Comparative analysis of local binocular and trinocular depth estimation approaches

Sergey Smirnov [a], Atanas P. Gotchev [a], Miska Hannuksela [b]
[a] Dep. of Sig. Proc., Tampere Univ. of Technology, Tampere, Finland;
[b] Nokia Research Center, Nokia Corp. Tampere, Finland.

ABSTRACT

In this paper, we present a comparative analysis of local trinocular and binocular depth estimation techniques. Local techniques are chosen because of their higher computational efficiency compared to global approaches. Our aim is to quantify the benefits of the third camera with respect to performance and computational burden. We have adopted the color-weighted local-window approach in stereo matching, where pixels within local spatial window around the pixel being processed are penalized by their colors in order to ensure better adaptivity to local structures. Thus, the window size becomes the main parameter which influences the quality and determines the execution time.

Extensive experiments on large set of data have been carried out to test trinocular versus binocular setting in terms of quality of estimated depth and execution time. Both natural and artificial scenes have been tested. A set of quality measures has been used to support the comparisons. MPEG Depth Estimation Reference Software has been used as a reference benchmark as well. Results show that from some window size on, the trinocular setting outperforms the binocular in general: providing higher quality for less computational time. While comparisons were done for 'pure' depth estimation, we also run post-processing on depth estimates in order to analyze the potential of estimated depths to be further improved.

Keywords: depth estimation, binocular, trinocular, color-weighted aggregation

1. INTRODUCTION

Depth information plays a key role in 3D video and Free-viewpoint TV technologies. Given a depth map associated with color frame, a visualization device may render novel virtual views of the scene within certain range to enable stereoscopic observation or free-viewpoint selection. Depth maps are usually acquired either using active-vision approaches (structured light or time-of-flight camera) or using passive vision, when depth is estimated from two or more video streams. The latter approach is more applicable because it does not require a specific hardware sensors and obtained color frame may be directly used along with the estimated depth forming ‘color plus depth’ representation.

Depth estimation based on two-camera (binocular) setting is a problem, which has been adequately addressed in a number of works [1][2][3]. The achieved quality of binocular matching is high and approaching the quality of depth acquired by active-vision methods. However, this is achieved for the prices of high computational costs. Modern matching algorithms generally employ comprehensive post-processing steps along with non-local optimization of matching costs (e.g. graph cuts or loopy belief propagation).Tools such as segmentation, iterative post-filtering and RANSAC fitting are used in order to deal with occlusions, outliers and low-confidence matches.

When computational cost and real-time operation are of primary concern, simplified methods have to be applied, which reduces the performance considerably [4]. Alternatively, implementations based on massive parallelization of calculations and used of GPUs or multi-core processors have been targeted [5][6][7].

An inherent drawback of binocular matching approaches is the impossibility to estimate depth in occluded areas of the scene, i.e. areas which are not visible from both cameras. For such cases, a standard remedy is to perform a kind of scene segmentation and to fit a plane within each segment assuming certain degree of smoothness of the depth function inside the segment. An attractive alternative is to extend the hardware to a trinocular camera arrangement. A view from a third camera in the camera array can tackle successfully occlusion problems and can contribute also to refining the precision of already estimated depth pixels. From computational point of view, addition of a third view can be compensated by reducing the size of the search window in local stereo-matching approaches.

The problem of multi-camera depth estimation has been quite intensively researched in recent years. Novel algorithms have been suggested aiming at recovering depth of higher quality and with improved computational efficiency. Approaches delivering computational time, linear to number of cameras have been suggested [8], as well as approaches based on multi-camera calibration [9] [10] or on trifocal tensors [11]. However, the effect of adding a third camera and going from binocular to trinocular setting has not been quite fully studied and quantified.

In this paper we aim at qualifying the effect of extended trinocular setting on the depth estimation quality and computational efficiency compared to classical binocular setting. For both binocular and trinocular settings we employ a local estimation approach as it is less computationally demanding than approaches based on global optimization. The approach utilizes a mixed spatial and color-based filtering while finding best local match. We compare it with the MPEG's Depth Estimation Reference Software (DERS) [12] which uses graph cuts for global optimization of the matching cost. Additionally, we present results of applying an eventual post-processing stage to improve the quality of an already estimated depth based on structural and color constraints coming from the associated color (texture) frame [13], [14]. The obtained results illustrate the good performance of the chosen approach. It is quite competitive to MPEG DERS and offers scalability through the possibility of selecting proper size of the processing window.

2. DEPTH ESTIMATION APPROACHES

Consider two or three images in YUV color space corresponding to binocular or trinocular scene observation with left, central and right camera, $\mathbf{y}_P(\mathbf{x}) = \left[y_P^Y(\mathbf{x})\ y_P^U(\mathbf{x})\ y_P^V(\mathbf{x}) \right]$, where $\mathbf{x} = [x_1\ x_2]$ is spatial variable and $P = \{L, C, R\}$ is the camera position. The baselines between central and left, and central and right camera are equal and the images are linearly rectified [9, 10]. For the binocular setting, the disparity associated with the central camera is obtained using images $\mathbf{y}_R$ and $\mathbf{y}_C$.

