Guided Filtering based Pyramidical Stereo Matching for Unrectified Images

Tian Xu
School of Computing Science
University of Glasgow
t.xu.1@research.gla.ac.uk

Paul Cockshott
School of Computing Science
University of Glasgow
William.Cockshott@glasgow.ac.uk

***Abstract*—Stereo matching deals with recovering quantitative depth information from a set of input images, based on the visual disparity between corresponding points. Generally most of the algorithms assume that the processed images are rectified. As robotics becomes popular, conducting stereo matching in the context of cloth manipulation, such as obtaining the disparity map of the garments from the two cameras of the cloth folding robot, is useful and challenging. This is resulted from the fact of the high efficiency, accuracy and low memory requirement under the usage of high resolution images in order to capture the details (e.g. cloth wrinkles) for the given application (e.g. cloth folding). Meanwhile, the images can be unrectified. Therefore, we propose to adapt guided filtering algorithm into the pyramidical stereo matching framework that works directly for unrectified images. To evaluate the proposed unrectified stereo matching in terms of accuracy, we present three datasets that are suited to especially the characteristics of the task of cloth manipulations. By comparing the proposed algorithm with two baseline algorithms on those three datasets, we demonstrate that our proposed approach is accurate, efficient and requires low memory. This also shows that rather than relying on image rectification, directly applying stereo matching through the unrectified images can be also quite effective and meanwhile efficient.**

I. Introduction

Dense-correspondence stereo-matching algorithms are an essential element in computer vision. They are widely used in image and video processing, as well as for robotic vision. Commonly, stereo matching requires rectified images that are computed from calibrated cameras. This pre-processing step used in the stereo matching algorithms is image rectification, a transformation process used to project stereo images onto a common image plane, so that the correspondence points have the same y-direction coordinates. This could essentially simplify the 2D stereo correspondence problem to 1D and various algorithms have been proposed to rectify images [14], [18]. However, parametric camera models used for calibration are only approximations of physical cameras while the rectification process itself can add computational complexity to the algorithm. Even cameras that are calibrated to sub-pixel accuracy can cause large matching errors. Hirschmuller and Stefan [5] present the example of a service robotics scene with round objects, e.g. glasses, where calibration errors of just 0.25 pixel cause artificial disparity discontinuities of 2 pixel. Although the normal process is to first rectify image, and then apply existing stereo matching algorithms, nevertheless, this procedure could be expensive and the non-ideal stereo configurations usually produce inferior results.

Recently, cloth perception and manipulation has become popular and stereo matching can enable the delivery of a robotic system that accomplishes automatic sorting and folding of a laundry heap [9], [13]. However, stereo matching in this context is quite challenging. First of all, since the cloth manipulation requires more accurate representation for the stereo matching (e.g. we need to capture cloth wrinkles), therefore, dealing with high-resolution images is required. This requires the stereo matching algorithm to have low memory cost when dealing with high resolution images. In addition, the cloth manipulation also requires the image matching algorithm not only to be capable of deriving disparity maps from the objects with drastic depth changes (e.g. a box on a table), but also dealing with cloth containing smooth depth changes (such as small cloth wrinkles). Furthermore, due to the real-time nature of the cloth manipulation task, this requires the stereo matching algorithm to be both efficient and accurate. An algorithm that is easily to parallelize is preferred.

To cope with the above challenges and remedy the rectification issue, we adopt a single stage stereo matching methodology that derives the disparity information directly from the unrectified images. Specifically, we adapt a state-of-the-art guided filtering algorithm [4] within a pyramid based stereo matching framework [16] to process directly the unrectified images. This proposed algorithm is capable of processing high resolution images with low memory cost, is easy to parallelize (by multi-core architecture such as GPU), and accurately derives depth information for small cloth wrinkles. To evaluate the performance of our algorithm, we utilize three datasets. The first dataset is created by automatically deriving the ground-truths of the depth map, on drastic depth change (box on the table). The second one is created by generating disparity maps to simulate micro depth changes (small cloth wrinkles). The third dataset is a real-world cloth manipulation dataset that we aim to present the performance in the real task.

