goal of this paper is to evaluate the impact of several different stereo matching algorithms in a view interpolation context, relating the quality of the disparity maps with the quality of the corresponding synthesized views using standardized datasets. In this paper, experiments using the MPEG reference software for view interpolation and more than twenty datasets are presented and discussed. Our results indicate that the use of the common percentage of bad pixels as a metric for stereo matching methods does not translate well to the quality of view interpolation.

Index Terms—view interpolation, stereo matching, quality assessment

1. INTRODUCTION

The possibility of generating multiple interpolated views is very useful and can be used in several areas, such as sports broadcasting, movies industry, surveillance systems, etc. A common approach to generate the interpolated views is to join the information of one or more cameras with the 3D information of the scene, usually encoded by a disparity map.

However, view interpolation and disparity estimation are often tackled as two disjoint problems. Stereo matching algorithms are evaluated (and ranked) according to objective error metrics computed between the generated disparity map and ground truth data available for some standardized datasets. A popular example of publicly available dataset with ground truth data, as well as objective error metrics for benchmarking is the Middlebury benchmark [1]. For view interpolation algorithms, the ultimate validation is visual (does the synthesized view look real?), but it is also common to use quantitative metrics for quality assessment, such as MSE or PSNR [2]. Clearly, the quality of the generated view depends on the quality of the corresponding disparity maps.

Even though the Middlebury benchmark is a useful platform to quickly evaluate and compare different existing disparity estimation techniques, its primary metric does not correctly evaluate all the characteristics that a good disparity map must have for view interpolation techniques. For instance, in the default metric used to rank the algorithms, each pixel is classified as either correct or incorrect according to the ground truth (using a given error threshold). However, for the interpolation process, smaller errors in the disparity estimation will probably produce less visible artifacts than pixels with larger errors. Also, the smoothness of the disparity map, which may potentially impact the quality of the synthesized view, is not evaluated by the Middlebury error metric.

In [3], the authors evaluated whether the error of view interpolation is a good indicator of the quality of the disparity maps used to generate the synthesized views, concluding that it is not necessarily true. Hence, it is natural to ask the complement: will a stereo matching technique that produces good disparity maps (according to traditional quality measures such as those adopted/proposed by the Middlebury benchmark) produce good quality synthesized views when used for view interpolation? This is a relevant question, since considerable effort is being devoted to decrease the disparity map error (usually at the cost of high computational complexity), and it may not improve the visual quality in view interpolation applications. This paper aims to evaluate several disparity estimation techniques, relating the quality and characteristics of the obtained disparity maps and the interpolated views. In this paper, the MPEG View Synthesis Reference Software (VSRS) [4] was used to generate all interpolated views, so the only variables are the different stereo matching algorithms.

2. DISPARITY ESTIMATION METHODS

In our evaluation we aim to test a wide variety of disparity estimation techniques, within a wide range in the Middlebury benchmark results. We have also selected techniques that pro-
vide source code (or executable binary) to reproduce the results contained in the Middlebury site, aiming to cover the two classes of stereo matching approaches: global and local.

Global techniques usually model the disparity estimation problem as the minimization of a global energy function that encodes evidences of the disparity map from the stereo pair as well as other information (such as smoothness, occlusions, etc.). To avoid the costly minimization procedure, local techniques use only the local information around a given pixel to estimate its disparity value. The traditional framework is to use a simple (and usually fast and suitable for parallelization) region matching technique, and filter the resulting costs with a more sophisticated technique, called aggregation step according to the taxonomy of [5]. In general, local methods present a noisier result when compared to global methods, but tend to be faster. Next we briefly describe some global and local approaches that were evaluated in our experiments.

One common solution in global methods is to use Belief Propagation, but the computation cost quickly becomes a problem. To avoid this limitation, [6] proposes a looping belief propagation that solves the message updating problem in a time linear in the number of the pixels and using constant memory space.