2.1 Binocular depth estimation

Binocular depth estimation is illustrated in Figure 1 (right side). The approach makes use of local matching algorithm with *color adaptive weights* [15] . For each stereo-pair (central-left or central-right), two cost volumes are constructed, as follows:

$$Cost_L(\mathbf{x},d) = \left| \mathbf{y}_L(x_1,x_2) - \mathbf{y}_R(x_1-d,x_2) \right|_1;\quad Cost_R(\mathbf{x},d) = \left| \mathbf{y}_R(x_1,x_2) - \mathbf{y}_L(x_1-d,x_2) \right|. \tag{1}$$

Each cost volume is then aggregated

$$CostAggr_P(\mathbf{x},d) = F\{Cost_P(\mathbf{x},d)\},\ d = 1..\max(disparity) \tag{2}$$

The aggregation operator essentially filters the cost slices for each d with weights determined by spatial locality and color similarity

$$F\left(Cost_P(\mathbf{x},d)\right) = h(\mathbf{x}) \sum_{\mathbf{u}\in\Theta_\mathbf{x}} w_s\left(\|\mathbf{x}-\mathbf{u}\|\right) w_c\left(\left|\mathbf{y}_P(\mathbf{x}) - \mathbf{y}_P(\mathbf{u})\right|\right) Cost_P(\mathbf{x},d), \tag{3}$$

where $w_a(t) = e^{-\frac{t}{\gamma_a}}$, $a = s, c$; $h(\mathbf{x}) = \left[\sum_{\mathbf{u}\in\Theta_\mathbf{x}} w_s\left(\|\mathbf{x}-\mathbf{u}\|\right) w_c\left(\left|\mathbf{y}_P(\mathbf{x}) - \mathbf{y}_P(\mathbf{u})\right|\right) Cost_P(\mathbf{x},d) \right]^{-1}$, $\Theta_\mathbf{x}$ is a square window around $\mathbf{x}$, γ_c and γ_s are adjustable parameters of the matching [15].

At the next step, the cost volumes are used to estimate the respective disparity levels by *winner-takes-all*

$$D_{L/R}(\mathbf{x}) = \arg\min_d Cost_{L/R}(\mathbf{x},d). \tag{4}$$

Along with disparity, our implementation allows obtaining a confidence measure, which can be used for further post-processing, prediction or fusion. The confidence measure is based on Peak Ratio measure, i.e. the distance between the winner minimum and the second strongest local minimum normalized by the former [16]. Left-to-Right correspondence check [16] is applied then to mark occlusions and outliers for the central disparity images

$$Occl_L(x_1,x_2) = \Gamma\{|D_L(x_1,x_2) - D_R(x_1 - D_L(x_1,x_2),x_2)| - \xi\};$$
$$Occl_R(x_1,x_2) = \Gamma\{|D_R(x_1,x_2) - D_L(x_1 + D_R(x_1,x_2),x_2)| - \xi\}, \quad (5)$$
$$\text{where } \Gamma(t) = \begin{cases} 1 & \text{for } t > 0 \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Trinocular depth estimation

In our implementation, the trinocular setting simply extends the binocular one. While this might be not optimal compared with e.g. a trifocal tensor matching approach, it explicitly characterizes the effect of adding a third camera for handling occlusions and on the overall quality of the estimated depth. Binocular disparity estimation is performed also for the left and central camera pair. Thus, for the central camera there are two disparity estimates which can be fused.

For fusing two disparity estimates, the confidence measure is used and the estimate with higher confidence is selected. As the Peak-Ratio measure is scaled from zero to one, is somehow represents the probability of the valid estimate. Selecting the estimate with higher confidence is equivalent to employing Bayesian classification. In this approach, occluded pixels have to be marked with zero confidence prior to fusion.

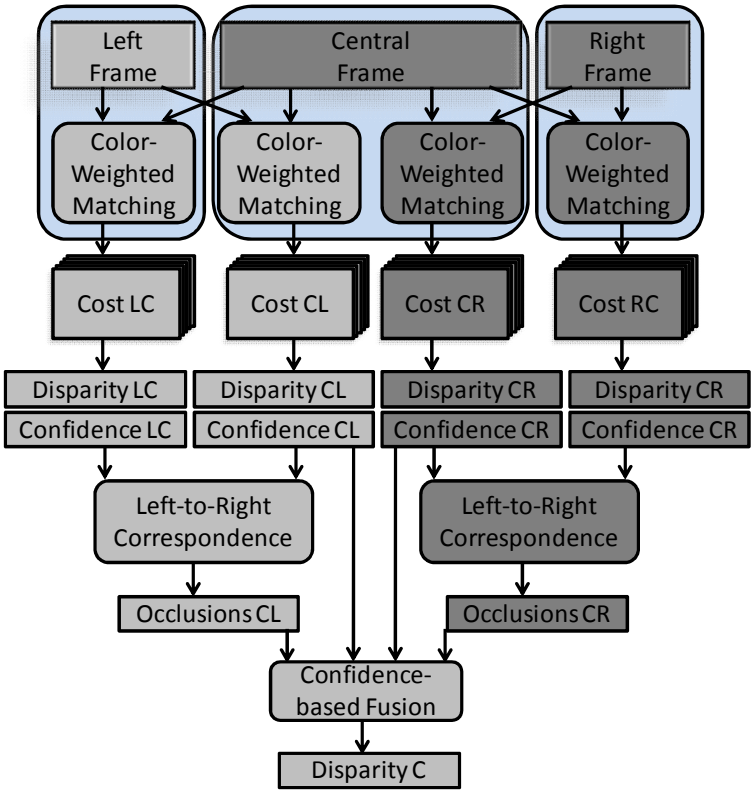

Figure 1. Scheme of the matching approach

3. EXPERIMENTAL RESULTS

3.1 Data sets

Two groups of test datasets have been used in the experiments. The first group contains multi-view frames accompanied with known (ground true) depth maps. Within this group, the 'Dancer' dataset, as provided by Nokia Research Center, comprises twenty views rendered from a photorealistic virtual scene in HD resolution (1920x1080 pixels) from

equidistant virtual cameras. The ground true depth associated with each view is available as well. The Art', 'Cones' and 'Mobius' datasets are part of the Middlebury framework. True depth maps have been calculated using structured light approach [17] [18]. The datasets are illustrated in Figure 2.