The contributions of our paper are two fold:

- We propose a new stereo matching algorithm for unrectified images by adapting the Guided Image Filtering approach (which is previous applied to rectified images) to the pyramidical stereo matching framework, and demonstrate that our algorithm can cope with unrectified images directly, achieving relatively high performance while outperforming current solutions. The proposed approach is accurate, efficient, and requires low memory for processing high resolution images that can capture more details, and it enables effective robot cloth manipulation.

- We present three datasets for evaluating stereo matching in the context of cloth manipulation (one simulated and two real-world), not only considering drastic depth change, but also measuring micro depth change such as cloth wrinkles. The proposed evaluation methodology can be useful and the datasets can serve as standard testbeds for measuring stereo matching for cloth manipulations.

II. Related Work

A. Image Matching Algorithms

Stereo image matching entails discovering the most likely matches between pixels in two images. The technique is widely used in computer vision and robotics, including: data visualization, three dimensional map building and robot pick and place. It has been studied over several decades in computer vision and many researchers have worked at solving it [12]. Stereo matching algorithms could be divided into two categories: local methods and global methods. Local algorithms are statistical methods that are usually based on correlation. For global algorithms, the task of computing disparities is cast in terms of energy minimization, and is solved by various optimization techniques [7], [8]. Compared with local algorithms, global algorithms are normally computationally much more expensive.

There are two main classes of local algorithms: feature-based algorithms and area-based algorithms. Feature-based matching algorithms [2], [15] attempt to establish correspondence of a sparse sets of image features (e.g. edges). Although, feature-based algorithms work very fast, they can only generate sparse disparity maps. So they are not suitable for many applications (e.g. reconstructing surfaces) that require dense disparity maps.

Area-based algorithms [3], [17] are also called correlation-based algorithms. These methods merge the feature detection step with the matching part, which means it deals with the images without attempting to detect salient objects in the images. Correlation-based stereo methods match neighbouring pixel values, within a window, between images. Adaptive support weight [6] approach is one of the classic correlation-based algorithms which gives good quality results. However due to the large amount of time and memory required in both the pixel-wise support window weight computation and the aggregation processes, it is hardly to be considered as an efficient algorithm. The guided filter [4] is a fast and non-approximate linear-time algorithm, whose computational complexity is independent of the filtering kernel size. This state-of-the-art filter also has the edge-preserving property, which preserves object boundaries and depth discontinuities. Some researches [10], [11] have demonstrated that the guided filter is both effective and efficient in stereo matching applications.

To the authors' knowledge, all of previous matching algorithms using guided filter are applied on rectified stereo images, however, the facilities of guided filter could also be applied on matching algorithms for unrectified images. In this paper, we demonstrate that adapting guided filter to pyramidical correlation based stereo matching framework [16] could also achieve high quality result when applying on large unrectified images. In addition, we also apply this stereo matching to the cloth manipulation context that recently becomes popular [9], [13].

B. Stereo Matching Image Datasets

In order to compare stereo matching algorithm's performance, several datasets have been developed, while among them, Middlebury Stereo Datasets [12] is the most popular one. There are various versions of Middlebury Stereo Evaluation datasets while the most popular version (Version 2) [1] generally contains low resolutions images of less than 640×480 pixels. The most recent version[2] maintains images of higher resolutions, up to around 3000×2000 pixels. Low resolution could not satisfy today's resolution requirement for stereo matching as it may cause the loss of image details. From both accuracy and efficiency perspective, the algorithms that perform well for small images may not perform well for high resolution images.

However, we do not use Middlebury datasets for our evaluation. This is due to the fact that all the stereo images are rectified images, which makes it not suitable for evaluating matching algorithms for unrectified images. Furthermore, the Middlebury dataset is not suitable to evaluate stereo matching algorithm performance in the context of cloth manipulations. Although containing a few cloth related images, the dataset does not cover a wide range of materials and cloth pose configurations for thorough evaluation. Considering the reasons above, we evaluate our algorithm and baseline algorithms on three new datasets in the context of cloth manipulations.

