2010 11th. Int. Conf. Control, Automation, Robotics and Vision
Singapore, 5-8th December 2010

A 3D Reconstruction System Based on Improved Spacetime Stereo

Federico Tombari[1] Luigi Di Stefano[1] Stefano Mattoccia[1] Andrea Mainetti[2]
DEIS - ARCES
University of Bologna
Bologna, Italy
[1] {name.surname}@unibo.it [2] andrea.mainetti@studio.unibo.it

***Abstract*—Spacetime stereo is a promising technique for accurate 3D reconstruction based on randomly varying illumination and temporal integration of the stereo matching cost. In this paper we show that the standard spacetime stereo approach can be improved in terms of accuracy of disparity estimation and convergence speed by adoption of suitable matching algorithms based on adaptive support windows. We also present a practical and cost-effective 3D reconstruction system that deploys the proposed improved spacetime method together with cheap commercial off-the-shelf hardware (a PC, a stereo camera and a projector). Experimental results show that the proposed system can yield rapidly accurate 3D reconstruction of various types of objects and faces.**

***Index Terms*—Spacetime stereo adaptive support 3D reconstruction**

I. Introduction

Accurate 3D reconstruction is crucial in many applications such as reverse engineering and rapid prototyping, 3D object recognition, 3D biometrics and forensics, robot vision. A typical solution to obtain 3D data from the observed scene consists in the use of laser scanners such as those based on time-of-flight (TOF) sensors (e.g. flash ladars [1]) or on optical triangulation (e.g. line stripe systems [2]). The main disadvantages of the former approach are represented by low resolution and noisy range data, while for the latter they generally consist in high costs and dense depth measurements constrained to static scenes. Another solution is given by structured light systems, which often rely on expensive high frame-rate projectors and cameras.

Recently, a technique based on stereo vision and pattern projection, referred to as *spacetime stereo*, has been proposed [3], [4]. This technique holds the potential to yield accurate 3D reconstruction of static objects based on the analysis of a stereo sequence where a variable random pattern is projected on the scene by means of an off-the-shelf light projector. A major difference with respect to other projection-based approaches is that the projected light is *unstructured*, i.e. the image analysis module does not know the nature of the pattern being projected, which is used only to augment the scene with texture. This is particularly attractive in a robotic scenario where more than one robot might attempt to reconstruct the structure of the same object. In fact, robots would not interfere each other as instead would be the case of conventional structured light systems. Finally, it is worth pointing out that some attempts have been made to extend spacetime to dynamic scenes but all proposed approaches suffer from major limitations [4]

The contribution of this paper is twofold. First, we investigate on improving the standard spacetime approach by deploying more advanced stereo matching algorithms. In particular, we analyze state-of-the-art global and local matching algorithms and identify in variable support approaches based on shifted windows the most promising approach for spacetime. Experimental evaluation indicates that the identified matching algorithms can yield better accuracy and faster convergence with respect to the standard spacetime approach. The latter advantage might allow for an effective extension of spacetime to slowly moving objects/camera. As for the second contribution, we describe a practical and cost-effective PC-based 3D reconstruction system that relies on the proposed improved spacetime framework. In this context, we highlight the important filtering stages (i.e. fast consistency check, background subtraction and depth filtering, bilateral filtering) that are required to come up with accurate 3D reconstruction from spacetime disparity estimations and provide several result concerning reconstructions of objects and faces.

II. Spacetime stereo using a variable support

A. Spacetime stereo

Given two rectified views, stereo matching aims at finding corresponding points in the two views, or, equivalently, at computing the disparities associated with each point of the view chosen as *reference*. Generally speaking, correspondences are found based on the similarity between pixel intensities or colors. Since relying only on individual pixel intensities/colors to compute similarities is not sufficiently robust, the standard, or block-based (BB), stereo algorithm [5] deploys a spatially extended support (typically a squared window of pixels) to reinforce disparity estimation, with larger supports yielding higher robustness. Yet, the increase in robustness is counterparted by a decrease of reconstruction accuracy along object borders, where the assumption of constant depth within the window underlying BB stereo fails. Moreover, it is well known that the BB approach also fails at occluded regions and within low-textured areas.

The rationale behind spacetime stereo is that, instead of extending the support spatially, to robustly compute disparity at a given point one can rely on a local set of temporally

978-1-4244-7815-6/10/$26.00 ©2010 IEEE

correlated pixels. Analogously to the spatial case, this requires that the depth of the point within this set is constant over time, that is, the scene must be static. Under this assumption, a matching function C_{sts} is computed by evaluating a pointwise similarity function S over a spatio-temporal support including all pixels that fall within a squared window $W_r(p)$ of radius r centered at a given point p over a certain number of frames F [4], [3]:

$$C_{sts}(x,y,d) = \sum_{t \in F} C(x,y,d,t) \tag{1}$$

where

$$C(x,y,d,t) = \sum_{(i,j) \in W_r(p)} S\left(I_L(i,j,t), I_R(i-d,j,t)\right). \tag{2}$$

with I_L, I_R denoting the left and right stereo sequences, (x,y) being p's coordinates and d being the disparity currently evaluated. Once C_{sts} is computed over all disparities for the given point p, the disparity $\hat{d}$ associated with p is selected as the one minimizing C_{sts} (*Winner-Take-All* approach):

$$\hat{d} = \operatorname*{argmin}_{d} C_{sts}(x,y,d) \tag{3}$$

It is worth pointing out how in its standard formulation spacetime stereo basically consists in the application of the standard BB stereo algorithm on a spatio-temporal - instead of only spatial - support. However, while the spatial support aggregates local information by including pixels around p, if the scene appearance does not change no additional information is brought in by the use of a temporal support. For this reason, in [4], [3] it is proposed to vary the illumination of the scene: in particular, a light projector can be deployed to augment the scene with a random pattern that varies at each frame of the sequence.