Figure 2. Thumbnails of test datasets with ground truth depth maps: 'Dancer', 'Art','Cones','Mobius'.

The second group of datasets contains real-world multi-view images from linearly rectified cameras with no available ground true depths. This includes the 'Dog' dataset courtesy of Tanimoto Lab at Nagoya University [19] and the 'Book Arrival' dataset courtesy of Fraunhofer HHI [20]. Thumbnails of these datasets are given in Figure 3.

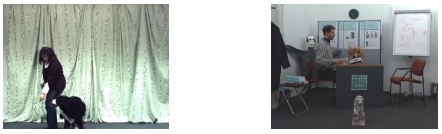

Figure 3. Multi-view datasets: 'Dog', 'Book Arrival'.

3.2 Quality measures

A widely used quality measure in depth estimation applications is the *'number of bad pixels'* [21]:

$$BAD = \frac{100}{\mathrm{N}} \sum_{x \in \Xi} \left(\left| D(\mathbf{x}) - \hat{D}(\mathbf{x}) \right| > \delta_d \right), \tag{6}$$

where $D(\mathbf{x})$ is the true disparity image at pixel with coordinates $\mathbf{x}$, $\hat{D}(\mathbf{x})$ is estimated disparity, δ_d is a pres-specified threshold, (usually set to 1), N is the number of pixels in the region of interest Ξ. We have calculated BAD for the whole image as well as for areas near object borders (*BAD near discontinuities*). To mark these areas, we have run a Canny edge detector over the ground true depth map and dilated the marked edges applying a circle kernel of radius of 5 pixels. To characterize the behavior of depth estimation approaches around edges, we have also used a metric called *Depth Consistency*. It measures the percentage of pixels, having magnitude of the gradient of the difference between true depth and processed depth $\nabla\xi = \nabla(D - \hat{D})$ higher than a pre-specified threshold

$$CONSIST = \frac{100}{N} \sum_{x \in \Xi} \left(\left\| \nabla \xi \right\| > \delta_{consist} \right). \tag{7}$$

The measure gives preference to non-smooth areas in the estimated depth considered as main source of geometrical distortions.

PSNR of estimated disparity image and *PSNR of rendered image* have been used as well. In addition to the latter, we have calculated also *Gradient-normalized RMSE* over the luminance channel of rendered image. This is a room-mean squared error, normalized by the gradient, thus penalizing local intensity variations in textured areas [22]:

$$NRMSE_\eta = \sqrt{\sum_{x \in \Xi} \frac{(y^Y(\mathbf{x}) - \hat{y}^Y(\mathbf{x}))^2}{\left\| \nabla y^Y(\mathbf{x}) \right\|^2 + 1}}, \tag{8}$$

where $y^Y(\mathbf{x})$ is a grayscale image rendered from a true depth map and $\hat{y}^Y(\mathbf{x})$ is an image rendered from the estimated depth. Occluded pixel values are excluded from the estimate [23].

Execution time has been normalized with respect of the dataset complexity. That is, the processing time for each image is divided by the image size and disparity range. The obtained numbers are them multiplied by 10^6 so that one unit of so *normalized time* corresponds to 'time spent by the algorithm to process 1 Megapixel image with 1 disparity level'.

3.3 Experimental setup

The size of matching (aggregation) square window has been considered as the main adjustable parameter to change performance and has been varied within certain range. The adjustable parameters γ_a in the color weighting scheme have been fixed to 20. The size of square window used in the post-processing procedure has been fixed to 11x11 pixels. The quality metrics have been then calculated against the aggregation window size. Equally well, they can be also put against the execution time.

The rendered channels have been obtained as follows: first, two triplets of cameras have been selected from the datasets. Depth estimates for the central camera of each triplet have been obtained using either binocular or trinocular setting. Using the obtained depth estimates and the associated images, new virtual view between them has been synthesized at the place of a given view of the dataset to be compared with. This approach decreases the occlusion artifacts in the synthesized view and allows for fairer comparison of the effect of different depth estimation approaches.

4. RESULTS

We first illustrate the dependence between the size of local processing window and the execution time. Figure 4 shows execution time vs window size for 'Dancer' and 'Art' data sets. For the rest of datasets, the dependence is quite the same. There is a quadratic dependence between processing time and window size for local methods. The difference in execution time between binocular and trinocular methods grows with the window size but both times are acceptably close. The time consumed for post-filtering is quite negligible, compared with the time for depth estimation.

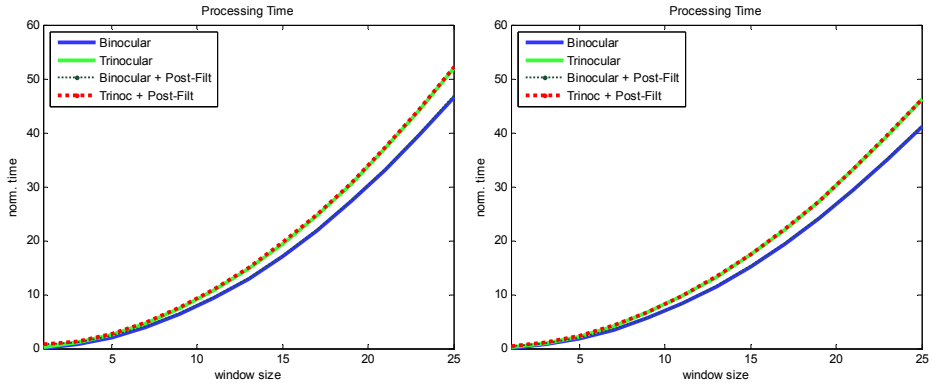

Figure 4. Processing time vs window size. Left – 'Dancer'; right - 'Art' dataset.