Another popular technique explored by global methods is graph-cut. In both [7] and [8], the energy function presents three terms: the data term that represents the difference between two corresponding pixels, the smoothness term that makes neighboring pixels tend to have similar disparities and the occlusion term, which imposes a penalty for making a pixel occluded. Both techniques present good and very similar results. It was also proposed in [9] the use of a hierarchical graph cut due to its faster computation speed. This technique presents some limitations, as it depends on the initial value when there are many local minima. Despite its increase in computational time, the results are worse than the other graph cut techniques [7, 8].

In a different approach made by Woodford and colleagues [10], it was used Second Order Smoothness Priors to model the scene. Using an optimization based on the Quadratic Pseudo-Boolean Optimization (QBPO), this technique was able to achieve better results than the reviewed graph cut methods.

Aiming to avoid both the computational cost from energy minimization techniques and the limitations of local techniques, Yang [11] proposed to aggregate the matching costs based on pixel similarity in a tree structure derived from the stereo image pair. The nodes of this tree are the image pixels, and the similarity between any given two pixels is their shortest distance in the tree. Using this non-local approach, Yang’s method was able to outperform most of the local techniques while presenting a very fast algorithm.

Recent local techniques have explored the aggregation step to improve the disparity map’s quality. For instance, Richardt and collaborators [12] used the bilateral filter in the aggregation step not only in the neighborhood of a pixel, but with its projection in the second image. They also explored the concept of dual bilateral filter, which smooths an image (disparity) with respect to edges in a different image (original color image). Their approach presented good results and it runs relatively fast.

In order to achieve the bilateral filter’s good results in the aggregation but avoiding excessive computational cost, Mattoccia et al. [13] proposed a fast approximation of the bilateral filter. Using lookup tables to avoid unnecessary computations, it was possible to achieve good results with a small processing time.

Another solution for the aggregation step was developed by [14]. It proposes a general labeling framework that, given a cost volume (height \times width \times disparity), uses an edge preserving guided filter for the aggregation followed by a winner-take-all strategy. Finally, the results of the disparity map are improved with a refinement step that, in the image domain, finds the occluded areas using the cross-check procedure.

Next, we present our experimental setup, along with the obtained results.

3. EXPERIMENTAL SETUP

The experimental setup is illustrated in Figure 1. Using multi-baseline datasets provided in [1], we select subsets of three images. The left and right images are chosen as input for the experiment and, using the stereo matching method that is being tested, two disparity maps are created using either images as reference. These disparity maps, along with the reference images, are the input for the view interpolation software, which produces a synthesized view with the virtual camera placed exactly at the location of the central image (reserved for validation). The virtual and the actual images are then evaluated quantitatively using both the PSNR error and the referenced structural image quality metric (SSIM) proposed by [15], which is expected to produce quantitative results that correlate better with the human visual system, particularly with the artifacts present in the interpolation process. Errors such as distortions of object shapes and ghosts could possibly produce a high PSNR value, since they may occur only in small portions of the image, whereas they are expected to present a larger error when using SSIM.

It is important to clarify that the MPEG View Synthesis Reference Software (VRS) [4] was used to generate the virtual views for all experiments, but our experimental setup could be repeated using another view interpolation method as well. Our results are shown next.

4. RESULTS AND DISCUSSION

In order to carry out the experiments in a fair manner, we only used methods in the Middlebury ranking for which the implementation or source code were publicly available. We tested a
total of 9 methods (3 local and 6 global methods) sparsely distributed in the ranking list, and briefly described in Section 2. The parameters of the methods were set to the values specified in their corresponding papers. For parameters not specified in the paper, we used the default values proposed in the implementation. The experiments were run on all the datasets in the so-called 2006 Middlebury datasets category\(^1\), totaling 21 pairs of images. The choice of this group of datasets is mainly due to the several challenges that are present in the sequences and because camera calibration is also provided (what is needed for VSRS). In order to compare the disparity estimates with the ground truth that is provided with the datasets, View1 and View5 were used as the left and right images.

