2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation

Paper Number: 152188089

GPU-based Parallelization of a Sub-pixel High-resolution Stereo Matching Algorithm for High-throughput Biomass Sorghum Phenotyping

Yin Bao, Lie Tang, Patrick S. Schnable, Maria G. Salas Fernandez
Iowa State Univeristy

Written for presentation at the

2015 ASABE Annual International Meeting

Sponsored by ASABE

New Orleans, Louisiana

July 26 – 29, 2015

Abstract. *To automate high-throughput phenotyping for infield biomass sorghum morphological traits characterization, a capable 3D vision system that can overcome challenges imposed by field conditions including variable lighting, strong wind and extreme plant height is needed. Among all available 3D sensors, traditional stereo cameras offer a viable solution to obtaining high-resolution 3D point-cloud data with the use of high-accuracy (sub-pixel) stereo matching algorithms, which, however, are inevitably highly computational. This paper reports a GPU-based parallelized implementation of the PatchMatch Stereo algorithm which reconstructs highly slanted leaf and stalk surfaces of sorghum at high speed from high-resolution stereo image pairs. Our algorithm enhanced accuracy and smoothness by using L2 norm for color distance calculation instead of L1 norm and speeded up convergence by testing the plane of the lowest cost within a local window in addition to the original spatial propagation. To better handle textureless regions, after left-right consistency check, the disparity of an occluded pixel is assigned to that of a nearby non-occluded pixel with the most similar pattern. Some of these occluded pixels in textureless region would survive a following left-right consistency check. Therefore more valid pixels would exist in textureless regions for occlusion filling. Accuracy and performance were evaluated on Middlebury datasets as well as our sorghum datasets. It achieved a high ranking in Middlebury table of subpixel precision and revealed subtle details on leaf and stalk surfaces. The output disparity maps were used to estimate stalk diameters of different varieties and growth stages. The results showed high correlation to hand measurement.*

Keywords. *High-resolution, Sub-pixel, High-throughput, GPU, PatchMatch Stereo, Sorghum, Field phenotyping.*

1. Introduction

Sorghum has become an attractive alternative to corn as a bioenergy feedstock (Rooney et al., 2007). Automation of high-throughput phenotyping for infield biomass sorghum morphological traits characterization would play an important role in breeding high yield hybrids among a large number of genetic lines. A 3D vision system capable of overcoming challenges imposed by field conditions such variable lighting, strong wind and extreme plant height is the foundation for 3D plant reconstruction and any analysis beyond. Available 3D sensing technologies include LIDAR, Time-of-Flight camera, structured light camera, 4D light field camera and stereo camera. Among all, stereo camera is a low-cost high-resolution solution. It consists of two color cameras with a small baseline between them to simulate human binocular vision. However, the challenge comes from the stereo matching algorithm. It has been an active research area for decades. State-of-the-art accuracy is achieved by computationally expensive global optimization techniques while fast algorithms often produces depth maps that lack of smoothness over slanted surfaces. For high-throughput infield sorghum phenotyping, canopies and stalks are often slanted to the 3D vision system. Therefore, an efficient algorithm that could handle slanted surfaces is more suitable for 3D plant reconstruction.

Stereo matching is the process of matching each pixel in the left image with its correspondence in the right image. The methods can be roughly classified into local and global (Scharstein et al., 2002). Local methods find the best match for each pixel by evaluating window-based correlation (Yoon et al., 2005). Since only a local pattern is used, local methods suffer from textureless and repetitive regions. Global methods penalize the disparity change of neighboring pixels and obtain a piece-wise smooth disparity map via discrete optimization algorithms such as Graph Cuts (Boykov et al., 2001; Taniai et al., 2014) and Belief Propagation (Sun et al., 2003; Besse et al., 2012) or continuous variational methods (Pock et al., 2008; Ranftl et al. 2012; Heise et al., 2013). Despite its superiority over local methods, global method is extremely time-consuming. For infield plant phenotyping, one advantage of stereo camera over other 3D sensors is its high resolution. Therefore, global methods are not for high-throughput processing.

For plant growth estimation, Lati et al., 2013 used hue-invariant transformation for robust green plant segmentation and adopted dynamic programming (DP) to solve scanline optimization for the smoothness constraint. Hue-invariant transformation requires tuning the threshold for different stages of plant growth. It works well when the plants are healthy and green. However, at late stage, it would be difficult to segment yellowish parts and sorghum head which are close to soil, straw and debris in color. On the other hand, since DP is only a scanline optimization and there is no smoothness constraint between scanlines, DP is known for producing streaking effect and disparities across scanlines may not be smooth.

Local methods are far more efficient. Among various local methods, PatchMatch Stereo (Bleyer et al. 2011) searches for a 3D plane for each pixel to correct the bias towards front-parallel planes favored by traditional square window used by most local methods. It preservers the smoothness of slanted surface instead of generating a staircase effect, which is important to reconstruct free-form surface of sorghum leaf and stalks that are not parallel to the image plane.

Although PatchMatch Stereo is a window-based method and does not enforce smoothness between neighbors, the use of 3D plane in disparity space allows a square window in one image to shear and scale in the other image, producing a more accurate matching cost function for slanted surfaces. Its randomized searching strategy frees it from integer disparity estimation and sub-pixel accuracy is directly estimated. Its random memory access makes it hard to optimize. Thus it is slower than some other local methods which build a cost volume to find the best disparity for each pixel using a square window (Rhemann et al. 2011).

PatchMatch Stereo is a serial algorithm. It can be parallelized by simply dividing the image into chunks and running the algorithm on each individual part. This kind of penalization suits multi-core CPU.

Graphics Processing Unit (GPU) is a specialized device to accelerate computer graphics and image/video processing. For massive parallelization, it typically brings more performance gain. This paper presents a GPU-based parallelization which improves the original serial PatchMatch Stereo in terms of both accuracy and performance. Our algorithm was applied to reconstruct infield biomass sorghum for trait estimation. The experimental results showed promising potential for high-throughput plant phenotyping.