As for the pattern to be projected on the scene, we have performed a comparative study between different pattern typologies (some are shown in figure 1). The experiments have been carried out by projecting each pattern over some simple scenes and computing the RMSE error between the achieved disparity maps and the hand-labeled groundtruth. Our results indicate that the best performing pattern is of the same kind of those used in [4], [3], i.e. a sequence of vertical binary stripes of randomly varying thickness.

The temporal extension in spacetime stereo allows for using small spatial support, with the salutary effect of reducing wrong disparity estimations along object borders. Furthermore, the use of the projected pattern notably reduces the presence of low-textured areas, which, as previously mentioned, are a major source of errors for the standard BB algorithm. The combination of small supports and enhanced texture results typically in accurate disparity estimation.

B. Stereo matching approaches

Stereo matching is an intensively studied research topic, with an impressive number of algorithms capable of significantly improving disparity estimation with respect to standard BB approach proposed in literature in the past few years.

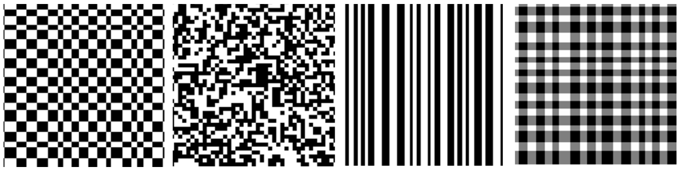


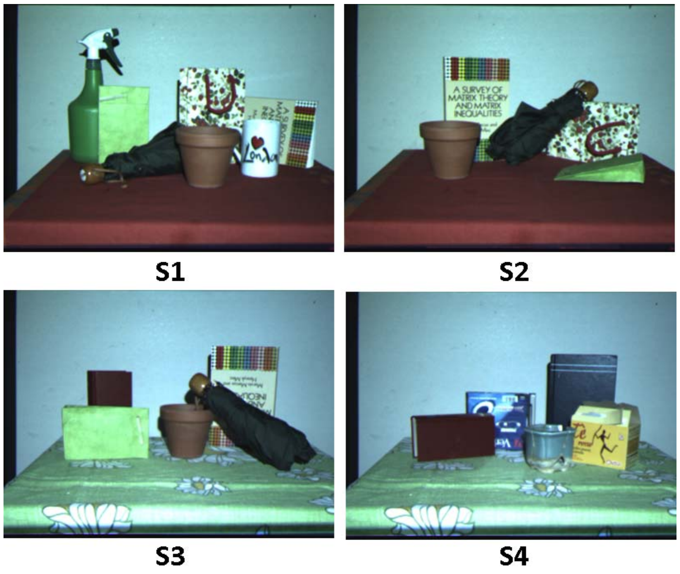
Fig. 1. Some evaluated patterns.

As reported in [4], almost any stereo matching method could be extended to the time domain within the spacetime stereo framework. Based on this observation, we have investigated on advanced stereo matching methods that could be effectively deployed within a spacetime stereo framework.

Current approaches can be divided between local and global [5]. As for local ones, the class of methods based on a *variable support* [6] represents the state of the art for what concerns accuracy. The main idea is to use a support that dynamically adapts its shape to the local spatial characteristics of the image rather than using a fixed window. Proposed methods select supporting points based on the similarity in intensities (or colors) between the reference point and its neighbours (e.g. [7]) or on the minimization of a similarity cost computed on a window over different neighbouring position (e.g. [8], [9], [10]). On the other side, global approaches aim at finding an approximate solution for a minimization problem based on an energy function computed globally on all image points, with such a function generally including a pointwise matching term and a smoothness term. Currently the best performing optimization methods are based on Belief Propagation (BP), Graph Cuts (GC), Dynamic Programming (DP), Scanline Optimization (SO) [5].

The use of a global method deploying a smoothness term allows for propagating depth information to those regions where the matching cost term is unable to reliably detect the correct disparity, i.e. especially on low-textured or even uniform areas. As a matter of fact, the use of a projected pattern typically greatly reduce the presence of low-textured areas in the scene to be reconstructed, thus not motivating the use of this class of methods for the spacetime stereo framework. In addition, the majority of these methods (e.g. those based on BP and GC) are particularly time-consuming. As for local methods, those relying on similarity of intensities/colors for the determination of the spatially adaptive support would simply be unfeasible in a spacetime context given the presence of the projector that alters the appearance of the scene. In fact, these methods are based on the assumption that depth edges coincide with intensity edges, while in a spacetime stereo context the projected random pattern creates many new edges that are totally uncorrelated with the depth borders present in the scene.