Table 1 shows the PSNR and SSIM values for the experiments using the Middlebury 2006 datasets. The methods are listed in the order they appear in the Middlebury website, i.e. the first row is the method with the best result using the metric of average of bad pixels (ABP)\(^2\), and so on. The exact place in the ranking is the number between parentheses. It is important to mention that this ranking is constantly changed and the values expressed in the table are the values as in January of 2013.

One of the aspects that can be clearly noticed from the results presented in Table 1 is that the Middlebury ranking does not directly correspond to the order of the PSNR values or the SSIM values, i.e. some of the methods in the lower half portion of the ranking present better quality (in terms of PSNR/SSIM) than some of the methods placed above them in the ranking. A plot relating the Middlebury metric for disparity maps and the SSIM of the interpolated view is shown in Figure 2, and it can be observed that several images with good SSIM values (close to one) were generated by stereo matching algorithms with a wide quality variation.

We have also computed the Pearson correlation coefficient between the disparity map quality measure (ABP) and the image quality metrics used in our experiments (PSNR and SSIM). Correlation value between ABP-PSNR and ABP-SSIM were, respectively \(-0.29\) and \(0.04\). In particular, the correlation ABP-SSIM was very low, and surprisingly, positive: since ABP is an error metric and SSIM a quality metric, the function that relates them should be monotonically decreasing.

\(^1\)http://vision.middlebury.edu/stereo/data/scenes2006/

\(^2\)A “bad pixel” is detected if its disparity value differs from ground truth by more than \(T\) pixels. In all experiments, we used \(T = 1\), as the default approach in the Middlebury ranking.

Based on the experiments and the characteristics explored by the evaluated stereo matching algorithms, we believe that two important factors must be present in disparity maps to generate good view interpolation results. First, discontinuities in disparity must be correctly estimated and placed at the right location in the map. Second, the map should be smooth in areas of constant depth to avoid artifacts in the final interpolation. Since smoothness is typically included in the optimization function explored by global methods, such class of approaches tend to produce better view interpolation results. In fact, our experiments corroborate with the second observation: global methods did generate the best overall results for interpolation. This is more evident in datasets where objects
As for the default metric used in the Middlebury ranking, it should be noticed that a pixel is binarily classified as a good or bad pixel, so the metric lacks the information of how bad (or good) a given disparity value is. Hence, a “bad” pixel may present a disparity error just above the threshold, so that view interpolation results may not be affected significantly, or an error way above the threshold, possibly creating artifacts in the synthesized view.

5. CONCLUSIONS AND FUTURE WORK

In this work, we evaluated the quality of a set of stereo matching methods that are listed on the Middlebury ranking applied to the problem of view interpolation. Our experimental results indicate that the default metric used in the Middlebury ranking (ABP) presents low correlation with the perceptual image quality index (SSIM) used to evaluate the interpolated views, so that care must be taken when devising a stereo matching technique for the purpose of view interpolation.

We have also observed some characteristics of stereo matching algorithms that tend to produce better synthesized views, such as smoothness of the disparity map. In fact, our experiments indicated that global stereo matching methods tend to outperform local ones. However, it should be noticed that these methods are often computationally expensive, which can be prohibitive in a number of applications, particularly real-time view interpolation. As future work, we intend to study the design of metrics that can evaluate disparity maps in a manner that is correlated with the quality of the synthesized image, possibly including the characteristics observed in our experiments.

