Performance Analysis of Matching Cost for Stereo Matching with CUDA

Gwang-Soo Hong, Woong Hoe, and Byung-Gyu Kim

Dept. of Computer Engineering,
SunMoon University, A-san, Rep. of Korea
{gs.hong,hoewoong,bg.kim}@mpcl.sunmoon.ac.kr

Abstract. Stereo Imaging is a powerful technique for determining the distance to objects using a pairs of camera spaced apart. The extremely high computational requirements of stereo vision limit application to non realtime applications where high computational calculation is available. To overcome the limitation, we reported the general strategy for parallelization of dense matching methods with CUDA (Compute Unified Device Architecture) programming.

Keywords: Stereo vision, CUDA.

1 Introduction to Stereo Correspondence Algorithm

The basic structure of a stereo vision application is shown in figure 1. Two cameras are spaced apart by the baseline distance B. Each camera images the object, but from a slightly different angle. The distance to the object can then be computed by:

$$D = \frac{Bf}{d} \tag{1}$$

Where D is the distance to the object, B is the baseline distance between the stereo images, f is the focal length of the camera, and d is the disparity. d is the difference in location of the image of the object between the left and right images. Finding d by determining the correspondence between pixels in the stereo pair is the primary problem to be solved by this application. Our work computed d in units of pixels, which can be converted to distance units by multiplying by the pixel size. Also, this simple formula assumes idealized optics.

To decide cost measure function is important in stereo vision. This chapter will introduce some concepts that are common in stereo algorithms of today. Its purpose is to give an introduction to the area before looking closer into some modern stereo algorithms, and before presenting the implementation part of the project with CUDA programming. In order to find corresponding pixels in target and reference image, there is obviously need for pixel similarity measure. It is used the term as dissimilarity measure or matching cost, which increases as the similarity between two compared pixels decreases.

James J. (Jong Hyuk) Park et al. (eds.), *Computer Science and Its Applications*,
Lecture Notes in Electrical Engineering 330, DOI: 10.1007/978-3-662-45402-2_88

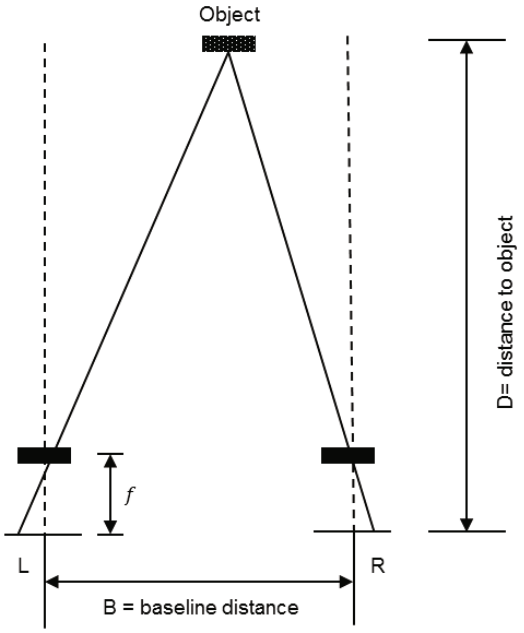


Fig. 1. The basic structure of a stereo vision

Matching cost is a function $C(x, y, d)$ of reference image coordinates and disparity. The cost function returns the value of dissimilarity for a coordinate (x, y, d) in the disparity space. The disparity space is made by the available image pixel coordinates and disparity search range. Generally, disparity search range has to be set manually and depends on the characteristics of the input image pair. The cost $C(x, y, d)$ is referred to as disparity space image (DSI). This is due to the fact that for each fixed integer disparity value , the function represents an image visualizing the cost for every pixel location.

