Real-Time High-Definition Stereo on GPGPU using Progressive Multi-Resolution Adaptive Windows

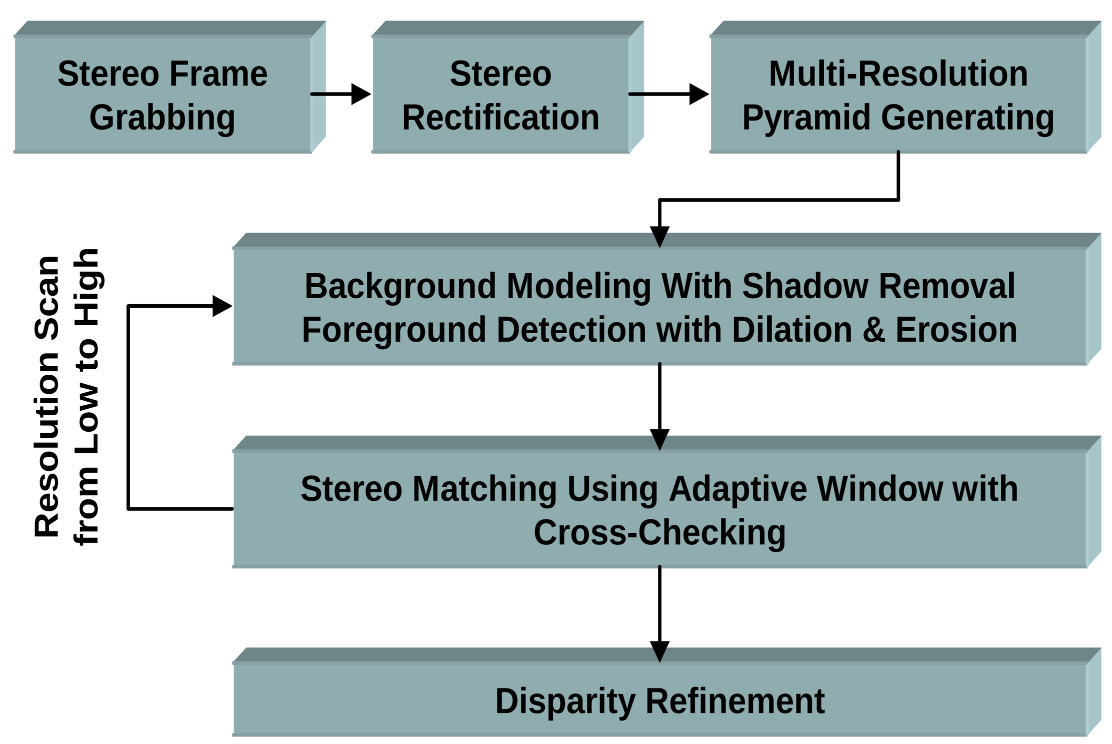
Yong Zhao, Gabriel Taubin. Brown University, Department of Electrical Science and Computer Engineering