2. Materials and methods

2.1. 3D reconstruction

2.1.1. Serial PatchMatch Stereo

For each pixel p, PatchMatch Stereo (PMS) seeks a plane f_p which minimizes a matching cost function. The plane is parameterized by three variables, a, b and c. Its disparity d_p can be computed as

$$d_p = ax_p + by_p + c, \tag{1}$$

where x_p and y_p are the pixel p's image coordinates. The matching cost function for pixel p is a weighted sum of pixel-wise dissimilarity $\rho(q, q')$ within a square window W_p centered at p as

$$mc(p, f_p) = \sum_{q \in W_p} w(p, q) \cdot \varphi(q, q - (ax_p + by_p + c)). \tag{2}$$

The weight function returns high values if a neighbor pixel q has similar color to the center pixel p. The assumption is that adjacent pixels on the same surface tend to be similar in color.

$$w(p, q) = e^{-\frac{\|I_p - I_q\|}{\epsilon}}. \tag{3}$$

The parameter ϵ controls how fast the weight drops given $\|I_p - I_q\|$ which is the L1 norm in RGB color space. The function $\rho(q, q')$ calculates pixel dissimilarity between pixel q and pixel q' in the other image as

$$\varphi(q, q') = (1 - \beta) \cdot \min(\|I_{q'} - I_q\|, \theta_{col}) + \beta \cdot \min(\|\nabla I_{q'} - \nabla I_q\|, \theta_{grad}), \tag{4}$$

where $\|\nabla I_{q'} - \nabla I_q\|$ computes absolute difference of gradient. θ_{col} and θ_{grad} are truncation thresholds to make cost function more robust in occluded regions where the correspondence does not exist. β balances the contribution between color difference and gradient difference. After camera calibration and stereo image rectification, q and q' have the same row index. In equation (2), q represents a pixel in the left image. If q is in the right image, then

$$q' = q + (ax_p + by_p + c). \tag{5}$$

The introduction of a plane can lead to lower matching cost on slanted surface compared to using a square window for both images since a square window in one image could be sheared and scaled in the other image to adapt. However, the number of possible planes is infinite. PMS adopted a uniform random sampling scheme to initialize the disparity map and plane parameters. Each disparity is randomly generated within the possible range. And each plane is determined by a random unit vector (n_x, n_y, n_z). Both distributions are uniform. The plane parameters, a, b and c are computed as

$$a = -\frac{n_x}{n_z},\ b = -\frac{n_y}{n_z},\ c = \frac{n_x x_p + n_y y_p + n_z d_p}{n_z}. \tag{6}$$

The algorithm is an iterative process. It traverses the image from top left to bottom right for even iterations and the opposite direction for odd iterations. When it visits a pixel, the algorithm finds the minimum matching cost using the disparities and planes of current pixel, left neighbor and upper neighbor in even iterations (current pixel, right neighbor and lower neighbor in odd iterations), which is called spatial propagation. Then the disparity and plane of current pixel q's correspondence q' in the other image is tested. This is referred to as view propagation for each pair of correspondence should have the same disparity.

Because random sampling is not possible to sample all correct disparities and planes, the algorithm has a plane refinement step to search better solutions for each pixel. It randomly picks a disparity from an exponentially reducing range and perturbs the unit vector with an exponentially reducing magnitude. Then it evaluates the new set of parameters to see if a lower cost can be reached. The process terminates when the disparity searching range is less than a threshold.

After the iterative process, post-processing handles occluded regions. Occlusion often happens at depth discontinuities (object boundaries). Occluded pixels are identified by checking disparity consistency of each correspondence pair. If the absolute difference is larger than 1, it is considered occlusion. For an occluded pixel, its disparity is extrapolated using the plane of the nearest non-occluded pixel. This process produces streaking effect at occluded regions. Therefore a weighted median filter is applied only to occluded pixels for the final result.

2.1.2 GPU-based Parallel PatchMatch Stereo

CUDA is a parallel computing platform and programming model to use CUDA-enabled NVDIA GPUs. A grid of thread blocks is created for 2D image data such that each pixel of output disparity map is processed by a unique thread. All threads run the same kernel function.

Serial PMS sequentially processes each pixel and current process can make use of the available results from current pixel's neighbors, which makes the algorithm efficient. However, from a practical point of view, random memory access harms performance most for CPU-based machines. Such problem can be alleviated by GPU texture memory. It allows faster memory access if adjacent threads access adjacent locations.

Our parallel PMS adopted the idea of spatial propagation, view propagation and plane refinement. Since GPU thread execution order is unknown, our spatial propagation tests the disparities and planes of all eight neighbors around a pixel to help propagate correct disparities towards all directions. Such propagation also avoids streaking effect which would happen if there is no interaction between adjacent threads.

In order to further accelerate spatial propagation, our PMS stores the minimum matching cost per each pixel in a cost map and the disparity and plane parameters from the minimum cost within a local window is tested as well. The reason is that if a nearby pixel, which does not belong to the 8 neighbors, has the minimum matching cost, it may share the same disparity with the current pixel.

The kernel function consists of spatial propagation, view propagation and plane refinement. One processing iteration is done by launching the kernel function once. Between iterations, all threads are synchronized.

Local stereo matching methods cannot handle textureless region. To remedy this drawback, our post-processing includes two stages. In the first stage, invalid pixels are identified through left-right consistency check where textureless regions are also detected. Then for each invalid pixel, the nearest valid pixel in the same row with the most similar local pattern is found using the same matching cost but a square window because similar pattern indicates similar disparity within a local area. This stage would produce more valid disparities in textureless region. The second stage includes another left-right consistency check followed by a weighted median filter as in the original PMS. The full algorithm is summarized in Algorithm 1 and 2.