Based on these considerations, the class of stereo matching methods that appear most suitable to be applied in a spacetime stereo context is that relying on the minimization of a similarity cost computed on a window over different neighbouring positions [8], [9], [10]. This class of methods typically deploys rectangular windows shifted over different



S1

S2

S3

S4

Fig. 2. The 4 sequences used in the experiments.

positions in a local neighbourhood to select the best horizontal and vertical displacement that maximizes the number of pixels in the window lying at the same depth. As pointed out in Section II, in the spacetime stereo framework the preferred pattern consists of vertical stripes (i.e. characterized by vertical uniform areas). Hence, though in general it would be useful to account also for vertical shifts, variations in the texture of the scene tend to be along horizontal lines. Thus, we propose to evaluate the best displacement only along horizontal shifts, which allows a limited computational overhead with respect to the BB algorithm. In particular, we take into consideration three different variable support algorithms:

- *Shiftable Windows* (SW): similarly to the proposal in [10], the cost is the minimum obtainable by shifting a square window over $2r + 1$ horizontal shifts centered in p:

$$C_{SW}(x, y, d, t) = \min_{k \in [x-r, x+r]} C(k, y, d, t) \qquad (4)$$

- *Three Windows* (3W): the cost is the minimum obtainable by shifting a square window over 3 horizontal shifts centered in p:

$$C_{3W}(x, y, d, t) = \min_{k \in \{x-r, x, x+r\}} C(k, y, d, t) \qquad (5)$$

  This approach can be regarded as a simplified version of method SW, and it is similar to the methods proposed e.g. in [11], [12].

- *Multiple Windows* (MW): inspired by [8], the cost is computed as that of a square window centered on p plus the minimum obtainable by shifting a square window over two additional 2 horizontal shifts:

$$\begin{aligned} C_{MW}(x, y, d, t) = {} & C(x, y, d, t) + \\ & + \min_{k \in \{x-r, x+r\}} C(k, y, d, t) \end{aligned} \qquad (6)$$

Once one of these costs (depending on the chosen algorithm) is computed at every position (x, y), disparity d and frame t, it is substituted to term C in equation (1) to yield the final C_{sts} cost.

C. Experimental evaluation

This section proposes an experimental analysis aimed at evaluating whether the use of variable support matching algorithms within the spacetime stereo framework can improve disparity estimation with respect to the standard formulation. In particular, we compare the use of the 3 algorithms identified in Section II (i.e. SW, 3W, MW), with respect to that of standard BB matching on 4 stereo sequences referred to as $S1$, $S2$, $S3$, $S4$. Each sequence corresponds to a different scene composed by several objects at different depths, as shown in figure 2. For all experiments, the pointwise matching function S is the absolute difference between pixel intensities. As for the size of the spatial windows, for each algorithm and each sequence, three different values of the window radius r are considered: 3, 5, 7.

In order to perform a quantitative evaluation and in absence of a method to easily obtain groundtruth information from the rather complex scenes used for the experimental evaluation, we compute the percentage of points in the resulting disparity map that are discarded by means of a *left-right consistency check* [13]. As pointed out in [13], this indicator can be regarded as a measure of the quality of the retrieved disparity map. Obviously, the number of discarded points does not depend only on the number of matching errors, but also on the number of occluded pixels, that can not be handled by the standard spacetime stereo approach. This must be taken into consideration when analysing the reported values of the indicator.

The graphs, in figures 3, 4 show for each algorithm the percentage of matching errors detected by the left-right consistency check as a function of the frame number along the sequence. Each row in the two figures refers to a different radius, while each column refers to a different sequence. As it can be seen from the graphs, the BB algorithm is less accurate than the 3 considered variable support algorithms on all the sequences and independently of the radius size. It is worth pointing out here that, since the values of matching errors include also occluded pixels, which account for the same quantity for all methods, the actual discrepancies in accuracy rates between the compared methods are indeed greater than those reported in the graphs. Moreover, the BB algorithm yields the highest percentage of errors on the long run (i.e. after processing several frame rate, when the error percentage stabilizes on a particular value), but also the slowest to converge to a stable error value. This means, to achieve a desired level of accuracy one has to process a significantly higher number of frames with BB matching than with the considered variable support algorithms. For instance, the graph concerning sequence $S1$ and radius $r = 5$ shows that to get to an error value below 12% we need to process 66 frames with the BB algorithm, while only 8 frames are sufficient with the MW algorithm. As for