{yong_zhao, taubin}@brown.edu

BROWN

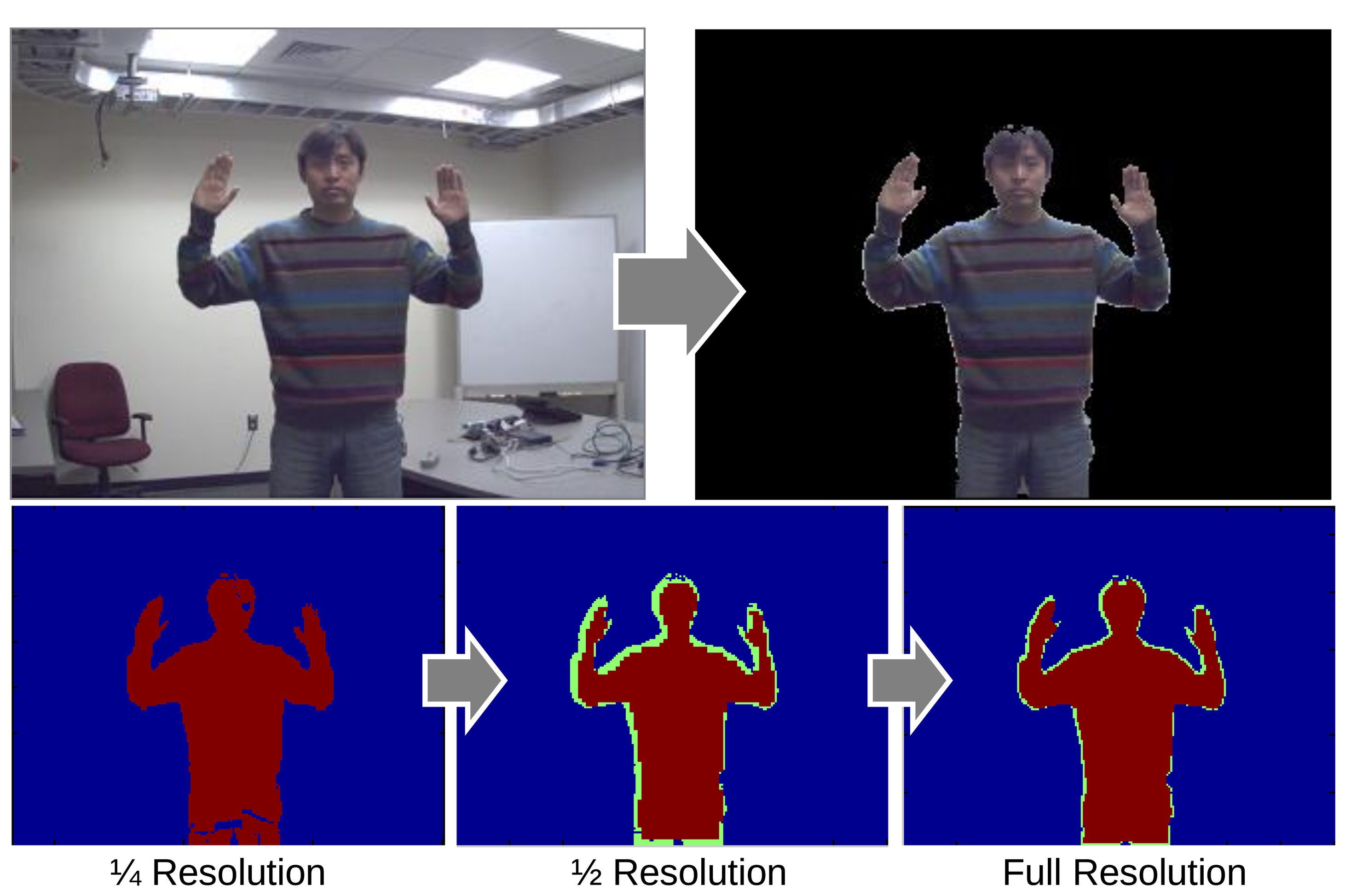
Overview

We introduce a new GPGPU-based real-time high resolution high frame rate dense stereo matching algorithm. The algorithm is based on a progressive multi-resolution pipeline which includes background modeling and dense matching with adaptive windows. For applications in which only moving objects are of interest, this approach effectively reduces the overall computational cost quite significantly, and preserves the high definition details. Running on an off-the-shelf Nvidia graphics card, our implementation achieves 7200M disparity evaluations per second. The current implementation achieves 60 fps stereo matching on 800x600 stereo video with a fine 250 pixels disparity range. In our implementation only the GPU is used for processing the video data. Therefore the CPU power is free to perform other tasks. We envision a number of potential applications such as real-time motion capture, as well as tracking, recognition and identification of moving objects in multi-camera networks.