Algorithm 1. Parallel PatchMatch in CUDA.

Initialize random disparity and plane parameters for both frames.

Compute gradient for both frames.

Set up texture memory for each channel of the color image pair and gradient images.

For i = 1 to *N* **do**

  Launch kernel function.

End for

Left-right consistency check.

Recover textless regions.

Left-right consistency check.

Fill occlusion and apply weighted median on occluded pixels.

Algorithm 2. Kernel function.

```
// Do the following for the left frame
Spatial propagation over the 8 neighbors of the center pixel.
Find the minimum cost within a local window in the cost map and test the corresponding disparity and plane parameters.
View propagation of the center pixel.
Plane refinement.
Update cost map.
Repeat the above steps for the right frame.
```

2.1.3 Implementation

Texture memory was used to store each channel of the color stereo image pair and gradient images. Linear filter mode offers linear interpolation between two neighboring pixel values. Since PMS directly estimates sub-pixel disparity, linear filter mode facilitates the evaluation of window-based matching cost.

In the kernel function, when evaluating the matching cost, our implementation did not manually handle illegal addressing beyond image width or below zero because putting conditional statements in the kernel function costs performance. There are 4 types of address mode for texture memory, Wrap, Clamp, Mirror and Border. Wrap mode produced the best accuracy around image borders in our experiment.

For generating random real numbers, cuRAND library was used. It initialized random states for each thread, which took about 15 seconds for a stereo image pair of 2 megapixels. Since the random states can be reused once initialized, the time is negligible for batch processing.

2.2. Evaluation datasets

Middlebury stereo datasets (http://vision.middlebury.edu/stereo/data/) were used to evaluate the accuracy of our parallel PatchMatch Stereo in comparison to the original algorithm. It is a widely accepted stereo benchmark started in 2002. To date, 162 algorithms have been registered. The datasets consist of 4 stereo image pairs known as Tsukuba, Venus, Teddy and Cones. The image resolutions are 384x288, 434x383, 450x375 and 450x375.

Side-view stereo image pairs of infield sorghum were collected using our autonomous phenotyping system (Bao et al. 2014). Image size was 1624x1224 pixels. Disparity ranges from 0 to 100. Stalk diameters of 26 plants were manually measured and the measuring spots were marked so that the stereo camera could see. These plants were different varieties at different growth stages. There were 12 plants specially planted with a larger spacing in order to expose more details in the images since agronomically sorghum plants are planted with a high density. The stalk sections were manually extracted from the images and the diameters were computed using the disparity map and camera parameters. The computed results were compared to the hand measurements.

3. Experimental results

3.1. Experimental settings

The parameters were set to $\{\boldsymbol{\beta}, \boldsymbol{\epsilon}, \boldsymbol{\theta_{col}}, \boldsymbol{\theta_{grad}}\} = \{\mathbf{0.9}, \mathbf{10}, \mathbf{10}, \mathbf{2}\}$ and window size was 35x35, following the original PMS algorithm. However, it was observed in our experiment that L2 norm for RGB color distance produced less speckle noise than L1 norm. Therefore L2 norm was used in our implementation. Gradient was computed using central difference on the grayscale image. Texture memory was used to store color image and gradient image. The pixel intensity was linearly interpolated for sub-pixel accuracy. Our algorithm was run on a NVIDIA Tesla K20 GPU.

3.2. Middlebury stereo evaluation for accuracy

The accuracy was determined by average percent of bad pixels compared to the ground truth of all 4 datasets. The ground truth was obtained by structured light technology. Figure 1 illustrates how the percentage drops after each iteration with and without post-processing. The curve of without post-processing shows 10% of disparities were correctly estimated in the first iteration and 55% in the second iteration. The rate of change slowed down to zero after the second iteration until convergence. Our post-processing was able to accelerate disparity propagation for the first two iterations and it was effective to reduce bad pixels at both occluded regions and textureless regions. To demonstrate this, the results with and without post-processing from the first five iterations were provided against the ground truth in Figure 2. The textureless region at top right corner of Teddy was not able to be reconstructed correctly without post-processing in Figure 3. Whereas, post-processing fully recovered that area with only two main iterations in Figure 4.

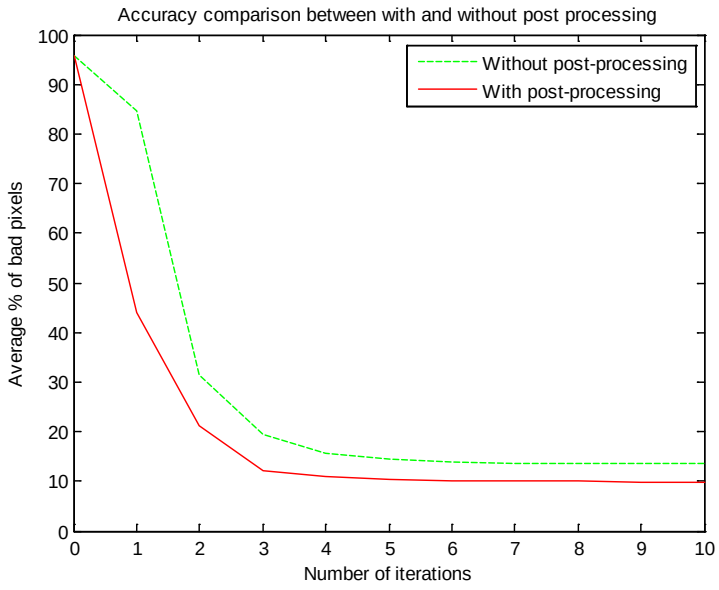

Figure 1. Accuracy comparison with and without post-processing.

Figure 2. Teddy left image and ground truth.

