

Fast Narrow-Baseline Stereo Matching Using CUDA Compatible GPUs

Tao Chen, Yiguang Liu^(✉), Jie Li, and Pengfei Wu

Vision and Image Processing Laboratory, College of Computer Science,
Sichuan University, Chengdu 610065, People's Republic of China
liuyg@scu.edu.cn

Abstract. The Phase Correlation(PC) method demonstrates high robustness and accuracy for measuring the very subtle disparities from stereo image pairs, where the baseline (or the base-to-height ratio) is unconventionally narrowed. However, this method remains inherently computationally expensive. In this paper, an adaptive PC based stereo matching method is proposed, aiming to achieve higher speed and better stereo quality compared to the existing methods, while also preserving the quality of PC. Improvement was achieved both algorithmically and architecturally, via carefully dividing the computing tasks among multiprocessors of the GPUs under a novel global-pixel correlation framework. Experimental results on our hardware settings show that the method achieves as high as 64× and 24× speedup compared to single threaded and multi-threaded implementation running on a multi-core CPU system, respectively.

Keywords: Stereo matching · Narrow baseline · Phase Correlation(PC) · CUDA

1 Introduction

Stereo vision is an attractive topic in the realm of computer vision, while stereo matching [3], targeting at extracting disparity information from a pair of images, is the corner stone of the entire task. Though wide-baseline stereo matching [6], [10] is commonly used because of its high estimation accuracy, its narrow-baseline counterpart is, in contrast, much more challenging due to narrow triangulation. It is nevertheless worth addressing because it can alleviate the occlusion problem (Fig. 1), while requiring a smaller disparity search range.

To attain high accuracy, PC was integrated into narrow-baseline stereo matching [1], [8], [11]. The PC based method is more robust in illumination changes than simple correlation function based matching, while able to measure the very subtle disparities that result from low base-to-height ratio (e.g., less than 0.1). Thus potentially allowing applications, such as digital elevation models (DEMs), to be derived from images that previously might not have been considered suitable for stereo vision. However, the PC based method remains inherently computationally expensive. With the increasing of image resolutions, the computational time may even become prohibited.

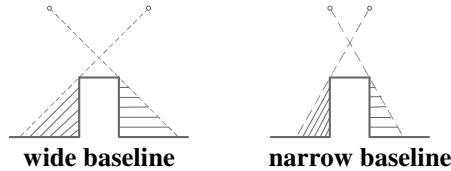


Fig. 1. Differences of occlusion zones. Occlusions in the left image of the stereo pair are signaled by horizontal lines, occlusions in the right image of the stereo pair are signaled by slanted lines. We see that in a wide-baseline system, the occlusion differences are much more critical than in a narrow-baseline system.

To attain fast speed, stereo on graphics processing units (GPUs) is an attractive trend. Specially, CUDA (Compute Unified Device Architecture) is a modern GPU architecture designed for writing and running general-purpose applications on the nVIDIA GPUs. Utilizing the horsepower of massive parallel processors, CUDA is effective to accelerate stereo algorithms by exploiting their potential parallelism. Several recent methods have reached fast speed on CUDA. Both Sarala [2] and Zhu [12] implemented a Normalized Cross Correlation (NCC) approach and obtained significant improvement in terms of computational efficiency. Their approach fails to maintain the matching quality, however, if the baseline is unconventionally narrowed [8]. Kentaro [7] suggested to perform parallel PC within a single image-block pair. It offers great speed improvement, only if the image-block size is sufficient large. Unfortunately, this is not the case in stereo matching. In most applications, the size of image-block is fairly small while the number of blocks is very large. Due to the limitation of memory bandwidth, this approach does not have a noticeable effect on the run-times. Similarly, the PC method accelerated for image fusion (e.g. Falk [9]), is also not suitable for stereo matching.

In this paper, the PC method is re-examined, and a novel stereo matching framework based on CUDA especially optimized for narrow-baseline scenario is proposed. Using both algorithmic and architectural means, we carefully divide the task among multiprocessors of the GPUs and exploit its texture memory. Furthermore, we compare our results against single and multi-threaded CPU based implementation. Experimental results demonstrate the significant speedup of our approach. The remainder of this paper is organized as follows. Section 2 briefly introduces the PC algorithm for narrow-baseline stereo matching. Section 3 gives a detailed description of the proposed framework of CUDA PC. The algorithm’s performance are analyzed in detail in Section 4. Finally, we conclude in Section 5.