Figure 5 to Figure 10 show results for datasets with available ground true dept maps. Along with tested local methods, the MPEG's DERS results are shown with horizontal lines as the window size is not applicable to that reference software. The superiority of trinocular approach combined with post-processing is clearly seen in all metrics. It gets very nicely saturated after some, relatively small, window size. The effect of post-filtering is best expressed by the number of bad pixels near discontinuity, where the filtering procedure manages to effectively decrease the errors encountered at small matching window sizes. In the case of no post-processing stage involved, the trinocular setting shows considerable advantage compared to the binocular one. Its higher performance is achieved at smaller window sized and is much more pronounced than the computational overhead required for processing the data from the third camera. The local methods are quite competitive to the global one and for some cases better for relatively small matching windows.

The performance measured through the PSNR of the rendered channel has to be especially analyzed. According to this measure, the binocular approach with no post-filtering has some advantage with respect to trinocular only for very small windows. This is an effect caused by over-penalization of textured areas (best visible in the 'Art' sequence), which is compensated when using NRMSE instead. Post-filtered results demonstrate the potential for further improvement in both performance and computational time.

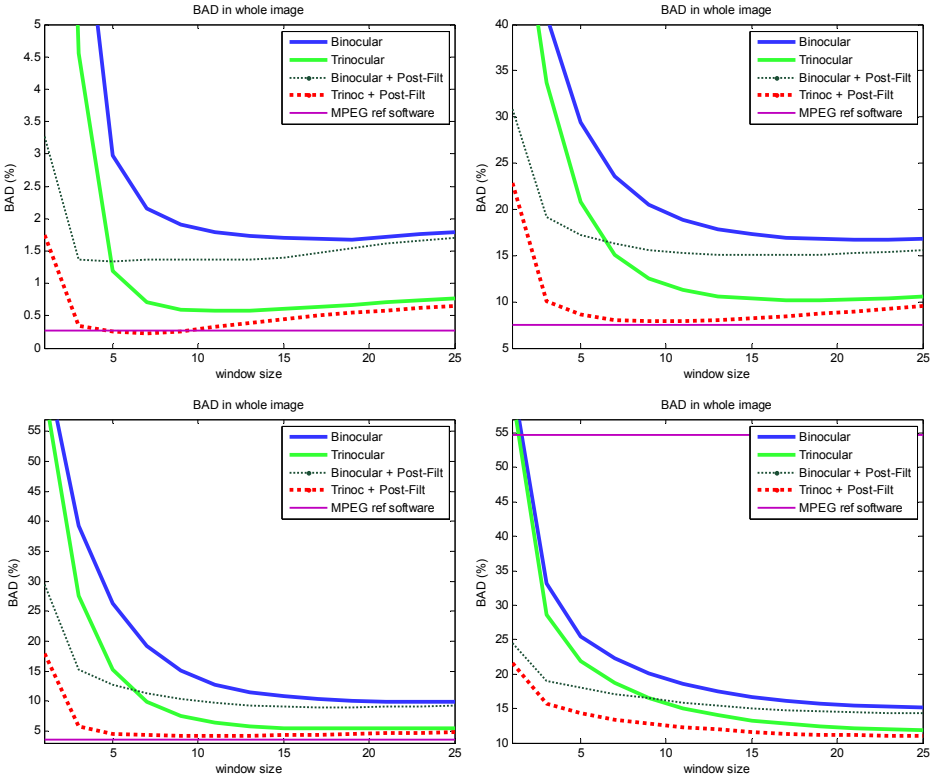

Figure 5. Percentage of bad pixels for the whole image. Datasets from top clockwise: 'Dancer', 'Art', 'Moebius', 'Cones'.