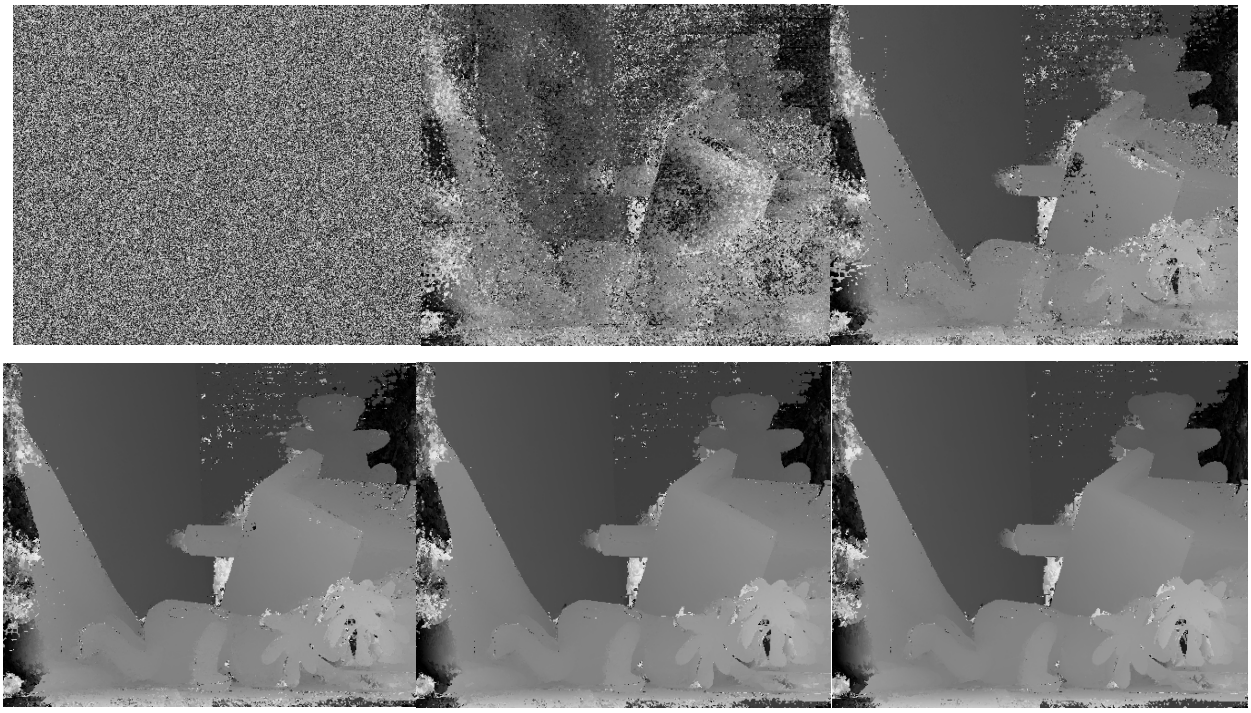

Figure 3. Random disparity map and results without post-processing for the first five iterations.

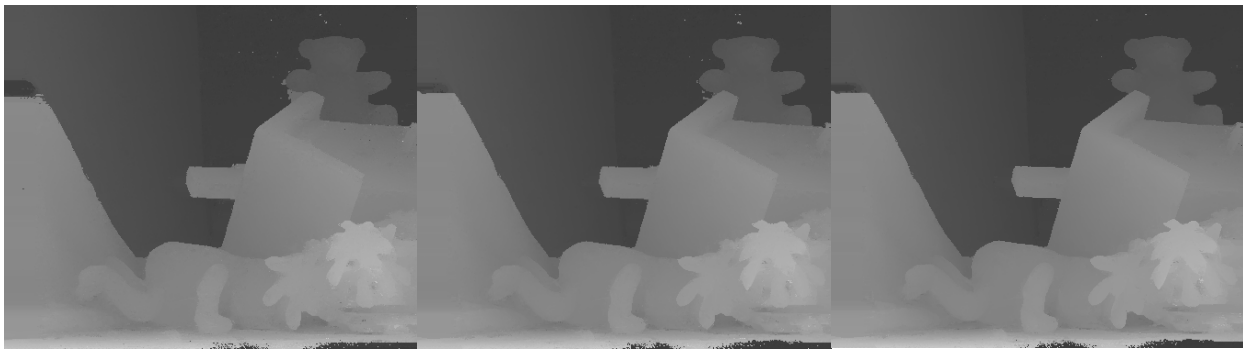

Figure 4. Random disparity map and results with post-processing for the first five iterations.

In comparison to the original PMS, only 6 iterations with post-processing were required to improve the original's ranking. Figure 8 shows the ranking and accuracy on each dataset. The evaluation was based on 0.5 error threshold for sub-pixel accuracy. Note that our GPU-based PMS provided higher average ranking. However, for Venus, Teddy and Cones, the scores were slightly behind the original. Our result on Tsukuba was substantially improved in Figure 7. The improvement mostly came from textureless regions in the background thanks to our textureless region recovery step in post-processing.

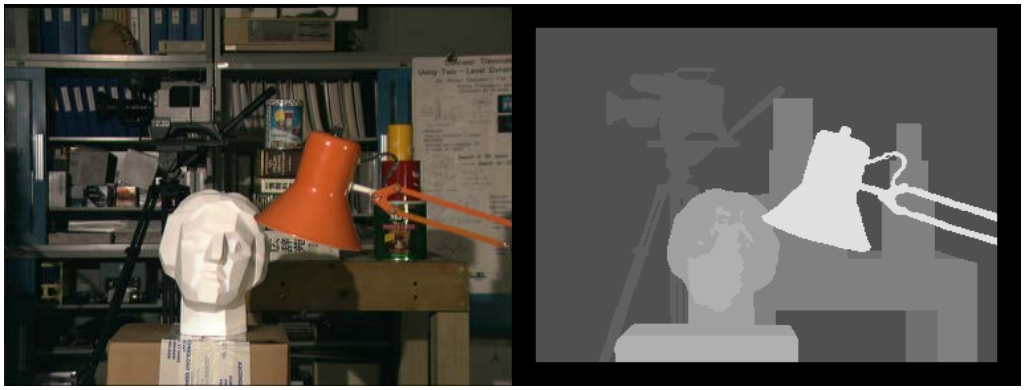

Figure 5. Tsukuba left image and disparity ground truth.