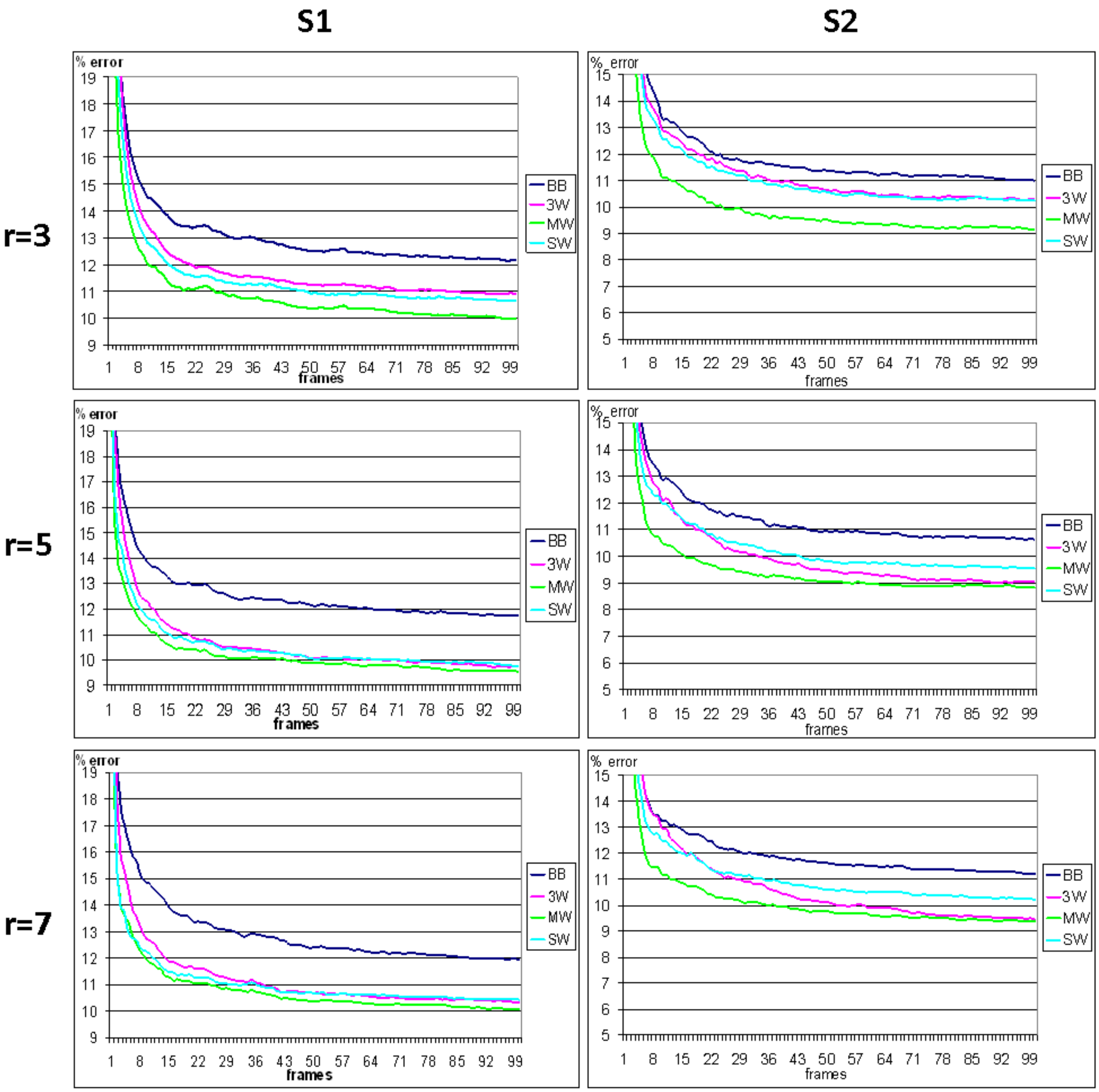


Fig. 3. Experimental comparison of different aggregation strategies on sequences $S1$, $S2$.

radius size, the lowest error rates are generally reported by the algorithms with size 5. As for comparison between MW, SW and 3W, results show that, when many frames are available, the two best performing algorithms, MW and 3W, have similar performance, with MW performing overall slightly better with radius $r = 5, 7$ and notably better with radius $r = 3$. Moreover, when only few frames are processed, MW outperforms the other approaches in all sequences and with all the considered radius sizes. Hence, MW will be the algorithm used in our 3D reconstruction framework, which will be described in the next Section. Computationally, it requires roughly a $25 \sim 30\%$ overhead with respect to the BB algorithm.

Eventually, to highlight the advantages of the multiple-window approach, in Fig. 5 we qualitatively compare the disparity maps yielded by BB and MW on dataset S1 with $r = 5$. As expected, MW yields thinner regions of matching errors (blue pixels) at depth discontinuities (e.g. red boxes) as well as within visible areas adjacent to occlusions (e.g. green box).