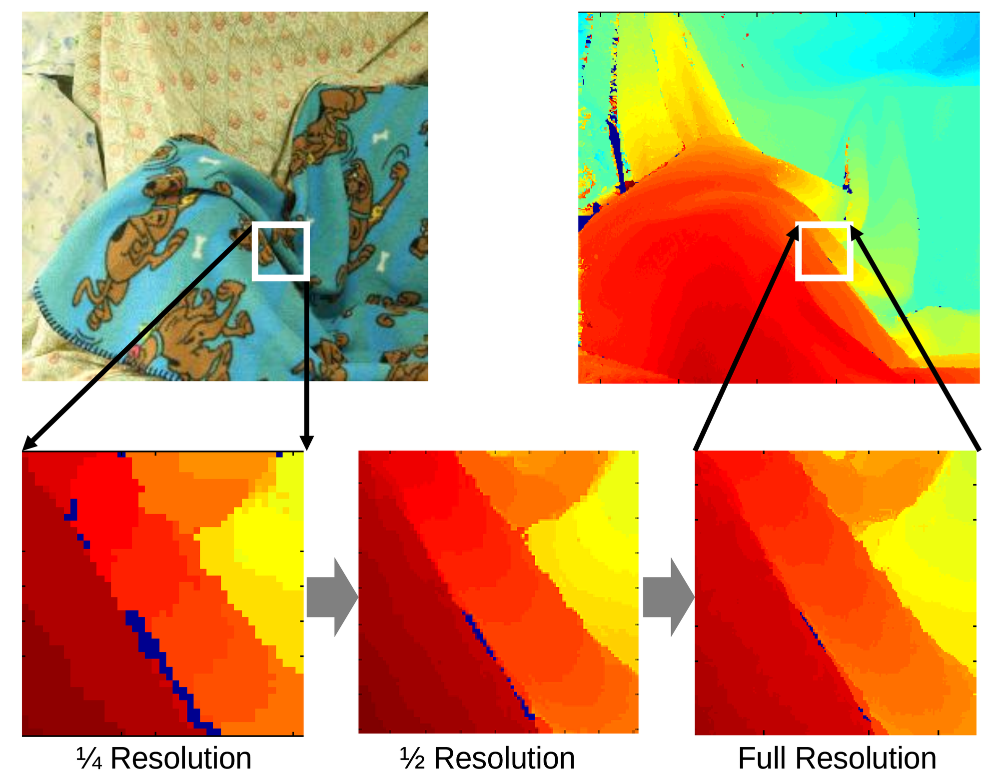
pipeline of stereo algorithm

Multi-Resolution Background Modeling



Foreground detection is performed on multiple resolutions, from low to high. On higher resolutions, the actual foreground detections only take place at the boundary blocks of foreground blobs. Only the green pixels on the above foreground masks represent the foreground pixels detected by comparison to the background model