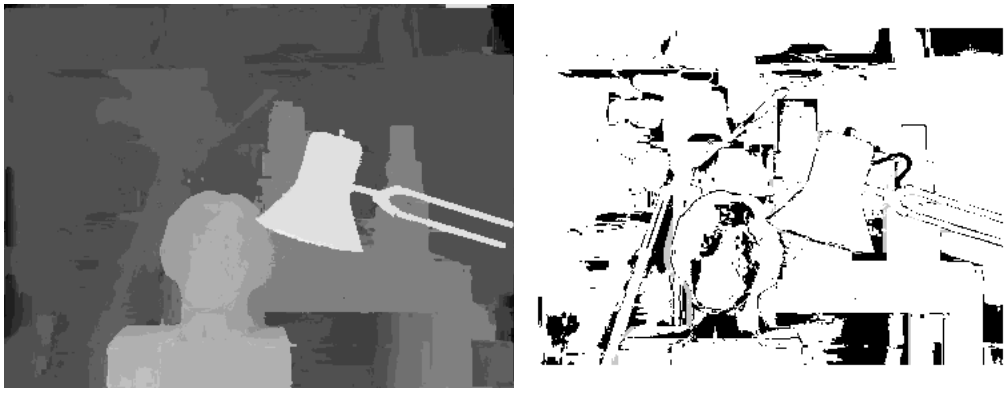

Figure 6. Original PMS result and error map (absolute disparity error > 0.5).

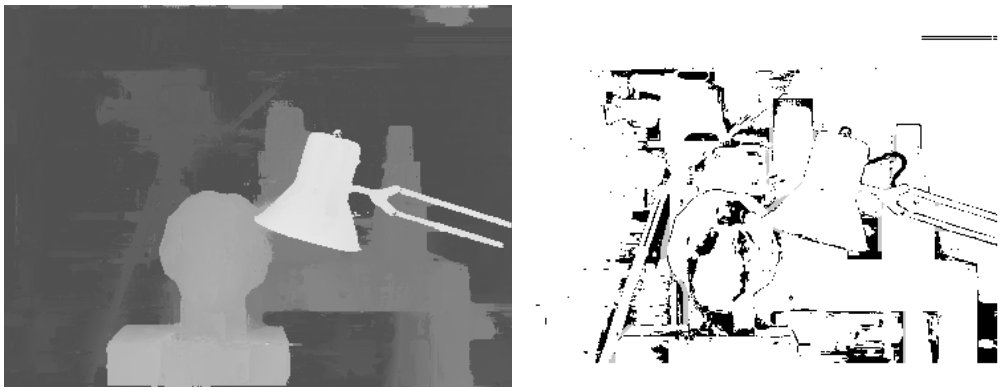

Figure 7. Our result and error map (absolute disparity error > 0.5).

Error Threshold = 0.5 (Error Threshold...)		Sort by nonocc				Sort by all				Sort by disc				
Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all	Venus disc	Teddy ground truth nonocc	Teddy all	Teddy disc	Cones ground truth nonocc	Cones all	Cones disc	Average Percent Bad Pixels
GC+LSL [136]	5.5	5.04 2	5.56 2	14.0 12	0.66 5	0.88 5	5.82 7	**4.20** 1	7.12 2	12.9 3	3.77 7	9.16 8	10.4 12	6.63
PM-PM [154]	7.8	7.16 12	7.66 9	15.1 22	0.58 3	0.79 2	4.67 4	5.21 5	11.9 10	15.9 7	3.51 6	8.86 6	9.58 8	7.58
PM-Huber [125]	8.0	7.12 10	7.80 12	13.7 10	1.00 11	1.40 12	7.80 18	5.53 7	9.36 4	15.9 8	**2.70** 1	7.90 2	**7.77** 1	7.33
ARAP [143]	8.2	7.17 13	7.67 10	16.0 30	0.64 4	0.87 4	6.17 8	5.52 6	10.7 7	15.6 6	3.00 3	8.55 4	8.35 3	7.52
SubPixSearch [109]	9.1	5.60 3	6.23 4	9.46 3	1.07 13	1.64 13	7.36 14	6.71 10	11.0 8	16.9 10	4.02 11	9.76 9	10.3 11	7.51
PMF [119]	12.1	11.0 39	11.4 36	16.0 31	0.72 7	0.92 6	5.27 6	4.45 3	9.44 6	13.7 4	2.89 2	8.31 3	8.22 2	7.69
SegAggr [145]	15.6	12.4 59	12.9 53	17.3 52	**0.28** 1	**0.41** 1	**2.09** 1	4.33 2	7.80 3	12.0 2	3.06 4	8.63 5	8.62 4	7.49
LAMC-DSM [123]	17.2	9.34 29	10.1 30	13.5 9	1.48 17	2.10 16	8.19 23	7.29 12	14.6 19	19.0 14	4.00 10	11.0 18	9.79 9	9.20
PM-Forest [156]	19.0	11.1 41	11.8 43	17.3 53	3.11 31	3.14 23	4.57 3	4.95 4	**5.45** 1	**11.3** 1	5.11 22	**6.35** 1	9.31 5	7.80
PMBP [113]	19.1	11.9 52	12.3 48	17.8 61	0.85 9	1.10 7	6.45 10	5.60 8	12.0 11	15.5 5	3.48 5	8.88 7	9.41 6	8.77
RealtimeEDP [152]	20.8	7.81 22	8.79 22	16.5 45	1.76 19	2.35 19	7.75 16	8.69 15	15.2 23	21.5 19	4.41 13	11.1 20	11.7 17	9.80
SNCC+AM [101]	24.0	9.96 33	10.4 31	19.7 80	0.76 8	1.11 8	6.23 9	8.70 16	14.3 18	23.2 29	4.47 15	11.1 19	12.4 22	10.2
YOUR METHOD	25.6	8.92 28	9.60 24	16.7 47	2.61 25	3.16 24	13.7 87	7.03 11	13.7 14	18.5 12	4.18 12	9.79 10	11.3 15	9.92
PatchMatch [96]	27.7	15.0 74	15.4 73	20.3 90	1.00 12	1.34 11	7.75 16	5.66 9	11.8 9	16.5 9	3.80 8	10.2 11	10.2 10	9.91