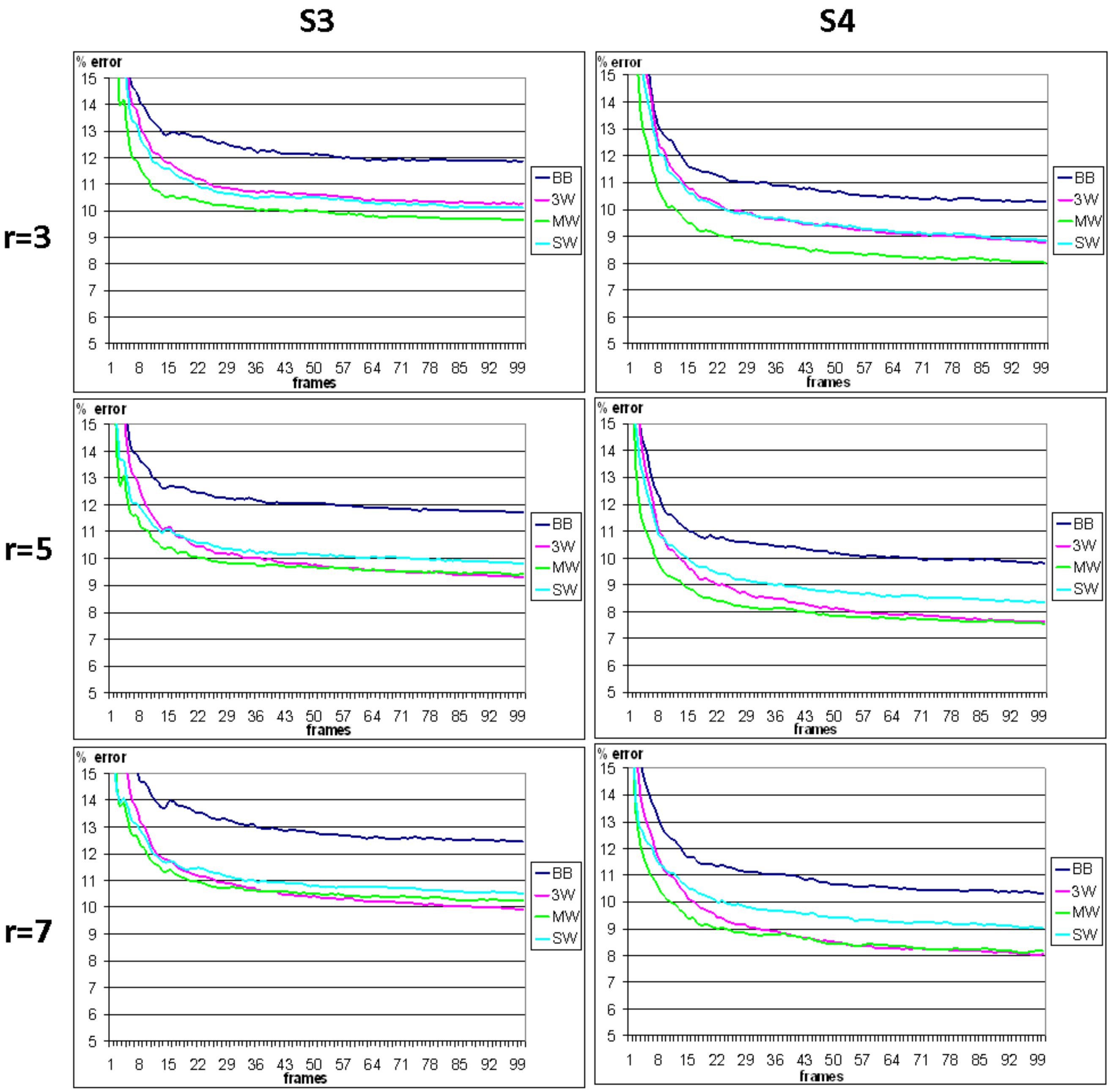


Fig. 4. Experimental comparison of different aggregation strategies on sequences $S3$, $S4$.

III. The proposed 3D framework

In this section we present a framework based on spacetime stereo designed to provide efficient and accurate 3D reconstruction. In particular, we propose the use of several stages to filter and improve the spacetime output. As a matter of facts, the output of the spacetime stereo algorithm, though being usually really precise, still present inaccuracies due to presence of noise, specularities and occluded regions, that ought to be filtered out. The proposed 3D reconstruction pipeline is shown in fig. 6.

A. Spacetime stereo

As for the spacetime stereo module, we apply the MW algorithm given the benefits demonstrated in the previous section. Our implementation of this stage works at ~ 300 ms per processed frame (e.g. with F=20 approximately 6 seconds) using images of size 640×480 and disparity ranges of $70 \sim 90$ pixels.

B. Fast consistency check

In order to discard mismatches and occluded regions, we propose to apply a left-right consistency check [13] after

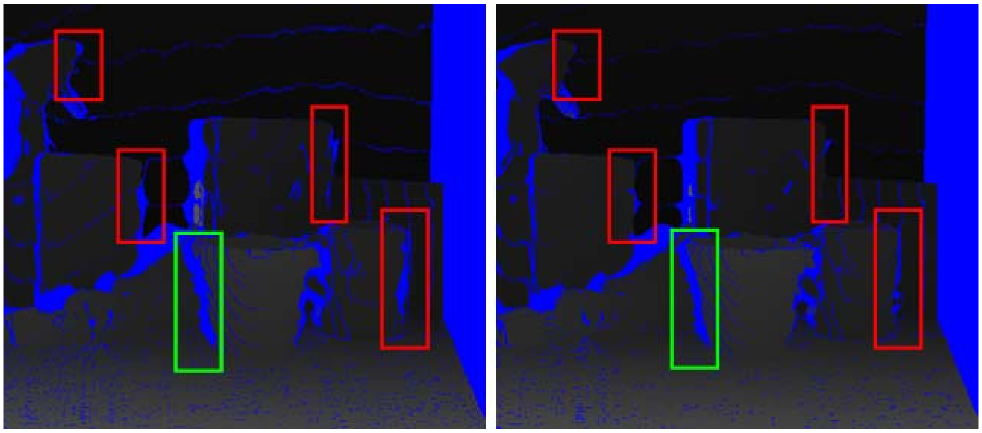


Fig. 5. Comparison between BB (left) and MW (right) on S1 dataset.