III. Unrectified Matching Algorithm

The aim of the stereo matching algorithm is to compute a disparity map for a stereo-pair of images. The disparity map refers a two dimensional array of displacement vectors which map the pixels in one image onto their corresponding pixel in the other. In the case of un-rectified images, i.e. images from converged stereo cameras, there are two disparity maps, one for horizontal displacements and one for vertical displacements, specifying orthogonal components of the disparities.

1) Pyramid Creation: The matching algorithm employs a pyramid representation (Figure 1), which is applied to the input images. Pyramid level is adjustable to make sure the image size from top pyramid level is small enough, and the disparities for this level are considered to be equal or less than one pixel. In this multiple scale scheme, an initial estimate for the disparity is computed at a low resolution and the initial disparity estimate from this scale is refined at higher resolutions until the target resolution is achieved. The strategy to generate and refine the estimate is described as below.

[1] http://vision.middlebury.edu/stereo/eval/
[2] http://vision.middlebury.edu/stereo/eval3/

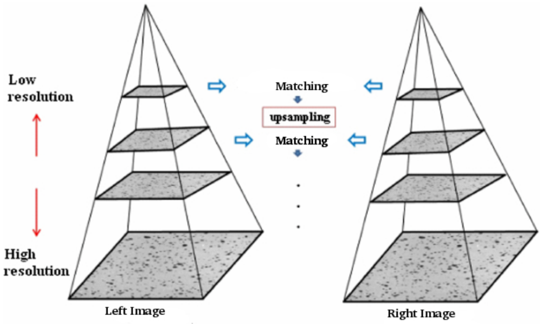


Fig. 1: Pyramid representation of stereo input image to perform matching at multiple scales

2) Cost Computation and Aggregation: In order to generate or refine the disparities at each level of the input pyramid, the algorithm attempts to maximize the similarity between pixels in windowed regions of the left and right images.

First, color difference is calculated between pixel p of the left image and the pixel at coordinates $(p-d)$ of the right image. d is the disparity between two coordinates of the pixels from both images. The color differences $D(p,d)$ for matching pixel p at disparity d are computed as:

$$D(p,d)=\sum_{i=1}^{3}|I^{i}_{left}(p)-I^{i}_{right}(p-d)| \quad (1)$$

where i demonstrates the ith color channel in RGB space. Then the aggregated matching costs $C(p,d)$ at pixel p and disparity d are computed as:

$$C(p,d)=\sum_{q\in W_p} W(p,q)D(q,d) \quad (2)$$

In the equation above, weighting function $W(p,q)$ computes the likelihood that pixel q lies on the same disparity with the window's central pixel p. The windowed correlation is supported by guided filter [4], whose weighting function $W(p,q)$ is defined as follows:

$$W_{(p,q)}=\frac{1}{|w|^2}\sum_{k:(p,q)\in w_k}(1+\frac{(I_p-\mu_k)(I_q-\mu_k)}{\sigma_k^2+\epsilon}) \quad (3)$$

Here, I is the reference image, μ_k and σ_k^2 are the mean and variance of I in window w_k centered at the pixel k, $|w|$ is the number of pixels in this window, and ϵ is a regularization parameter.

The window in the left image is named as reference window and the window in the right image is called search window. Because for the unrectified images, both horizontal and vertical disparities need to be considered. So the search window is moved in four directions (up, down, left and right) using a one pixel search step. If we include the null move, there are in total five positions. For each position, aggregated matching costs $C(p,d)$ is computed.

3) Polynomial Minimization: Similar to the Parallel Pyramid Matcher [16], after obtained five aggregated matching costs matrices, 2nd order polynomial minimization is applied to the corresponding elements of the matrices. The task is to find, for each pixel, the local minimum of the curve, which indicates the position that has the minimum cost. The position could be sub-pixel position to achieve high accuracy. If the local minimum is found to be more than one pixel away from the current position, the relative displacement is clipped to ± 1.