Figure 8. Algorithm ranking on Middlebury stereo evaluation (10 iterations with post-processing).

3.3. Performance analysis on Middlebury datasets

PMS can be computed in $O(WHD)$ time where W is image width, H image height and D window width. It is not related to disparity range unlike many other local methods which require building cost volumes with height equal to the number of possible disparities.

Since not all of the 4 datasets have the same resolution, Figure 9 shows the elapsed time for all 4 datasets after each iteration. The rate of change is the highest for the first iteration and second highest for the second iteration. The following iterations take about the same amount of time. This indicates that most of work was done within the first 2 iterations. For 6 iterations, the average processing time is about 15 seconds whereas 1 minute was reported for the original PMS. Our post-processing took several seconds on average.

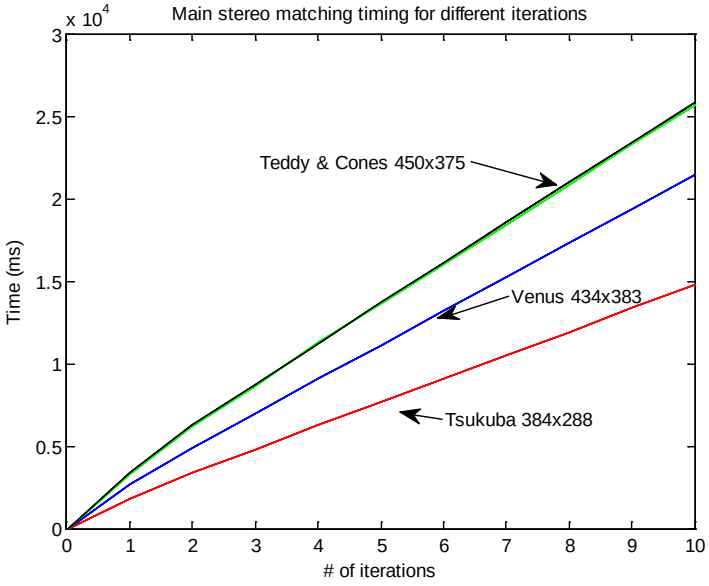

Figure 9. Performance evaluation on Middlebury datasets.

3.4. Evaluation on our sorghum datasets

Stalk diameter was estimated by first choosing two pixels (x, y) and (x', y') on the stalk edges in the image such that they could construct a perpendicular line cross the stalk as shown in Figure 10. Then these two points were projected back to 3D space using disparity value on the stalk between them and stereo camera parameters. The distance between the two 3D points served as the stalk diameter estimation.

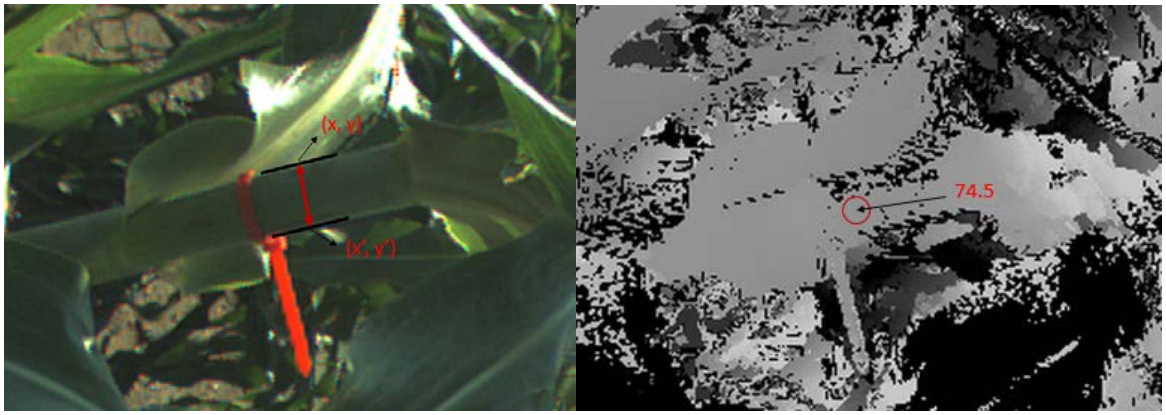

Figure 10. Stalk diameter estimation.

Figure 11 show the scatter plot and regression line of hand measurement and machine estimation for 26

different sorghum stalks. The correlation coefficient was 0.9814. Therefore, the machine estimation and hand measurement were highly positively linearly related.