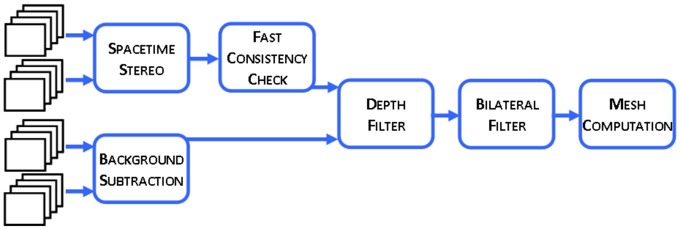


Fig. 6. The proposed 3D reconstruction pipeline.

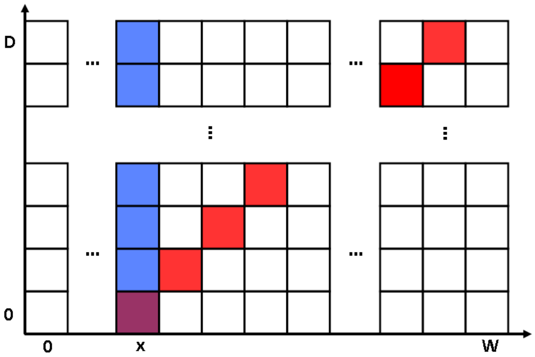


Fig. 7. The fast consistency check method.

processing the stereo sequence. The use of this filtering stage is particularly effective since it keeps only those disparities that are consistent between the two views, hence that can be considered reliable estimations. The main drawback is that, generally speaking, this step would require the computation of two disparity maps, the one referred to the reference view, and the one related to the target view (we will refer here to the two maps as *reference map* and *target map*). This notably affects the considered framework since indeed spacetime stereo is the most computationally expensive stage of the overall pipeline. Hence introduction of the left-right consistency check would almost double the computational time required to perform an object scan.

Nevertheless, we apply an efficient left-right consistency check, inspired by the approach proposed in [14] for trinocular stereo, that computes the target map by recycling the computations needed for the reference map. In particular, to compute cost (1), the spacetime approach needs to compute a three-dimensional array where for each $x \in [0, W]$, $y \in [0, H]$, $d \in [0, D]$ (W,H image size, D the disparity range) the cost associated with the local support (e.g. (2) for method BB) is accumulated. Figure 7 shows a slice of this array computed for a certain value of y. By properly analyzing this array we can determine the best disparity for each point of the reference map (traversing the blue path in figure 7)

$$\tilde{d}_{ref}(x,y) = \underset{d \in [0,D]}{\operatorname{argmin}} C_{sts}(x,y,d) \tag{7}$$

as well as the best disparity for each point of the target map (traversing the red path in figure 7)

$$\tilde{d}_{tar}(x,y) = \underset{d \in [0,D]}{\operatorname{argmin}} C_{sts}(x+d,y,d) \tag{8}$$

with no need to recompute the 3D array. In our experiments, using images of size 640×480 and a disparity range of $70 \sim 90$ pixels, the computation of the target map and the left-right consistency check altogether accounted for a total computational overhead as small as $150 \sim 200$ ms.

C. Background subtraction and depth filter

These stages are aimed at segmenting the foreground object from the scene background. This is performed both in the intensity domain as well as in the disparity domain.

As for the former, at start-up a background model is computed by averaging some frames of the scene in absence of the object to be reconstructed. At run-time, for each object that needs to be reconstructed, an average appearance frame (without pattern projection) is computed, then a simple pixelwise background subtraction is performed by thresholding those pixels in the current frame that are distant in the RGB space to the corresponding pixels in the background model. In our implementation, the threshold can be interactively adjusted by the user when this stage is reached. In addition,