4) Rescale: From current scale to higher resolution scale, interpolation is necessary for each pixel in the disparity map. An interpolated pixel is derived from its four neighbours. If

$$x_0=floor(x) \quad y_0=floor(y) \quad (4)$$

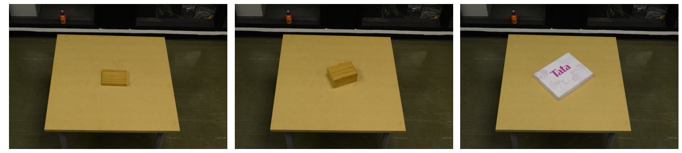
$$a=x-x_0 \quad b=y-y_0 \quad (5)$$

then

$$\begin{aligned} f_i(x,y)=&(1-a)(1-b)f(x_0,y_0)+(1-a)bf(x_0,y_0+1)\\ &+a(1-b)f(x_0+1,y_0)+a\cdot b\cdot f(x_0+1,y_0+1) \end{aligned} \quad (6)$$

The interpolated result f_i is the initial disparity map for the next higher scale resolution.

Iteratively implementing the same matching algorithm on each scale until the highest resolution scale, the final and more accurate disparity map is produced. Notice that in the context of cloth manipulation, we generally deal with garments with continuous surface placed on the table where no occlusion occurs in the area of interests (i.e. table area, see Fig. 5 for an example). Therefore, we do not need to explicitly consider occulsion detection in our stereo matching.



(a) First Image (b) Second Image (c) Third Image

Fig. 2: Examples of real scene with box on table dataset

IV. Evaluation Methodology

To evaluate the stereo matching algorithm performance, there are three datasets introduced: real scene with box on table dataset, simulated cloth dataset and garments dataset.

A. *Real scene with box on table dataset*

This dataset is formed by 24 images, and each scene shows a table with a rectangle box placed on it. Three boxes are used for this dataset, and each box is placed on the table with several different positions (see Figure 2 for an example). In order to show more details of the table and the box, the size of stereo images we captured is 4928×3264. The three boxes have different length, width and height, which have been carefully measured by vernier caliper before generating the dataset. A pair of cameras, focusing on the same point of table, are used to simultaneously capture stereo images.

Stereo matching algorithms for unrectified images could be applied on these image pairs directly, because these images are not rectified. With known camera parameters, it is also possible to rectify these images in order to apply matching

(a) Left Image (Identical Right Image) (b) Right Image with Horizontal sin Transform (c) Right Image with Diagonal sin Transform

Fig. 3: Example of cloth's left sand simulated right image

algorithms that are only available for rectified images. In the rest of the paper, we utilize the rectification algorithm provided by openCV[3] to rectify the images for our baseline algorithm applied to rectified images.

The ground truth for disparity map is difficult to generate, but since the depth of the box is known, we could use depth map to evaluate the algorithm performance. No matter the algorithms are applied on rectified or unrectified images, the depth of the box remains unchanged. So for each algorithm, we first try to fit the table surface and top box surface using least squares fitting method. And then we use both surface equations to calculate the distance between the box and the table using the central point of the box's top surface. By comparing this estimated depth of box with the ground truth depth of the box, we could measure how accurate is the depth map generated by a given stereo matching algorithm.

To make the evaluation more comprehensive, the depth map for the table and the box are generated for comparison. Because every image pair has exactly the same table placed at the identical location, it is easier to use the whole dataset to estimate the equation of the table plane given an assumed coordinate system. Several algorithms are applied on these image pairs to estimate the table surface, but when fitting the plane, only the root mean square error (RMS) less than a pre-defined threshold would be counted. The average of the counted estimation is recognised as the ground truth of the table. Assuming the top box surface is parallel to the table surface, the top box surface could be calculated by moving the table plane with the value of box height according to perpendicular direction.

B. Simulated cloth dataset

Because this stereo matching task is based on clothes, an image dataset for cloth textures is generated by the authors to test whether the algorithms could handle all kinds of cloth textures successfully. This dataset contains 78 images. A collection of different types of cloth images (e.g. cotton, wool, silk, leather, etc.) is used for this dataset. Images are cropped into three different sizes, including small size (512×512 pixels), medium size (1024×1024 pixels), and large size (2448×2050 pixels), and each of these single image is treated as left image (reference image).