2 PC for Narrow-Baseline Stereo

Based on the well-known Fourier shift property [5], the PC method is developed to estimate the translation displacement between two images. Consider two

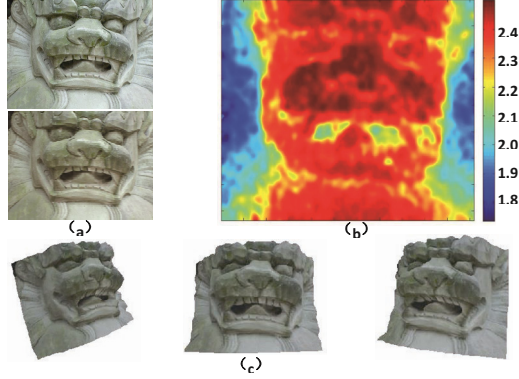


Fig. 2. (a) Stereo pair of the stone lion with B/H ratio = 0.001. (b) Disparity map after median filtering. (c) Textured reconstruction result.

images $I_1(x, y)$, $I_2(x, y)$ that are offset by simple translation a and b such that

$$I_2(x, y) = I_1(x - a, y - b), \quad (1)$$

their Fourier transforms (FTs) \hat{I}_1 and \hat{I}_2 are related by the Fourier shift property, such that

$$\hat{I}_2(u, v) = \hat{I}_1(u, v)e^{-2\pi i(au + bv)}. \quad (2)$$

This can be re-written as

$$P(u, v) = \frac{\hat{I}_1(u, v)\hat{I}_2^*(u, v)}{|\hat{I}_1(u, v)\hat{I}_2^*(u, v)|} = e^{-2\pi i(ua + vb)}, \quad (3)$$

Where \hat{I}_2^* denotes the complex conjugate of \hat{I}_2 , and $P(u, v)$ is referred to as the normalized cross power spectrum (NCPS) of the two signals. There are two possible ways of solving (3) for (a, b) [4]. One is to work in the Fourier domain directly. Employing singular value decomposition (SVD) and robust 2-D fitting algorithm, [8] solve the phase difference within the Fourier domain and achieved sub-pixel disparity measurement. The second possible approach is to first transform the NCPS back into spatial domain. It is then a simple matter to determine (a, b) , since from (3) the result is $\delta(x - a, y - b)$ which is a Dirac delta function centered at (a, b) . The sub-pixel translation can be estimated using interpolation-based approach after determining the max peak of the correlation surface on the integer-accuracy grid [4].

Both the Fourier domain and the spatial domain methods have been reported to achieved up to 1/20th pixel accuracy. The latter is employed in our implementation due to its relative efficiency and simplicity. As expected, this method is capable of precisely and directly measuring the fractional disparities that result from unconventionally narrow baseline images, images which would otherwise not be considered suitable for conventional stereo processing. Fig. 2(a) shows the resulting stereo pair. Due to the very low B/H ratio, the images are geometrically very similar and thus enjoy high correspondence with little occlusion. This work focuses on tackling the challenge of fast stereo matching for narrow-baseline scenario, as described in the next section.

3 CUDA PC

The CUDA environment exposes the single instruction multiple data (SIMD) architecture of the GPUs by enabling data-parallel computation for millions of threads. These threads are organized into a grid of thread-blocks. The highest performance is achieved when the threads avoid divergence and perform the same operation on their data elements. Overall, the prime computational challenge