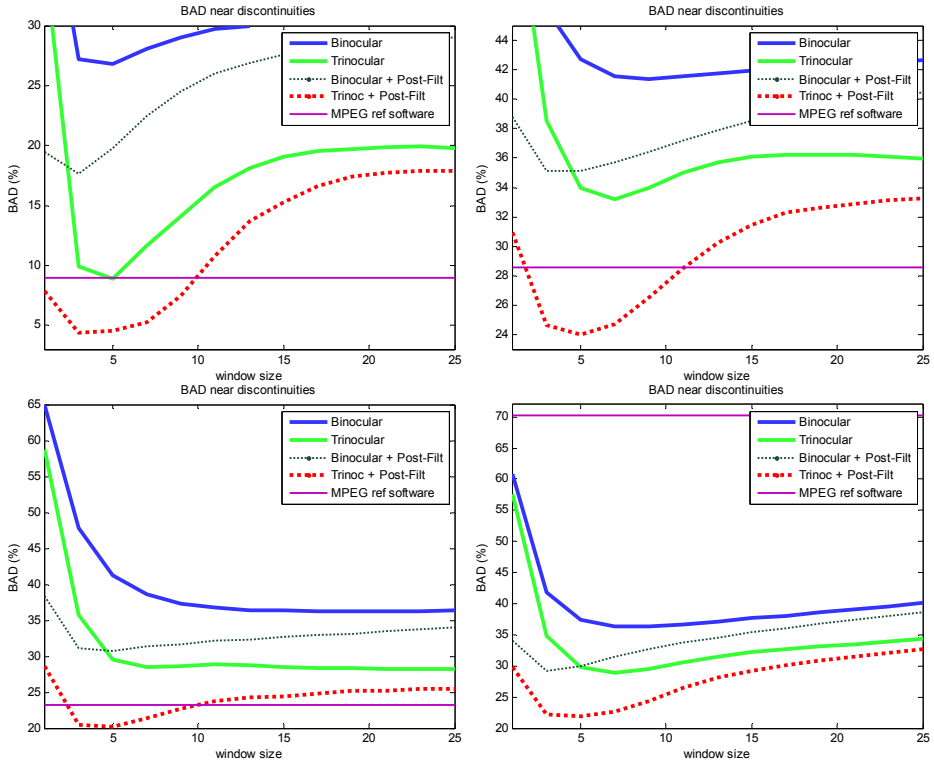

Figure 6. Percentage of bad pixels near discontinuities for the first data set. Datasets as in Figure 5.