Common measures that are used for pixel-wise comparison are absolute intensity difference (AD) [1], squared intensity difference (SD) [2] and absolute gradient difference (GRAD). The cost functions for pixel-wise comparison are written as:

$$C_{AD}(x, y, d) = |I_L(x, y) - I_R(x - d, y)| \tag{2}$$

$$C_{SD}(x, y, d) = |I_L(x, y) - I_R(x - d, y)|^2 \tag{3}$$

$$\begin{aligned} C_{GRAD}(x, y, d) = &| \bigtriangledown_x I_L(x, y) - \bigtriangledown_x I_R(x - d, y)| \\ &+| \bigtriangledown_y I_L(x, y) - \bigtriangledown_y I_R(x - d, y)| \end{aligned} \tag{4}$$

By extending the comparison to square window regions centered about the search and reference pixels, these measure are turned into sum of absolute intensity differences (SAD), sum of squared intensity differences (SSD) and sum of absolute gradient differences (SGRAD). The cost functions based for square comparison are defined as:

$$C_{SAD}(p,d) = \sum_{q \in N_q} (|I_L(q) - I_R(q-d)|) \tag{5}$$

$$C_{SSD}(p,d) = \sum_{q \in N_q} [I_L(q) - I_R(q-d)]^2 \tag{6}$$

$$\begin{aligned} C_{SGRAD}(p,d) = & \sum_{q \in N_p} | \bigtriangledown_x I_L(q) - \bigtriangledown_x I_R(q-d)| \\ & + \sum_{qinN_q} | \bigtriangledown_y I_L(q) - \bigtriangledown_y I_R(q-d)| \end{aligned} \tag{7}$$

Next following cost measurements relies on calculating at each position of the image under examination a correlation or distortion function that measures the degree of similarity or dissimilarity to a template sub image. Among the correlation/distortion functions proposed in literature, Normalized Cross-Correlation (NCC) [3] and Zero mean Normalized Cross-Correlation (ZNCC) [4] are widely used due to there robustness in template matching.

$$C_{NCC}(p,d) = \frac{\sum_{q \in N_p} I_L(q) \cdot I_R(q-d)}{\sqrt{\sum_{q \in N_p} I_L(q)^2 \cdot \sum_{q \in N_p} I_R(q-d)^2}} \tag{8}$$

$$C_{ZNCC}(p,d) = \frac{\sum_{q \in N_p} (I_L(q) - \overline{I}_L(p))(I_R(q-d) - \overline{I}_R(p-d))}{\sqrt{\sum_{q \in N_p} (I_L(q) - \overline{I}_L(p))^2 \sum_{q \in N_p} (I_R(q-d) - \overline{I}_R(p-d))^2}} \tag{9}$$

2 Experimental Results

We test out experimental results using middlebury stereo image pairs [5]. We consider only cost measure functions and used simple window aggregation and WTA(Winner-Take-All) optimization to test computational complexity [6].