Multi-Resolution Stereo Matching



Coarse-to-fine matching on multiple resolutions

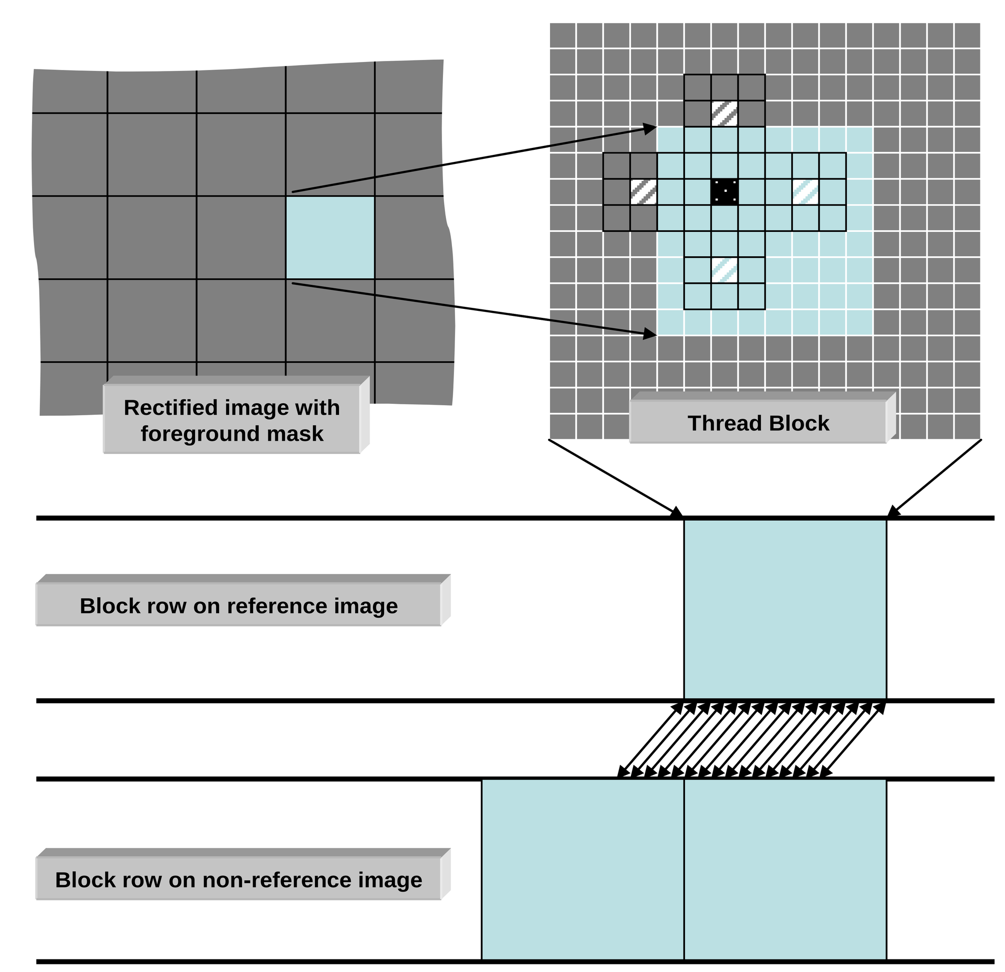
Our stereo matching is performed in a coarse-to-fine framework: after the lowest resolution pair has been processed, the resulting low resolution disparity map is used in the next pass of processing. In higher resolutions, the matching is performed only on the searching range suggested by lower resolution disparity maps, as well as on the foreground pixels detected at higher resolution. This iteration continues until the full resolution disparity map is obtained.

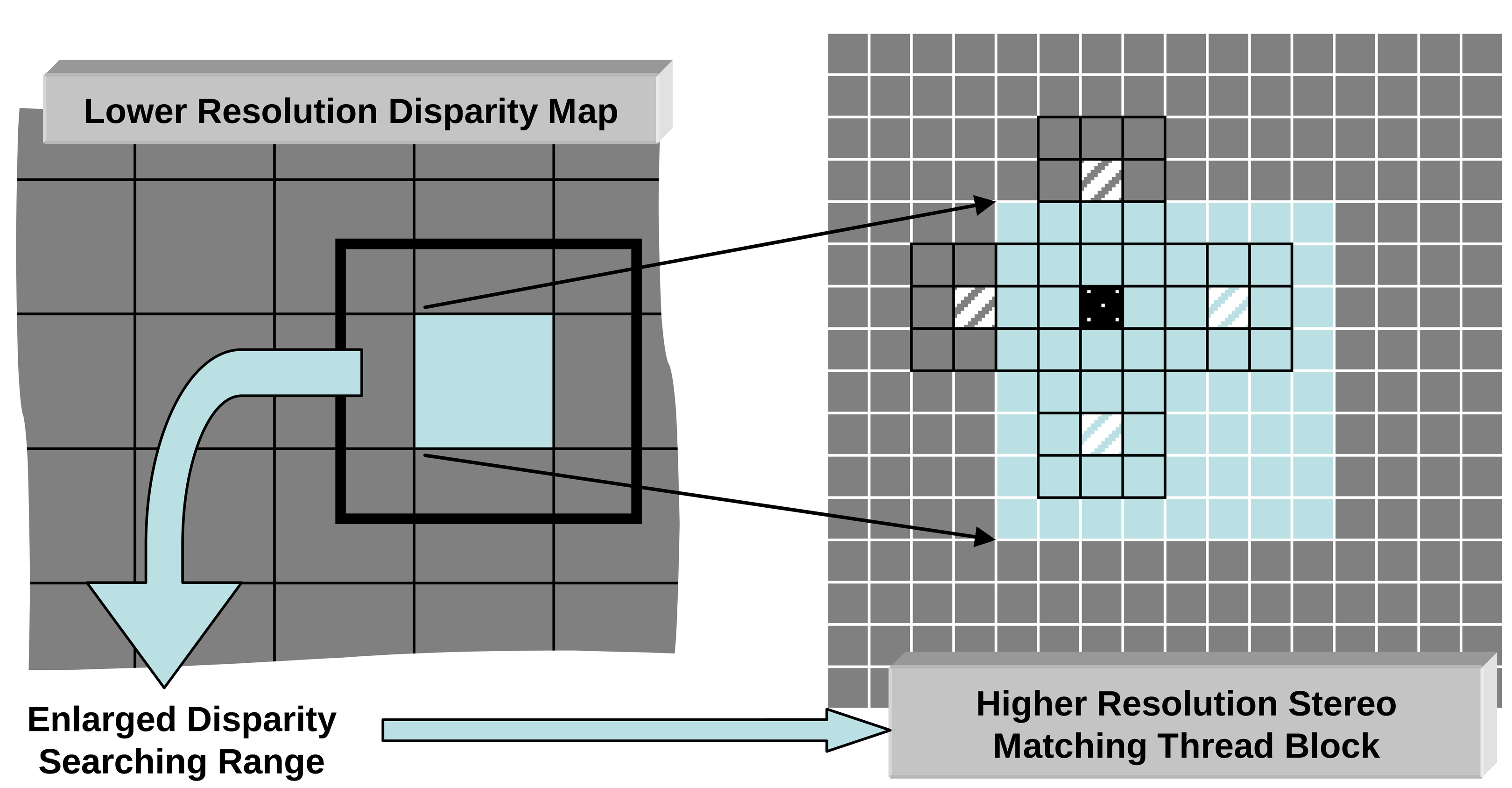
Once the original resolution disparity map is computed, sub-pixel disparity refinement is optionally applied in applications that need better precision.