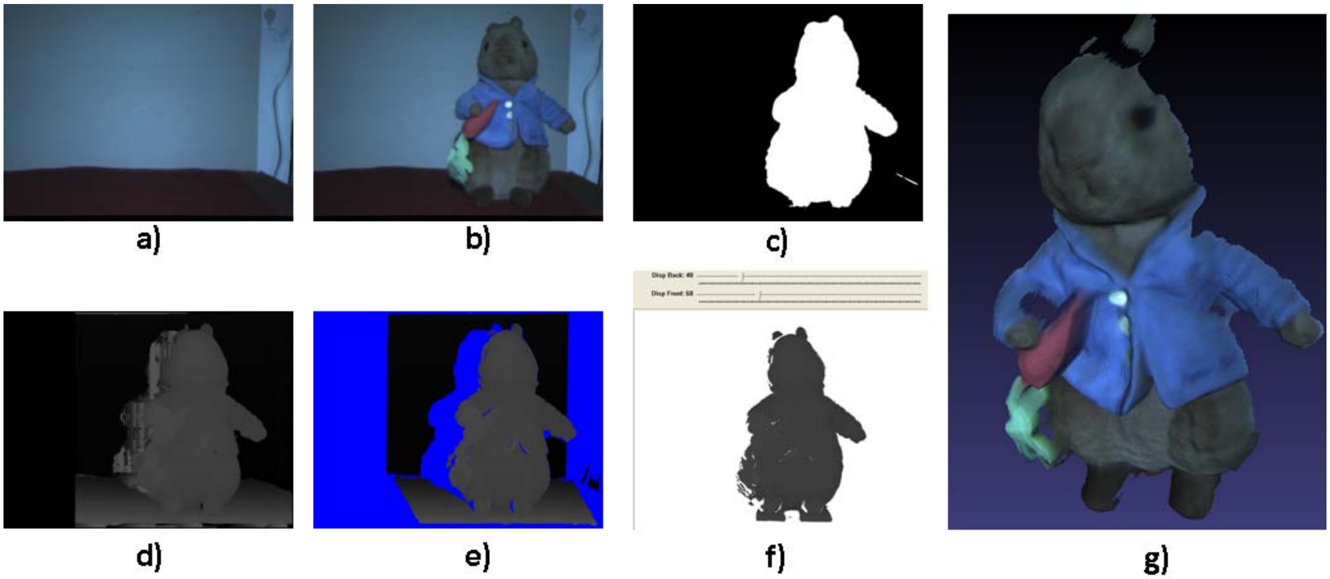


Fig. 8. Outputs at the various stages of the proposed reconstruction framework. a) Background model. b) Averaged object appearance (with no projected pattern). c) Binary change mask resulting from background subtraction and area-closing; d) Output of spacetime stereo based on the MW algorithm. e) Unreliable disparities (depicted in blue) filtered out by the fast consistency check. f) Manual adjustment of the depth filter thresholds. g) Resulting 3D mesh.

a simple labeling algorithm is deployed on the binary change mask to perform an *area-closing* step (all background pixels having an area smaller than a given threshold are converted to foreground). This helps reducing false negatives points in the foreground area. The computation of the average appearance of the object is also useful to associate texture information to each element of the point cloud in the mesh computation stage. As for the segmentation in the disparity domain, two thresholds are used to adjust the horopter of the stereo setup so that it contains only the volume where the object is currently scanned. Also at this stage, the thresholds can be interactively modified by the user.

D. Bilateral filter

After the previous stages have discarded those points in the disparity map corresponding to mismatches or to background parts, the remaining disparity values have to be filtered due to the presence of noise. We propose to apply a cascade of two filters. The first one is a median filter, to remove outliers possibly present in the disparity map. Then, we apply bilateral filtering [15], which is a non-linear filter used to denoise images while preserving intensify/color edges. In our case, we apply the bilateral filter on disparity images so as to smooth out the object surface while preserving depth borders (discarded points are excluded from the filtering process). In our experiments, with images sized 640×480, the computation of the bilateral filter accounted for a processing time of $40 \sim 60$ ms.

E. Mesh computation

During the final stage, disparities are mapped in the 3D space by means of calibration parameters and a meshing algorithm is applied to reconstruct the object surface. We exploit the disparity map to infer information on neighbouring pixels and apply the following simple meshing algorithm: a mesh triangle is created every three 4-connected valid neighbouring points in the disparity map whose difference in depth is never higher than a given threshold. In our experiments, using images of size 640×480, the mesh computation accounted for a processing time of $4 \sim 6$ ms.

Figure 8 shows the output of the proposed reconstruction framework at various stages of the 3D pipeline. In addition, some 3D reconstructions concerning objects and faces obtained with the proposed approach are presented in figure 9. Videos concerning these reconstructions can be found online[1]. As for the experimental setup, we have used a Videre Design stereo camera [16] together with an off-the-shelf lamp projector and PC.

IV. Conclusions

We have shown how the accuracy and convergence speed of standard spacetime stereo can be improved by means of stereo matching algorithms that rely on adaptive window positions. We have also described a practical and cost effective PC-based system based on improved spacetime that allows to obtain rapidly and easily (i.e. with very limited and intuitive user interaction) accurate 3D reconstructions. The result concerning faster convergence is important since it deals with the ability of achieving the required accuracy level using a significantly smaller number of frames than standard spacetime. This, in turn, may widen the range of applications of spacetime stereo, for it implies a less tight immobility constraint of the scene under reconstruction. We believe that this would be particularly attractive in e.g. 3D face recognition applications,