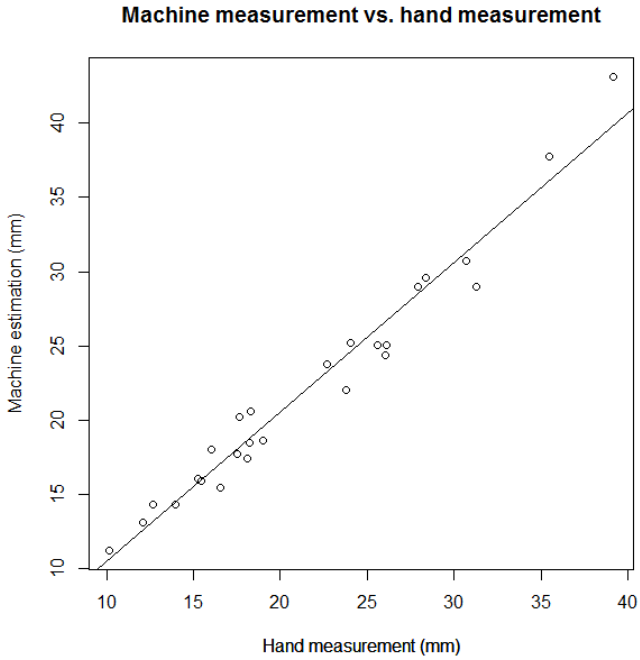

Figure 11. Hand measurement vs. machine estimation for stalk diameter.

The reconstruction quality depended on the complexity of the scene. For isolated plants, there was less occlusion, the reconstruction was visually recognizable and details were well preserved. However, as the plant density increased, occlusion and color ambiguity reduced the stereo matching accuracy. Figure 12-15 demonstrated how the scene complexity affects the stereo matching output.

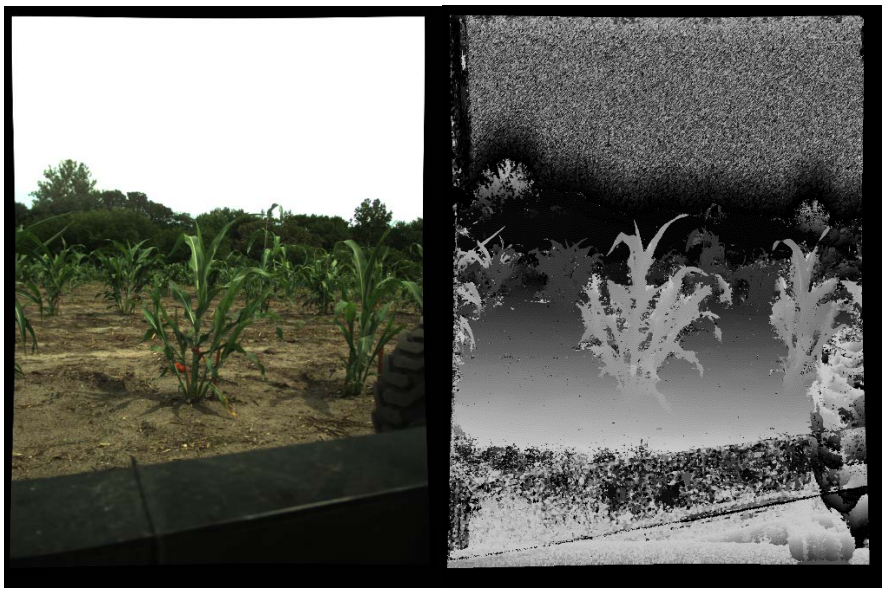

Figure 12. Color image and disparity map for isolated short sorghums.

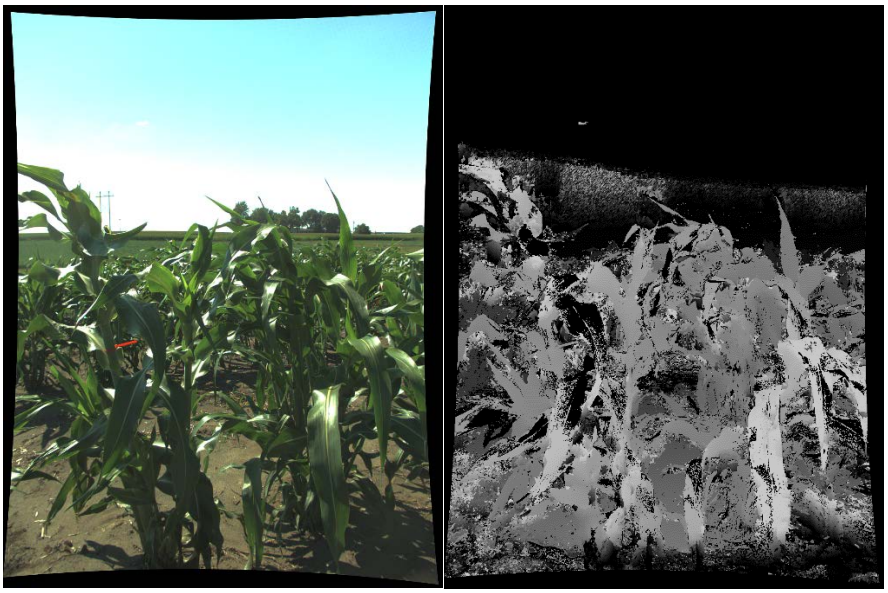

Figure 13. Color image and disparity map for isolated tall sorghums.

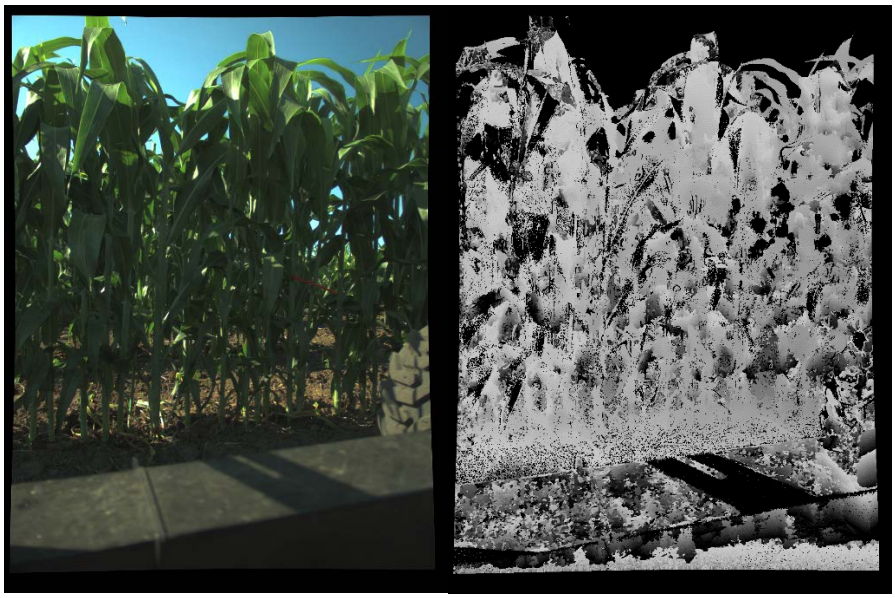

Figure 14. Color image and disparity map for densely populated short sorghums.