The CUDA implementation of our stereo matching is illustrated in the diagram on the right.

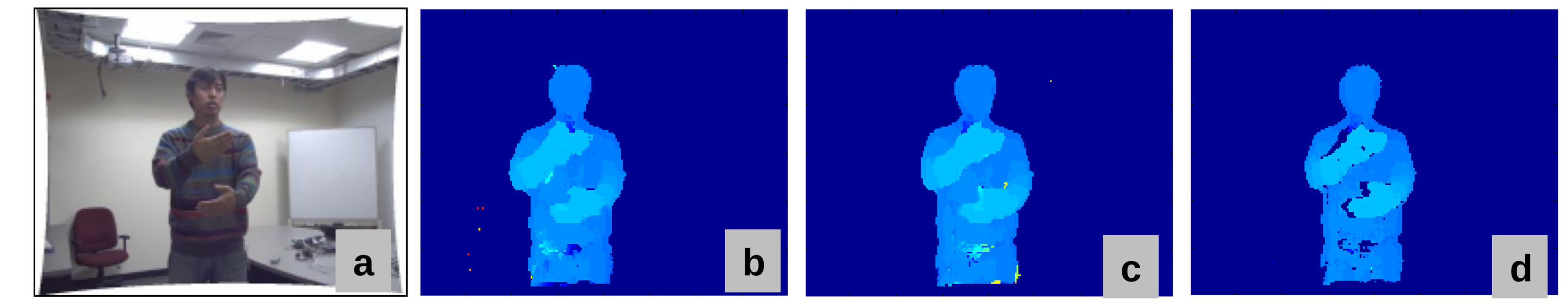
Each CTA of GPU takes care of the matching for a small block of rectified image. For each pixel, Yang's adaptive window algorithm [1] is used to search for the best matching along the scan line. A block of reference image is loaded into CTA before matching starts, and all the blocks on the non-reference image are loaded into CTA one by one as matching proceeds.

In order to accelerate the computation, A 2D scan algorithm within CTA is used to integrate image blocks so that the error aggregation can be performed faster.