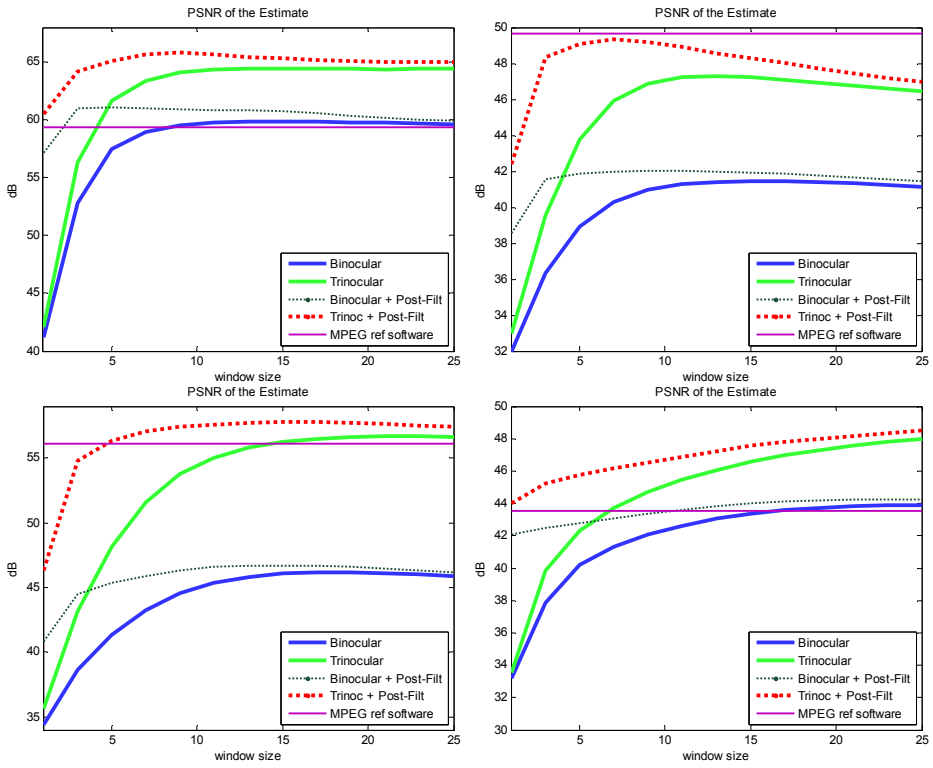

Figure 7. PSNR of the estimated depth. Datasets as in Figure 5

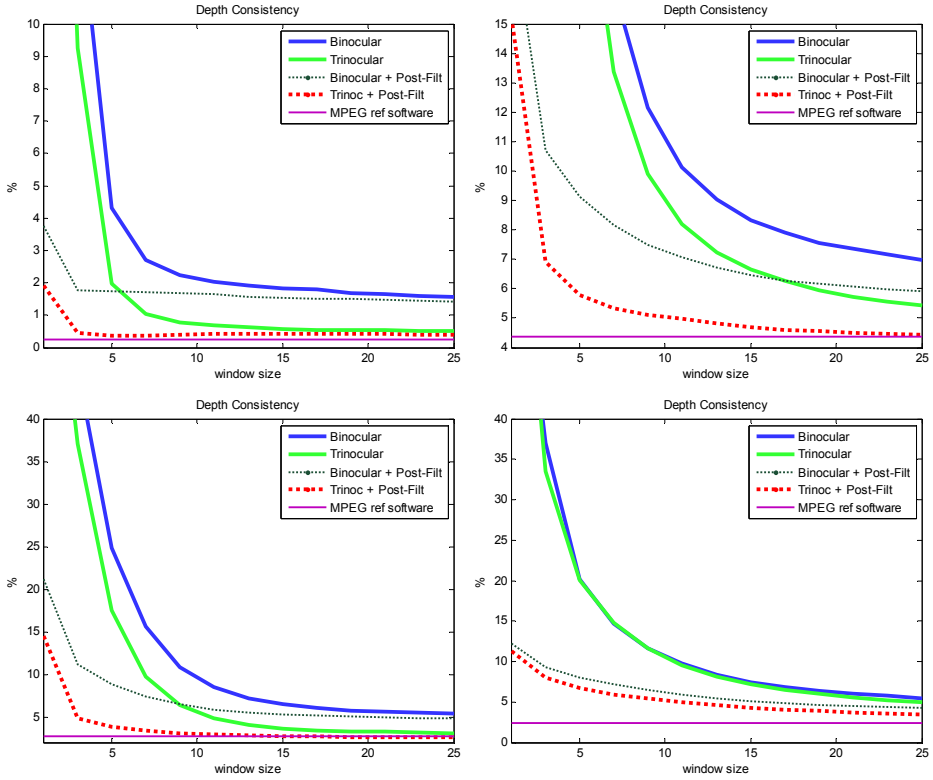

Figure 8. Depth consistency of estimated depth. Datasets as in Figure 5

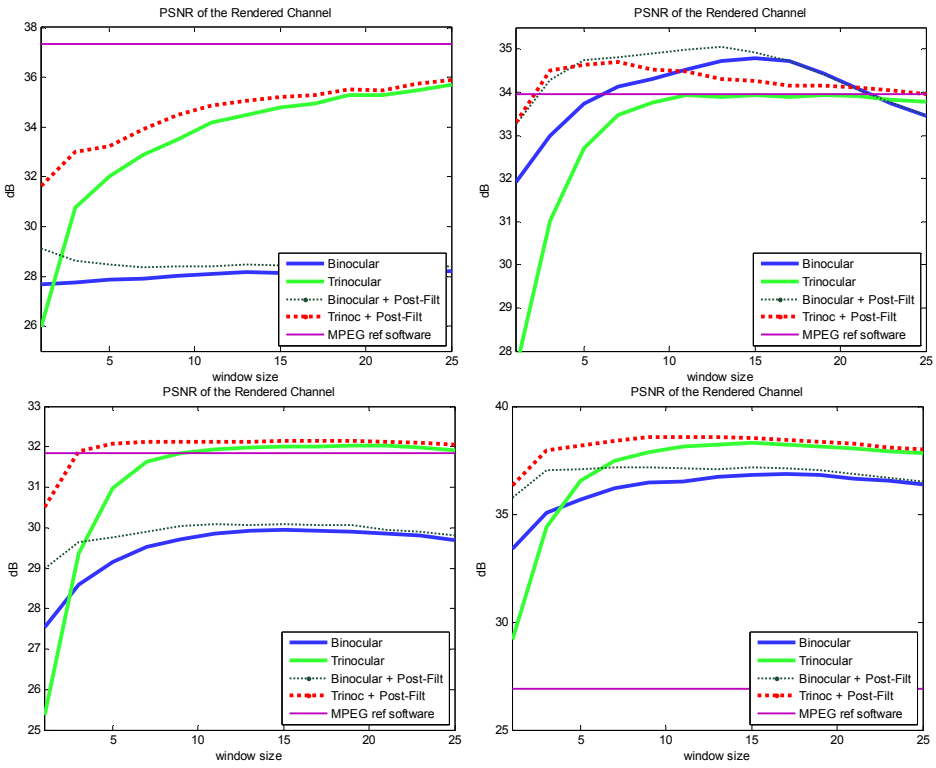

Figure 9. PSNR of rendered right channel. Datasets as in Figure 5.

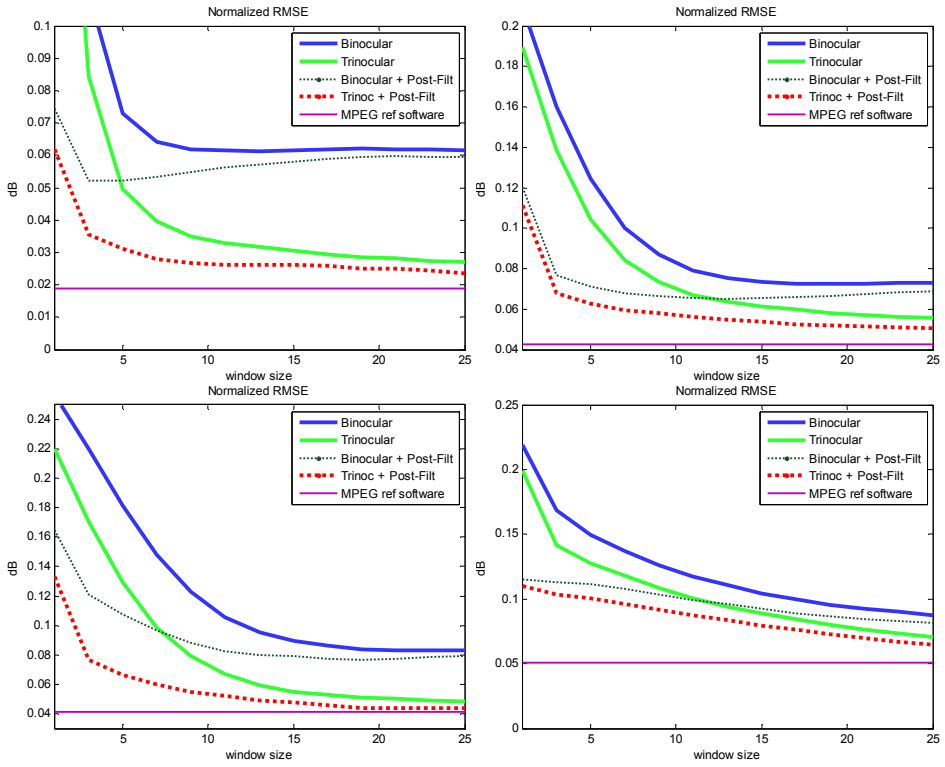

Figure 10. NRMSE of rendered right channel. Datasets as in Figure 5.