### Table 1. PSNR and SSIM values for the datasets tested. Bold values are the best values for each given dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>Aloe</th>
<th>Baby1</th>
<th>Baby2</th>
<th>Baby3</th>
<th>Bowling1</th>
<th>Bowling2</th>
<th>Cloth1</th>
<th>Cloth2</th>
<th>Cloth3</th>
<th>Cloth4</th>
<th>FlowerPots</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>24.96</td>
<td>27.06</td>
<td>25.99</td>
<td>27.48</td>
<td>24.62</td>
<td>26.91</td>
<td>27.03</td>
<td>24.66</td>
<td>25.69</td>
<td>23.51</td>
<td>27.16</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.838</td>
<td>0.905</td>
<td>0.918</td>
<td>0.891</td>
<td>0.941</td>
<td>0.909</td>
<td>0.851</td>
<td>0.863</td>
<td>0.861</td>
<td>0.815</td>
<td>0.932</td>
</tr>
<tr>
<td>CostFilter (#25)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.810</td>
<td>0.851</td>
<td>0.892</td>
<td>0.888</td>
<td>0.929</td>
<td>0.902</td>
<td>0.842</td>
<td>0.863</td>
<td>0.861</td>
<td>0.819</td>
<td>0.907</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.796</td>
<td>0.850</td>
<td>0.864</td>
<td>0.882</td>
<td>0.905</td>
<td>0.838</td>
<td>0.823</td>
<td>0.667</td>
<td>0.812</td>
<td>0.741</td>
<td>0.899</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.862</td>
<td>0.884</td>
<td>0.898</td>
<td>0.893</td>
<td>0.922</td>
<td>0.884</td>
<td>0.833</td>
<td>0.849</td>
<td>0.864</td>
<td>0.807</td>
<td>0.925</td>
</tr>
<tr>
<td>GC+occ (#92)</td>
<td>PSNR</td>
<td>24.10</td>
<td>26.69</td>
<td>24.78</td>
<td>25.84</td>
<td>23.95</td>
<td>24.97</td>
<td>27.05</td>
<td>22.99</td>
<td>25.63</td>
<td>22.15</td>
<td>24.04</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.797</td>
<td>0.842</td>
<td>0.816</td>
<td>0.812</td>
<td>0.908</td>
<td>0.864</td>
<td>0.865</td>
<td>0.692</td>
<td>0.851</td>
<td>0.794</td>
<td>0.883</td>
</tr>
<tr>
<td>MultiCamGC (#94)</td>
<td>PSNR</td>
<td>24.23</td>
<td>25.51</td>
<td>24.96</td>
<td>26.72</td>
<td>24.29</td>
<td>25.27</td>
<td>27.03</td>
<td>24.04</td>
<td>25.45</td>
<td>22.95</td>
<td>25.54</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.801</td>
<td>0.561</td>
<td>0.840</td>
<td>0.879</td>
<td>0.922</td>
<td>0.883</td>
<td>0.856</td>
<td>0.819</td>
<td>0.836</td>
<td>0.803</td>
<td>0.896</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.838</td>
<td>0.814</td>
<td>0.870</td>
<td>0.834</td>
<td>0.896</td>
<td>0.821</td>
<td>0.868</td>
<td>0.808</td>
<td>0.819</td>
<td>0.758</td>
<td>0.886</td>
</tr>
<tr>
<td>DCBGrid (#118)</td>
<td>PSNR</td>
<td>23.72</td>
<td>26.43</td>
<td>24.94</td>
<td>26.89</td>
<td>23.64</td>
<td>25.09</td>
<td>23.03</td>
<td>22.98</td>
<td>23.55</td>
<td>21.35</td>
<td>26.09</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.782</td>
<td>0.858</td>
<td>0.891</td>
<td>0.887</td>
<td>0.918</td>
<td>0.870</td>
<td>0.786</td>
<td>0.720</td>
<td>0.823</td>
<td>0.757</td>
<td>0.919</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.781</td>
<td>0.824</td>
<td>0.893</td>
<td>0.866</td>
<td>0.890</td>
<td>0.871</td>
<td>0.781</td>
<td>0.797</td>
<td>0.819</td>
<td>0.754</td>
<td>0.928</td>
</tr>
</tbody>
</table>

The authors would like to thank Brazilian agencies CNPq and Capes as well as HP Brazil R&D for supporting this work.
7. REFERENCES