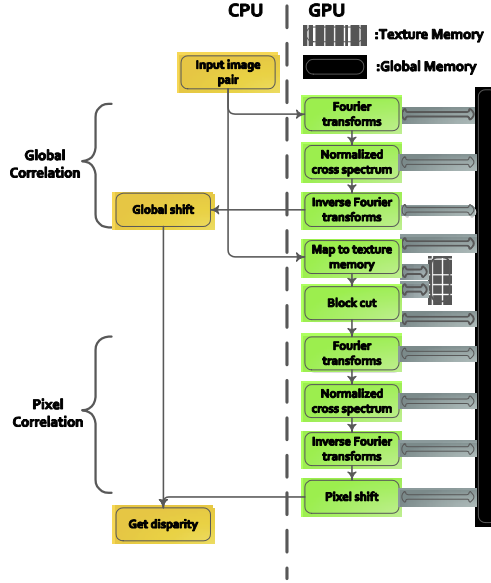


Fig. 3. Workflow of the proposed algorithm

in disparity estimation lies in the PC calculation for a huge number of image block pairs, aiming to find the translation displacement of each pixel. In this implementation, we use the CUFFT function of CUDA library for 2D FTs. The global correlation is employed to co-register the right image to the left via shift change. Thus allowing a smaller image-block size for pixel correlation, which is helpful to improve the estimation accuracy while reducing the computational cost. A parallel block-cut procedure is designed to make the PC calculation for each pixel independent to other pixels. And then the translation displacement of each pixel can be computed in parallel. Fig. 3 shows the principle workflow of the proposed algorithm.

3.1 Global Correlation.

For narrow-baseline images, it is worth to note that the disparities are likely to be in a small range, which is less than the size of image-block. Thus the process can be efficient as no search region is ever required. The only question remains of how to locate the corresponding block in the right image for each block in the left. We therefore employ a global correlation procedure to estimate the global translation relationship. Although, we assume that the images have been acquired with or resampled close to epipolar geometry.

Given image resolution $W \times H$, the global correlation consists of four steps: FTs for input images, NCPS calculation, inverse FTs and global shift calculation. We use the highly optimized CUFFT function of the CUDA for FTs and inverse FTs. NCPS calculation is parallelized with $W \times H$ thread. The threads are organized into a 2D grid and the thread-block size is 32×32 . Each thread takes care of computing an independent coefficient at a given frequency component. Finally, the so-called phase correlation surface (PCS) can be obtained as $p(x, y)$. In this step, an integer-level estimation is sufficient, so the global shift parameters (x_g, y_g) is given by

$$(x_g, y_g) = \underset{x, y}{\operatorname{argmax}} |p(x, y)|, \quad (4)$$

which can be linear processed on the CPU.

3.2 Pixel Correlation

The difference between global correlation and pixel correlation is subtle, but essentially different, especially at the parallel level.

The novel block-cut procedure plays a crucial role for improving the parallel level, which is also parallelized on the GPU, as depicted in Fig. 4. A 2D grid with $W \times H$ threads is created for each image, and every $B \times B$ threads takes care of copying a $B \times B$ section centered at a pixel to an individual memory space, where $B(\text{pixel})$ means the side length of the image-block. Considering global shift (x_g, y_g) , an image-block centered at (x, y) in the left image is then paired with a image-block taken at $(x + x_g, y + y_g)$ in the right. Also, we use texture memory for accessing images due to the fact that our copy procedure exhibit 2D locality, which provides a great scope for texture caching. Thus before block-cut, two input images are mapped to the texture memory space previously.

Then, N_{par} independent block pairs are prepared for the following FTs, NCPS calculation, and inverse FTs procedures, with $N_{par} = \lfloor \frac{W \times H}{B^2} \rfloor$. Finally, we get

N_{par} PCSs at a time. Unlike global correlation, the NCP is approximated as [4]

$$\dot{p}(x, y) \simeq \frac{\alpha}{B^2} \frac{\sin \pi(x + \delta_1)}{\sin \frac{\pi}{B}(x + \delta_1)} \frac{\sin \pi(y + \delta_2)}{\sin \frac{\pi}{B}(y + \delta_2)}. \quad (5)$$

Where $\alpha < 1$, the peak position of the function corresponds to the global displacement between the two images, and the α corresponds to the degree of correlation between the two images. After locating the main peak at some coordinates