Unlike left images, which are real cloth textures images, right images are the simulated ones (Fig 3). It is hard to have depth or disparity ground truth, because clothes are always soft and deformable. Thus, we try to simulate cloth wrinkle waves by the sin transform. Horizontal sin transform and diagonal sin transform are applied on the x-axis of left images to generate right images. In addition to the right image generated with no transformation on the reference image, we generate in total three kinds of right images with known disparities.

C. Garments dataset

To provide insight into cloth perception and manipulation with an active binocular robotic vision system, a database of 80 stereo-pair colour images [1] is compiled with corresponding horizontal and vertical disparity maps and mask annotations. This database is based on 16 different off-the-shelve garments. Each garment has been imaged in five different pose configurations on the project's binocular robot head, including: flat on the table, folded in half, completely folded, randomly wrinkled and hanging over the robot's arm. Since there are no depth map ground truths for this dataset, we therefore show several examples in Figure 5.

V. Experiments

Our proposed stereo matching algorithm is implemented using a 4-core Intel Core i5-2400, 3.1 GHZ computer. The GPU is a GeForce GTX770 graphics card with 4GB of memory from NVIDIA. We used the CUDA (Compute Unified Device Architecture) technique of NVIDIA Corporation for implementation on the GPU.

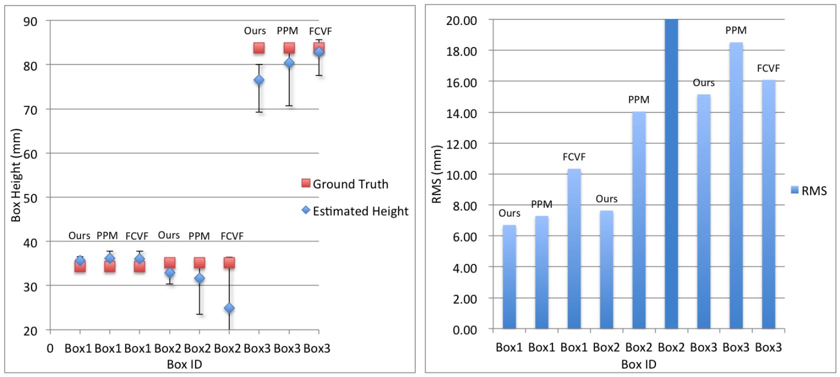
The experiment is based on 3 different datasets as described in the previous section. Our algorithm is evaluated by comparing to another two methods. One is Parallel Pyramid Matcher (PPM) [16], which is also a GPU based matching algorithm that utilizes pyramid based stereo matching framework. Another one is Fast Cost-Volume Filtering (FCVF) [11] for stereo matching on rectified images, which is one of the state-of-the-art algorithm that also involves guided image filtering [4] as part of it. This method is parallelable using GPU, but the authors only made the MATLAB code available. So in this experiment, we mainly focus on accuracy comparison. All experiments are conducted on the same computer.

A. Evaluation on real scene with box on table dataset

As mentioned in the previous section, to preserve details, the images of this dataset are quite large, each image contains 4928×3264 pixels. However, due to the huge memory requirement of Fast Cost-Volume Filtering (FCVF) algorithm to which applying on high resolution images are not applicable, we shrunk the images to 1/16 size (1232×816 pixels) for fair comparison to the other two algorithms. In addition, we also report the matching performance on the images of the original size for PPM and our proposed algorithm.