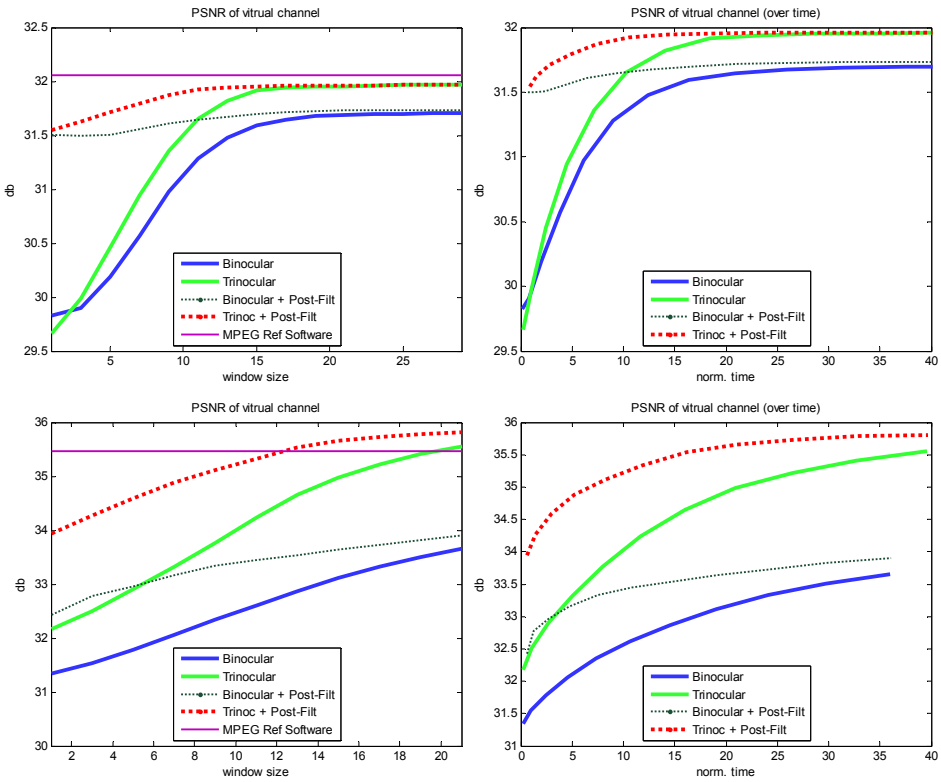

Figure 11. PSNR of virtual view and same alongside execution time. First row: 'Dog' dataset; second row: 'Book Arrival' dataset

Figure 11 compares the methods for datasets with no ground true depth maps available. For these, only indirect comparison, through the quality of the rendered virtual channel, is possible. Figure 11 shows the PSNR of the rendered channel against window size and execution time. The second column of Figure 11 practically says that there is no execution time, even very short, for which the binocular setting is better.

Figure 12 and Figure 13 illustrate the quality of obtained depth maps for the test sequences 'Art' and 'Dancer'.

5. CONCLUSIONS

Except better occlusion handling, trinocular approach is generally smoother compared with binocular approach for the same window size. This is due to the reduced ammount of non-occluded pixels in the trinocular setting. This makes possible to reduce widnow size and achieve the same quality as of binocular setting with wider window. Window size reduction dramatically reduces the computational time for trinocular setting, thus making it more preferable when targetting faster processing.

We have shown that additional camera improves quality of restoration already for small window size. For specific time-constrained applications third camera also may add significant improve to the reconstruction if the time constraint is large enough. Nearly real-time applications (with CPU implementations) may benefit from third camera by reduction of window size, required for matching and avoiding costly occlusion filling operations.

Our results were shown to be comparable with results of MPEG Depth Estimation Reference Software, based on the Graph Cuts global optimization approach. We have chosen local approach as it gives fairer comparison of binocular and trinocular settings. Furthermore, based on initial local estimates, further non-local optimizations can be successfully applied. In a sense, the desired smoothness of the solution, which is usually targeted by non-local optimizations, can be achieved by a local processing with suitably chosen, wide enough window. It is an interesting research question to study how small local window is sufficient to provide a reliable initial depth estimate to be further refined by post-processing, possibly run off-line