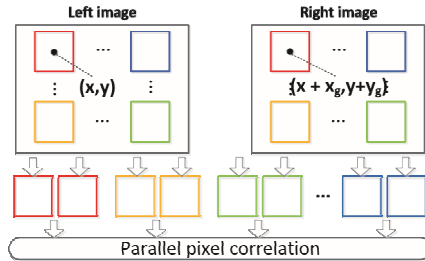


Fig. 4. block-cut procedure

(x_m, y_m) and two side-peaks at (x_s, y_m) and (x_m, y_s) where $x_s = (x_m \pm 1) \% B$ and $y_s = (y_m \pm 1) \% B$, the sub-pixel offset $(\Delta x_p, \Delta y_p)$ are decided by linear weighting such that

$$\Delta x_p = \frac{p(x_s, y_m)}{p(x_s, y_m) \pm p(x_m, y_m)}, \Delta y_p = \frac{p(x_m, y_s)}{p(x_m, y_s) \pm p(x_m, y_m)}. \quad (6)$$

For a single PCS, the major computing task lies in searching for the main peak, which is a linear process and shows greater advantage when processed on CPU. In spite of this, we distribute the computing task to GPU. The reason is that a N_{par} times parallelism not only can offset the inferiority on GPU, but also performs far more efficiently than a sequential process on CPU. Afterwards, the disparity (d_x, d_y) is decided such that

$$d_x = x_g + x_m + \Delta x_p, d_y = y_g + y_m + \Delta y_p. \quad (7)$$

Repeat the above steps, we extract the disparity information for each pixel. In addition, small disparity outliers are filtered using a median filter, resulting in the disparity map shown in Fig. 2(b) and the absolute final reconstruction result in Fig. 2(c).

4 Experiments

The hardware environment is based on Intel CoreI5-3470 CPU @ 3.2 GHz and NVIDIA GTX770 graphics card. In order to evaluate the efficiency, the proposed method was compared against both the single-threaded and 4-threaded CPU implementation. We have used stereo pairs of different sizes with image-block size set to 16×16 and 32×32 respectively. The timing results are summarized in table 1.

Our results demonstrate that the proposed method outperforms the CPU based implementation by a huge factor. For a stereo pair of size 1024×768 with image-block size set to 16×16 , the new method takes 0.6 seconds, bringing an impressive 64× speedup with respect to the single-threaded CPU implementation. Even compared to the 4-threaded CPU implementation, our method can

Table 1. Timing results for stereo matching

Image Size	CPU-1 thread(s)		CPU-4 threads(s)		This method(s)	
	16×16	32×32	16×16	32×32	16×16	32×32
320×240	3.2	5.5	1.1	2.2	0.2	0.8
640×480	14.0	29.5	5.0	10.4	0.3	1.6
1024×768	38.6	78.3	14.9	29.5	0.6	2.7
1600×1200	91.0	193.0	30.7	75.5	1.5	4.8
2048×1536	150.9	345.1	58.4	121.1	2.6	8.0
3024×2016	332.7	756.8	126.4	276.5	5.6	19.4

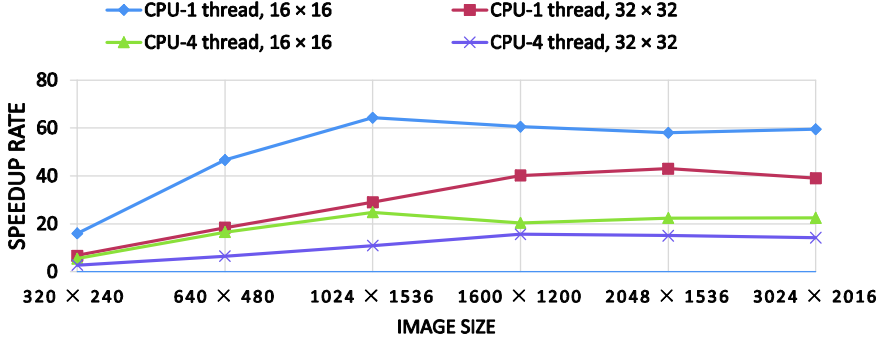


Fig. 5. Plot of the speedups achieved compared to the two CPU implementations

perform 24 times faster. Fig. 5 shows a detailed plot of the speedups achieved compared to the two CPU implementations. Not surprisingly, the speedup ratio increases along with the increase in image size. Theoretically, it can be explained that a larger image size means more image-block pairs parallelized at a time, with the relationship of $N_{par} = \lfloor \frac{W \times H}{B^2} \rfloor$. Then the speedup ratio comes to fluctuate in a small scope, corresponding to the full capacity for our NVIDIA GTX770 GPU. It is also obvious that reducing the image-block size helps to increase the parallel number. Nevertheless, the interval [16,32] for the side length of image-block has been empirically proved to be suitable for the narrow-baseline scenario, with regard to matching accuracy.