First the three box's height estimation performance is reported, shown in fig 4(a). Ground truth represents the real box height while the estimated height provides the average estimated height based on the matching algorithms. The black lines around the estimated height provide the range of estimated height (i.e. variance). Our algorithm performs the

[3] docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html



(a) Three box height estimates (b) RMS in pixel range for three algorithms

Fig. 4: Evaluation on real scene with box on table dataset (1/16 size images)

TABLE I: Effect of sub sampling on height errors

	Ours		PPM	
	Avg. % of Height Error	Plane Fitting RMS	Avg. % of Height Error	Plane Fitting RMS
1/16 Size	6.38%	11.47 mm	6.60%	12.16 mm
Full Size	**4.78%**	**9.60** mm	6.55%	9.91 mm

best when applied on boxes with smaller height, maintaining the least error range as well. For the box that has higher height (i.e. more drastic depth change), our algorithm could not perform as well as the other two algorithms. However, in the context of cloth manipulation, the depth changes on the cloth are normally smooth and do not maintain drastic change, we believe our algorithm can suit better and provide more accurate depth map dealing with cloths (as we also empirically demonstrate from the experiments conducted below).

Along with the box's height estimation, we also investigated the root mean square error (RMS) for the box's depth map (fig 4(b)). This provides the more refined evaluation than the single point height estimate. On average, our algorithm gives the least root mean square error, which indicates that our algorithm is the most stable matching algorithm among the three.

As we mentioned before, due to the huge memory requirement, Fast Cost-Volume Filtering algorithm can not run on the original size of the images. But we conduct another experiment to test whether the high resolution images (with more details) could lead to a better accuracy. Parallel Pyramid Matcher and our algorithm are both applied on the original images and the shrunk images. In addition, the output depth map of the shrunk images are enlarged to the original size to compare with ground truth depth map. The average performance is given on table I.

Among all, our algorithm with the full size images achieves the best performance on both the percent of height error and plane fitting RMS. The table also indicates that images with more details could help improving the matching accuracy, which is not surprising. Compared with 1/16 size images, our algorithm on the original size images reduced 25% of percent of height error and 16% of plane fitting root mean square error.

TABLE II: Disparity map evaluation on simulated cloth dataset

	Bad Pixel Rate(%)	Avg. Error(Pixel)	RMS(Pixel)
Ours	**1.184**	**0.083**	**0.431**
Parallel Pyramid Matcher	8.519	1.453	6.139
Fast Cost-Volume Filtering	4.443	0.711	1.309

B. *Evaluation on simulated cloth dataset*

To test how well an algorithm could handle cloth textures, we introduce three evaluation criteria as shown in table II. Bad Pixel Rate investigates when comparing algorithm's output disparity with ground truth disparity, how many percentages of pixel's error are larger than a threshold (1 pixel in this experiment). Average Error means the average absolute difference between the output and the ground truth disparity. Root Mean Square error (RMS) evaluates the square root of the mean of the squares of the disparities difference.

Table II shows the average performance of all kinds of different cloth textures. Among the table, our proposed algorithm achieves the best result. Only about 1% of pixels are bad pixel. In addition, the average error and root mean square error of disparity are both less than 0.5 pixel.

C. *Performance on garments dataset*

This is the specific task our algorithm really focus on. Figure 5 shows examples of one garment on the table. The figure shows that our approach is able to produce detailed and accurate depth maps, which is able to preserve fine shape and tiny wrinkles of clothes.

D. *Evaluation on Efficiency*

Efficiency evaluation is based on image datasets used in this paper. Because our algorithm and PPM algorithm are both implemented by GPU code, which is quite efficient, while FCVF one is using Matlab code, so timing comparison is not our main focus. From fig 6, we found that all the algorithms show linear trend, however, timing of our algorithm and PPM