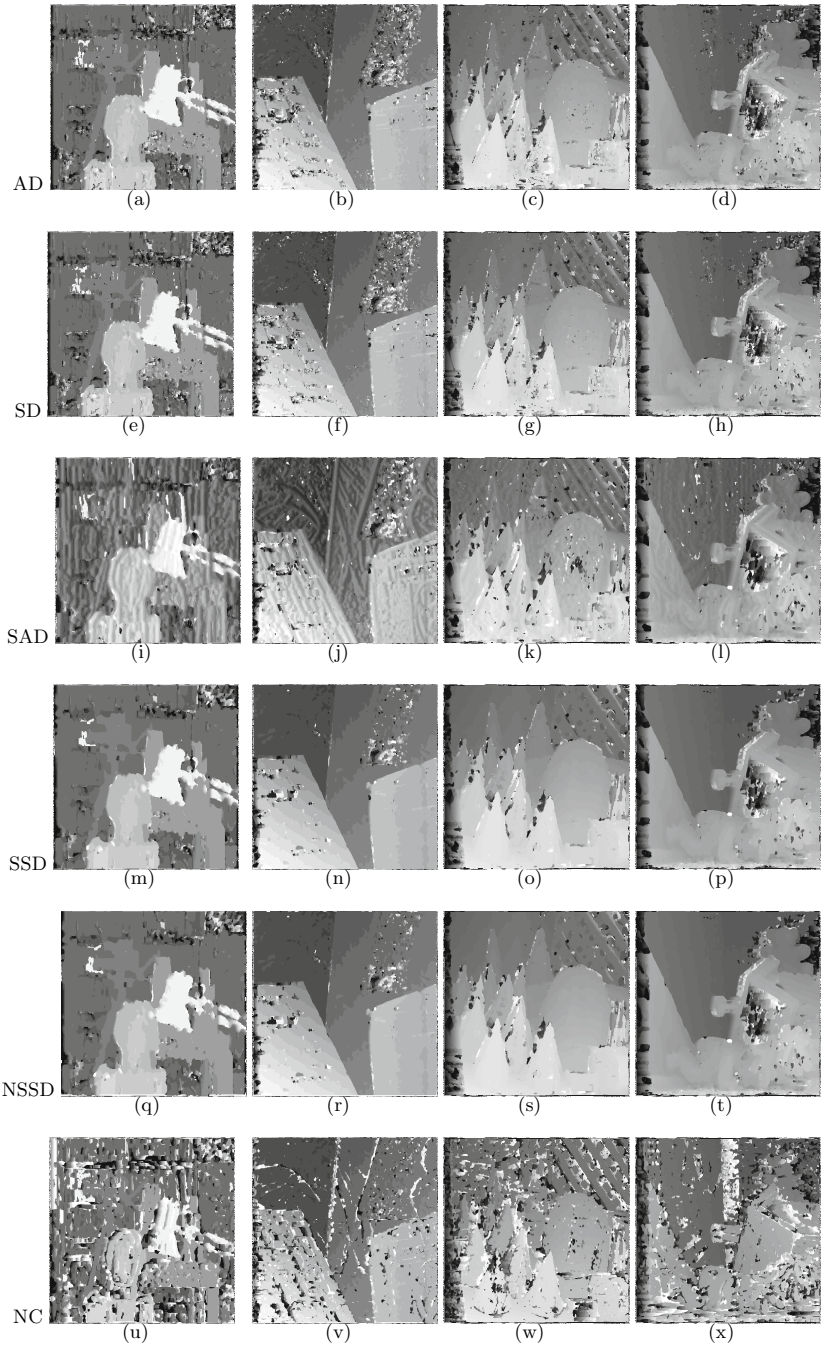


Fig. 2. The results of depth map: (a)(e)(i)(m)(q)(u): tsukuba image pair. (b)(f)(j)(n)(r)(v): venus image pair. (c)(g)(k)(o)(s)(w): cones image pair. (d)(h)(l)(p)(t)(x): teddy image pair.

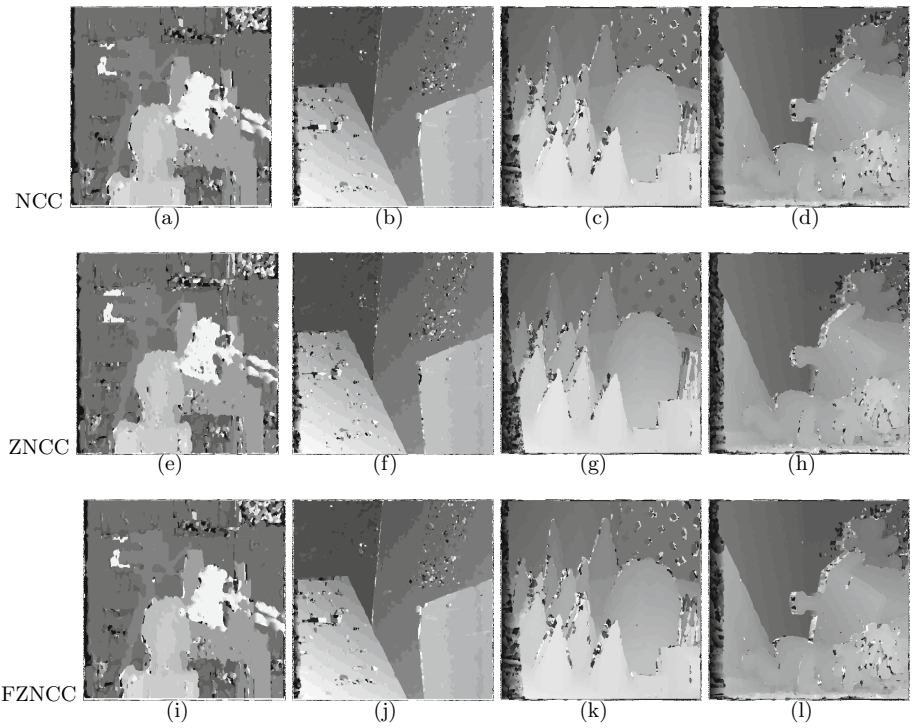
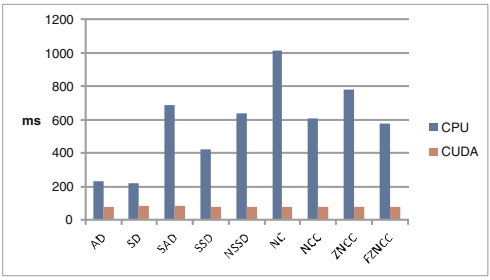