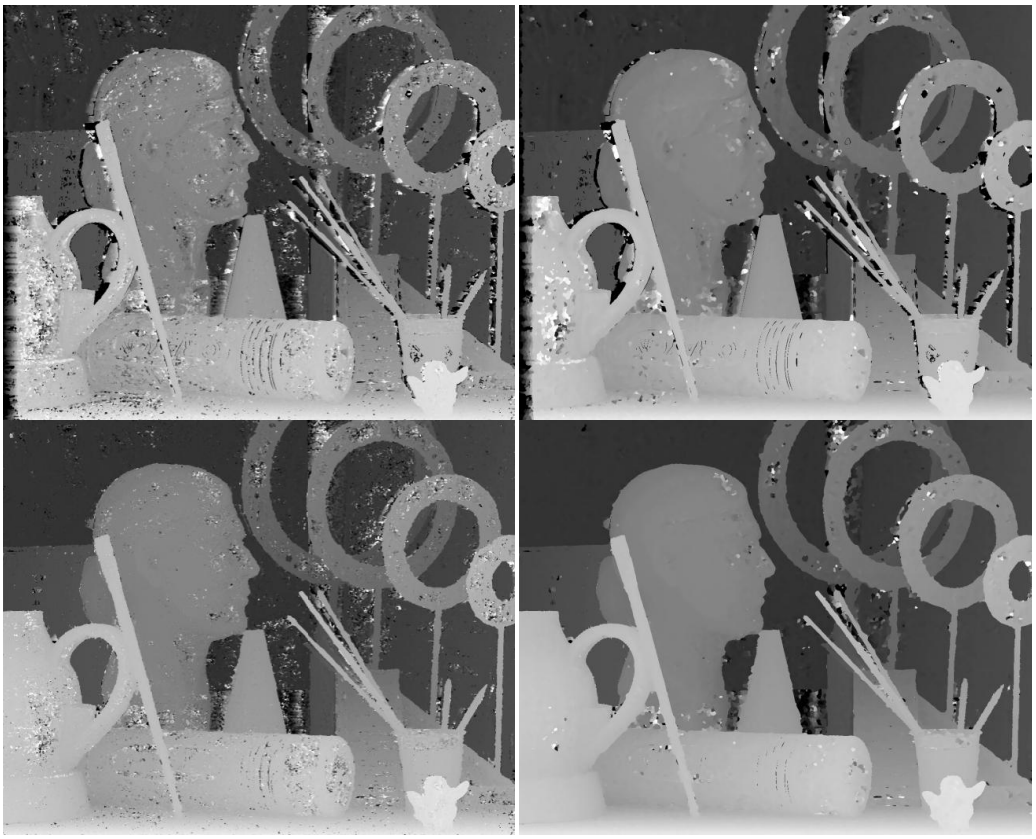

Figure 12. Depth estimation results for ‘Art’ dataset. From top clockwise: binocular; binocular with post-processing; trinocular with post-processing; trinocular.

Figure 13. Depth estimation results for ‘Dancer’ dataset. From top clockwise: binocular; binocular with post-processing; trinocular with post-processing; trinocular.

6. REFERENCES

[1] Klaus, A., Sormann, M., and Karner, K., "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," International Conf. on Pattern Recognition, Hong Kong, 15–18 (2006).

[2] Wang, Z. and Zheng, Z., "A region based stereo matching algorithm using cooperative optimization," IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, Alaska (2008).

[3] Yang, Q., Wang, L., Yang, R., Stewénius, H., and Nistér, D., "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," IEEE Trans. on Pattern Analysis and Machine Intelligence, 31(3) (March 2009).

[4] Tombari, F., Mattoccia, S., Stefano, L. Di, and Addimanda, E., "Near real-time stereo based on effective cost aggregation," Int. Conf. on Pattern Recognition, Tampa, USA (2008).

[5] Yang, Q., Engels, Ch., and Akbarzadeh, A., "Near Real-time Stereo for Weakly-Textured Scenes," British Machine Vision Conference (2008).

[6] Wang, L., Liao, M., Gong, M., Yang, R., and Nister, D., "High Quality Real-time Stereo using Adaptive Cost Aggregation and Dynamic Programming," Third Int. Symposium on 3D Data Processing, Visualization and Transmission (2006).

[7] Yang, R. and Pollefeys, M., "Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware," Computer Vision and Pattern Recognition , Madison, Wisconsin (2003).

[8] Collins, R. T., "A Space-Sweep Approach to True Multi-Image Matching," IEEE Conf. on Computer Vision and Pattern Recognition, San Francisco, 358 (1996).

[9] Heinrichs, M. and Rodehort, V., "Trinocular Rectification for Various Camera Setups," Symp. of ISPRS Commission III - Photogrammetric Computer Vision, Bonn, Germany, 43-48 (2006).

[10] Kangni, F. and Laganiere, R., "Projective rectification of image triplets from the fundamental matrix," IEEE Int. Conf. Acoustics Speech Signal Processing (August 2006).

[11] Shashua, A. and Werman., M., "On the trilinear tensor of three perspective views and its underlying geomtry," Int. Conf. on Computer Vision, Boston, 920-925 (1995).

[12] Stankiewicz, O. and Wegner, K., "Depth Map Estimation Software version 3," MPEG/M15540, ISO/IEC JTC1/SC29/WG11, Hannover, Germany, 2008.

[13] Qingxiong, Y., Ruigang, Y., Davis, J., and Nister, D., "Spatial-Depth Super Resolution for Range Images," IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, Minnesota (2007).

[14] Smirnov, S., Gotchev, A., and Egiazarian, K., "A Memory-efficient and Time-consistent Filtering of Depth Map Sequences," Image Processing: Algorithms and Systems VIII (part of Electronic Imaging Symposium), San Jose, USA (2010).

[15] Yoon, K.-J. and Kweon, I.-S., "Locally Adaptive Support-Weight Approach for Visual Correspondence Search," Conf. on Computer Vision and Pattern Recognition, 924 - 931 (2005).

[16] Egnal, G., Mintz, M., and Wildes, R. P., "A stereo confidence metric using single view imagery with comparison to five alternative approaches," Image and Vision Computing, 22(12), 943-957 (October 2004).

[17] Szeliski, D. and Scharstein, R., "High-accuracy stereo depth maps using structured light," Computer Vision and Pattern Recognition, Madison (2003).

[18] Scharstein, D. and Pal, C., "Learning conditional random fields for stereo," IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis (2007).

[19] Tanimoto, M., Fujii, T., Senoh, T., Aoki, T., and Sugihara, Yo., "Test Sequences with Different Camera Arrangements for Call for Proposals on Multiview Video Coding". ISO/IEC JTC1/SC29/WG11, Poznan, Poland, 2005.

[20] Feldmann, I. I., Kauff, P., Mueller, K., Mueller, M., Smolic, A., Tanger, R., Wiegand, T., and Zilly, F., "HHI Test

Material for 3DVideo," MPEG2008/M15413, Archamps, France, April 2008.

[21] Scharstein, D. and Szeliski, R., "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.," International Journal of Computer Vision, 47, 7-42 (April-June 2002).

[22] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R., "A database and evaluation methodology for optical flow," Proc. IEEE Int'l Conf. on Computer Vision, Crete, Greece, 243–246 (2007).

[23] Smirnov, S., Gotchev, A., and Egiazarian, K., "Methods for restoration of compressed depth maps: a comparative study," Int. Workshop on Video Processing and Quality Metrics for Consumer Electronics (2009).