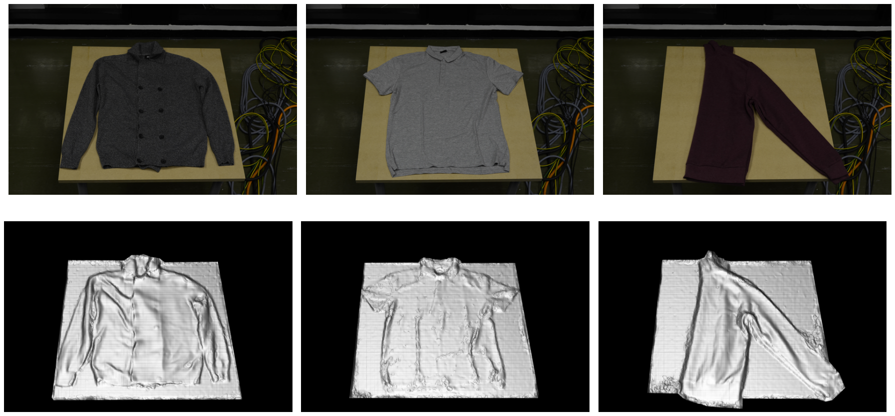


Fig. 5: Example range map of garments dataset

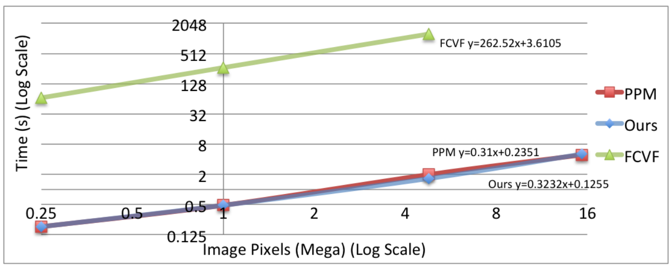


Fig. 6: Relationship between Image Size and Execution Time

one are independent from the number of disparity levels, while timing for FCVF is related to not only image size, but also disparity levels. So if the number of disparity levels increases, FCVF algorithm is expected to spend more time to compute. From this perspective, our algorithm is an efficient algorithm.

VI. Conclusions

As robotics becomes popular, conducting stereo matching in the context of cloth manipulation is useful and found to be quite challenging. This is resulted from the high efficiency, accuracy and low memory requirement under the usage of high resolution unrectified images for robotic cloth manipulations such as folding.

Therefore, we adapt a state-of-the-art guided filtering algorithm within a pyramid based stereo matching framework that works directly for unrectified images, and is shown empirically better suited to the characteristics of the task of cloth manipulations and easily paralleled. By comparing the proposed algorithm with two baseline algorithms on three created datasets, we demonstrate that our proposed approach is accurate, efficient and requires low memory. This also shows that rather than relying on image rectification, directly applying stereo matching through the unrectified images can be also quite effective and meanwhile efficient.

References

[1] G. Aragon-Camarasa, S. B. Oehler, Y. Liu, S. Li, P. Cockshott, and J. P. Siebert. Glasgow's stereo image database of garments. *arXiv preprint arXiv:1311.7295*, 2013.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 489–495. IEEE, 1999.

[3] O. Faugeras, T. Viéville, E. Theron, J. Vuillemin, B. Hotz, Z. Zhang, L. Moll, P. Bertin, H. Mathieu, P. Fua, et al. Real-time correlation-based stereo: algorithm, implementations and applications. 1993.

[4] K. He, J. Sun, and X. Tang. Guided image filtering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(6):1397–1409, 2013.

[5] H. Hirschmuller and S. Gehrig. Stereo matching in the presence of sub-pixel calibration errors. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 437–444. IEEE, 2009.

[6] K. jin Yoon, S. Member, and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. PAMI*, 28:650–656, 2006.

[7] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 15–18. IEEE, 2006.

[8] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515. IEEE, 2001.

[9] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2308–2315. IEEE, 2010.

[10] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 467–474. IEEE, 2011.

[11] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3017–3024. IEEE, 2011.

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[13] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. Siebert. Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 185–192, May 2015.

[14] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.

[15] V. Venkateswar and R. Chellappa. Hierarchical stereo and motion correspondence using feature groupings. *International Journal of Computer Vision*, 15(3):245–269, 1995.

[16] T. Xu, P. Cockshott, and S. Oehler. Acceleration of stereo-matching on multi-core cpu and gpu. In *High Performance Computing and Communications, 2014 IEEE Intl Conf on*, pages 108–115. IEEE, 2014.

[17] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision ECCV'94*, pages 151–158. Springer, 1994.

[18] Z. Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, 2000.