Here, we see that the excellent performance on efficiency of our method lies in a high-level parallelization, not in a common speed-accuracy tradeoff. Hence the quality of PC in the narrow-baseline scenario is well maintained. With high resolution stereo image pairs, our method has the ability to provide disparity information instantly with high accuracy.

5 Conclusion

In this paper, a novel stereo matching framework based on CUDA especially optimized for narrow-baseline scenario is proposed. We employed global correlation to improve the estimation accuracy while reducing the computational cost. Via a crucial block-cut procedure, we carefully divide the task among multiprocessors of the GPUs in a high parallel level. Texture memory is also used, providing a great scope for accessing images. Experimental results demonstrate that the proposed method outperforms the CPU based implementation by a huge factor, which is capable of instantly and precisely measuring the fractional disparities in narrow-baseline scenario.

Acknowledgments. The authors thank the editors and anonymous reviewers for their insights. This work is supported by NSFC under grants 61173182 and 613111154, funding from Sichuan Province (2014HH0048, 2014HH0025) and the Science and Technology Innovation seedling project of Sichuan (2014-033, 2014034, 2015-046, 2015-095).

References

1. Arai, T., Iwasaki, A.: Fine image matching for narrow baseline stereovision. In: 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 2336–2339. IEEE (2012)
2. Arunagiri, S., Jaloma, J.: Parallel gpgpu stereo matching with an energy-efficient cost function based on normalized cross correlation. In: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, pp. 86550X–86550X (2013)
3. Brown, M.Z., Burschka, D., Hager, G.D.: Advances in computational stereo. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 993–1008 (2003)
4. Foroosh, H., Zerubia, J.B., Berthod, M.: Extension of phase correlation to subpixel registration. IEEE Transactions on Image Processing, **11**(3), 188–200 (2002)
5. Kuglin, C.: The phase correlation image alignment method. In: Proc. Int. Conf. Cybernetics and Society, September 1975, pp. 163–165 (1975)
6. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vision Comput. **22**(10), 761–767 (2004)
7. Matsuo, K., Hamada, T., Miyoshi, M., Shibata, Y., Oguri, K.: Accelerating phase correlation functions using gpu and fpga. In: NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2009, pp. 433–438. IEEE (2009)
8. Morgan, G.L.K., Liu, J.G., Yan, H.: Precise subpixel disparity measurement from very narrow baseline stereo. IEEE Transactions on Geoscience and Remote Sensing **48**(9), 3424–3433 (2010)
9. Schubert, F., Mikolajczyk, K.: Benchmarking GPU-based phase correlation for homography-based registration of aerial imagery. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) CAIP 2013, Part II. LNCS, vol. 8048, pp. 83–90. Springer, Heidelberg (2013)
10. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to widebaseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(5), 815–830 (2010)
11. Yiguang, L., Chenhui, Z., Ronggang, H., Baofeng, D.: Rectification-free 3dimensional reconstruction method based on phase correlation for narrow baseline image pairs(无需图像矫正的高精度窄基线三维重建算法). Journal of University of Electronic Science and Technology of China **43**(2), 262–267 (2014)
12. Zunshang, Z., Zhen, G., Shengyi, C., Xiaoliang, S., Yang, S.: Research on cudabased image parallel dense matching. In: Chinese Automation Congress (CAC), 2013, pp. 482–486. IEEE (2013)

Advances in Image and Graphics Technologies
10th Chinese Conference, IGTA 2015, Beijing, China,
June 19-20, 2015, Proceedings
Tan, T.; Ruan, Q.; Wang, S.; Ma, H.; Di, K. (Eds.)
2015, XII, 456 p. 270 illus., Softcover
ISBN: 978-3-662-47790-8