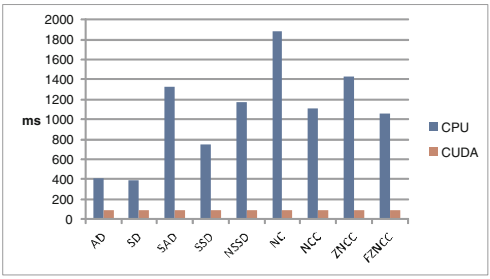
Fig. 3. The results of depth map: (a)(e): tsukuba image pair. (b)(f)(j): venus image pair. (c)(g)(k): cones image pair. (d)(h)(l): teddy image pair.

Figure 2,3 show the the results of depth map according to the cost functions. The cost functions based on pixel-wise show the worse results than the cost functions based on square window in subjective point of view. ZNCC and NC show batter results than the other cost functions in term of uniform region and depth discontinuities region.

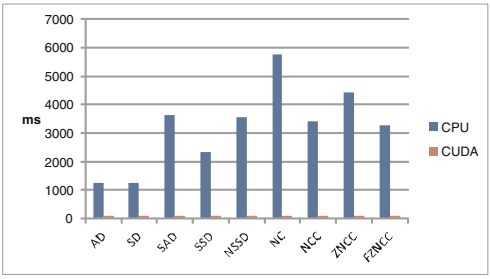
Figure 4 indicates comparison of the time-consuming performance between based on CPU and CUDA. By comparing to existing real-time stereo matching methods, the proposed method is evaluated in terms of speed. Computations of stereo matching based on CUDA coding is faster than based on CPU coding about from 4 times to 60 times. Since most time is consumed by memory allocation and memory copy, the results of time-consuming on CUDA coding are similar about 80ms regardless of cost functions. The part calculating matching cost is actually performed quickly by splitting the thread unit. The cost Functions based on square window have high computational complexity in CPU coding. In the CUDA coding, the cost functions based on square window make no differences in term of speed.



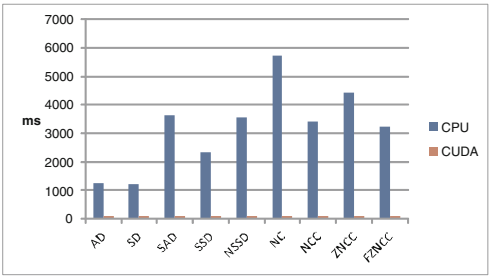
(a)



(b)



(c)



(d)

Fig. 4. Comparison of the time-consuming performance between based on CPU and CUDA: (a)tsukuba image pair. (b): venus image pair. (c): cones image pair. (d): teddy image pair.

3 Conclusion

We have reported the comparison performance between CPU and CUDA coding in stereo matching in terms of computational complexity. Computations of stereo matching based on CUDA coding is faster than based on CPU coding about from 4 times to 60 times. In the CUDA coding, pixel-wise cost functions as well as square window cost functions make depth map rapidly. Hence, to generate depth map is available near real-time in CUDA coding.

References

1. Kanade, T.: Development of a video-rate stereo machiine. In: Image Understanding Workshop, pp. 549–557 (1994)
2. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. International Journal Computer Vision 2(3), 283–310 (1989)
3. Hirschmuller, H., Innocent, P.R., Garibaldi, J.M.: Real-time correlation-based stereo vision with reduced border errors. International Journal of Computer Vision 47(1-3), 229–246 (2002)
4. Krattenthaler, W., Mayer, K.J., Zeiller, M.: Point correlation: a reduced-cost template matching technique. In: Proceedings of the IEEE International Conference on Image Processing, ICIP 1994, vol. 1. IEEE (1994)
5. `http://vision.middlebury.edu/stereo/`
6. Scharstein, D., Szeliski, R.: A Taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47(1), 7–42 (2002)