[1] Available at : *http://vision.deis.unibo.it/fede/demos/rec3D.htm*

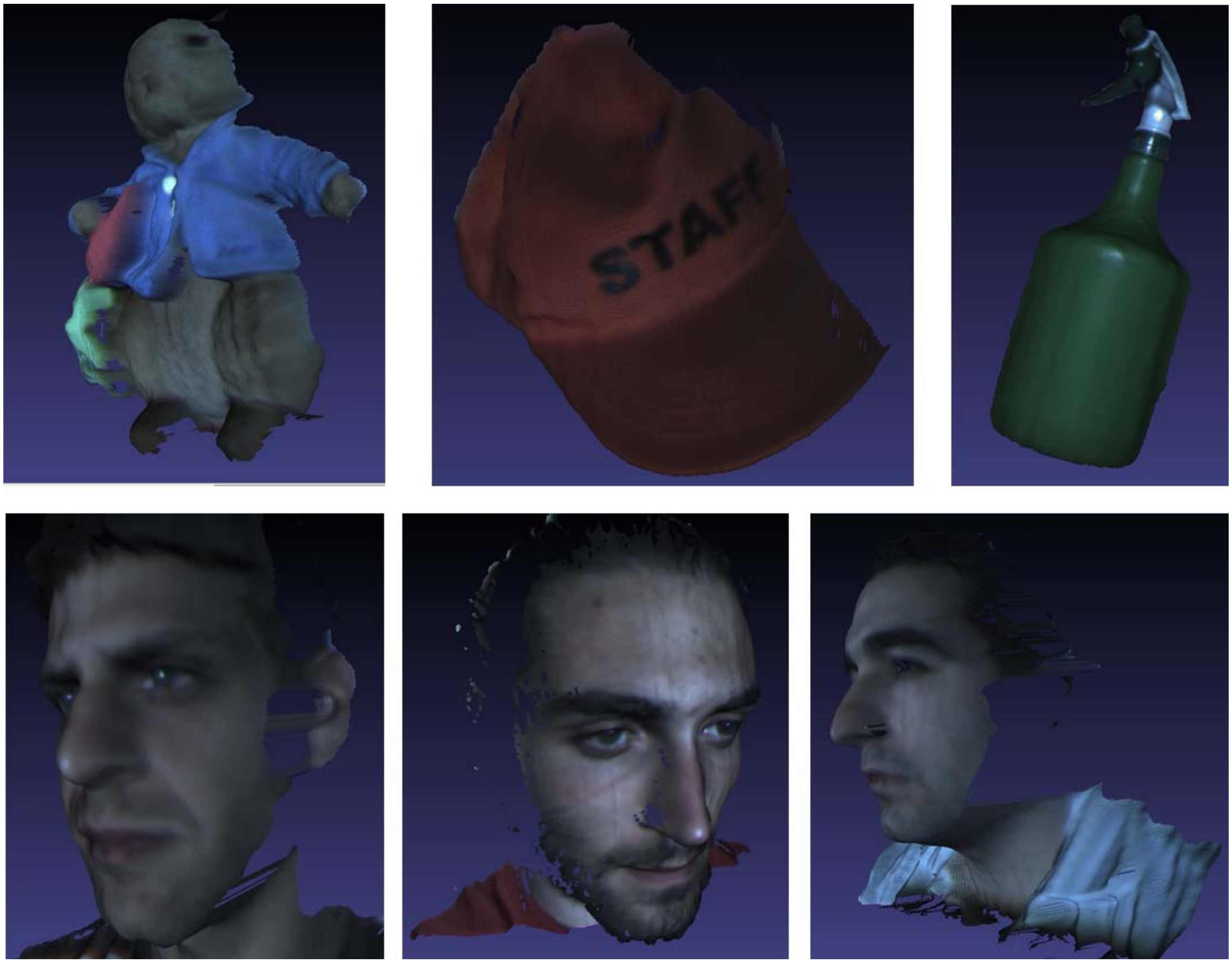


Fig. 9. Examples of reconstructions of objects and faces obtained by the proposed framework.

where a much shorter immobility time may notably reduce the invasiveness of the approach as well as the degree of cooperation required from the user. Hence, in our future work we plan to investigate further on this issues, so as to assess also whether the adaptive support matching approach might allow for an effective extension of spacetime to slowly moving objects/camera.

References

[1] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash ladar devices," in *Int. Conf. of Sensing and Technology*, 2005.

[2] B. Curless and M. Levoy, "Better optical triangulation through spacetime analysis," in *ICCV*, 1995.

[3] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework dor depth from triangulation," *IEEE Trans PAMI*, vol. 27, no. 2, February 2005.

[4] L. Zhang, B. Curless, and S. Seitz, "Spacetime stereo: shape recovery for dynamic scenes," in *Proc. CVPR*, 2003.

[5] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Jour. Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, 2002.

[6] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Proc. CVPR*, 2008.

[7] K. Yoon and I. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. PAMI*, vol. 28, no. 4, pp. 650–656, 2006.

[8] H. Hirschmuller, P. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *Int. Journ. of Computer Vision*, vol. 47, pp. 1–3, 2002.

[9] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. CVPR*, 2003, pp. 556–561.

[10] A. Bobick and S. Intille, "Large occlusion stereo," *Int. Journal Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[11] J. Zhao and J. Katupitiya, "A fast stereo vision algorithm with improved performance at object borders," in *Proc. Conf. on Int. Robots and Systems (IROS)*, 2006, pp. 5209–5214.

[12] L. Sorgi and A. Neri, "Bidirectional dynamic programming for stereo matching," in *Proc. Int. Conf. on Image Processing (ICIP)*, 2006, pp. 1013–1016.

[13] P. Fua, "A parallel stereo algorithm that produces dense depth maps and preserves image features," *Machine Vision and Applications*, vol. 6, no. 1, pp. 35–49, 1993.

[14] T. Ueshiba, "An efficient implementation technique of bidirectional matching for real-time trinocular stereo vision," in *Proc. Int. Conf. on Pattern Recognition (ICPR)*, 2006.

[15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. CVPR*, 1998.

[16] Videre Design LLC, "www.videredesign.com."