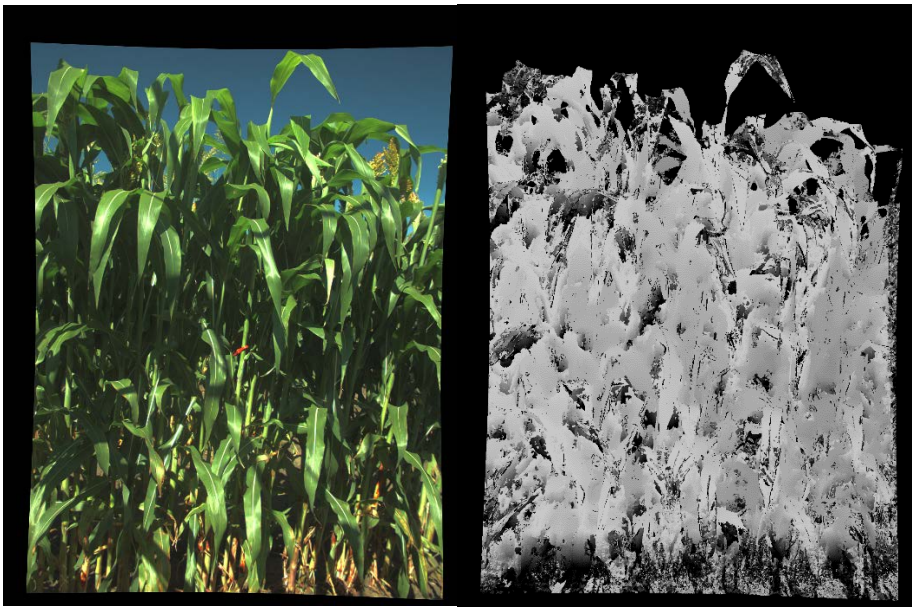

Figure 15. Color image and disparity map for densely populated tall sorghums.

4. Conclusion

This paper presented a GPU-based parallelization of PatchMatch Stereo. Our algorithm improved the original's sub-pixel accuracy on Middlebury stereo benchmark 4 times faster. This algorithm is well suited for reconstructing 3D plant model for infield biomass sorghum phenotyping in a high-throughput fashion because of its ability to handle highly slanted surfaces. Stalk estimation was highly correlated to hand measurement. Future work will be to explore CUDA optimization and reduce unnecessary computation to further improve performance.

Acknowledgements

The authors would like to thank USDA/DOE for their financial support under the project Genetic Architecture of Sorghum Biomass Yield Component Traits Identified Using High-Throughput Field-Based Phenotyping Technologies.

References

Bao, Y., Nakami, A., & Tang, L. (2014). Development of a Field Robotic Phenotyping System for Sorghum Biomass Yield Component Traits Characterization. ASABE Paper No. 141901199. Montreal, Quebec, Canada.: ASABE.

Besse, F., Rother, C., Fitzgibbon, A., & Kautz, J. (2014). Pmbp: Patchmatch belief propagation for correspondence field estimation. *International Journal of Computer Vision*, *110*(1), 2-13.

Bleyer, M., Rhemann, C., & Rother, C. (2011, August). PatchMatch Stereo-Stereo Matching with Slanted Support Windows. In *BMVC* (Vol. 11, pp. 1-11).

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *23*(11), 1222-1239.

Heise, P., Klose, S., Jensen, B., & Knoll, A. (2013, December). Pm-huber: Patchmatch with huber regularization for stereo matching. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (pp. 2360-2367). IEEE.

Lati, R. N., Filin, S., & Eizenberg, H. (2013). Estimating plant growth parameters using an energy minimization-based stereovision model. *Computers and Electronics in Agriculture*, *98*, 260-271.

Middlebury Stereo Datasets. http://vision.middlebury.edu/stereo/data/.

Middlebury Stereo Evaluation. http://vision.middlebury.edu/stereo/eval/.

Pock, T., Schoenemann, T., Graber, G., Bischof, H., & Cremers, D. (2008). A convex formulation of continuous multi-label problems. In *Computer Vision–ECCV 2008* (pp. 792-805). Springer Berlin Heidelberg.

Ranftl, R., Gehrig, S., Pock, T., & Bischof, H. (2012, June). Pushing the limits of stereo using variational stereo estimation.

In *Intelligent Vehicles Symposium (IV), 2012 IEEE* (pp. 401-407). IEEE.

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

Rooney, W. L., Blumenthal, J., Bean, B., & Mullet, J. E. (2007). Designing sorghum as a dedicated bioenergy feedstock. *Biofuels, Bioproducts and Biorefining*, *1*(2), 147-157.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, *47*(1-3), 7-42.

Sun, J., Zheng, N. N., & Shum, H. Y. (2003). Stereo matching using belief propagation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,*25*(7), 787-800.

Taniai, T., Matsushita, Y., & Naemura, T. (2014, June). Graph cut based continuous stereo matching using locally shared labels. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (pp. 1613-1620). IEEE.

Yoon, K. J., & Kweon, I. S. (2005, June). Locally adaptive support-weight approach for visual correspondence search. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 2, pp. 924-931). IEEE.