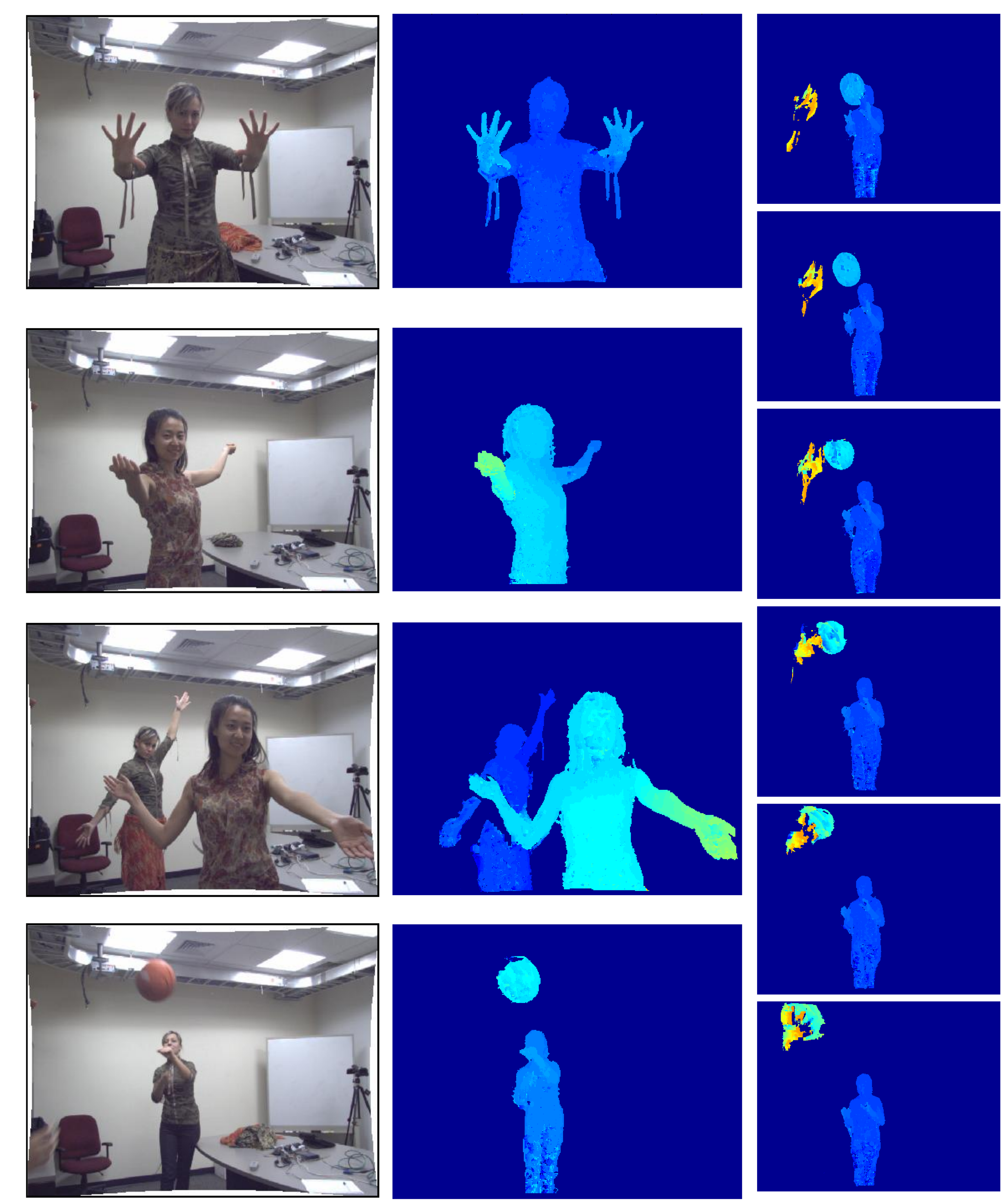


Meanwhile, the CTA thread for higher resolution stereo matching uses the disparity searching range suggested by an area on the low resolution disparity map, which is twice the size of corresponding area.



On each resolution, we run dense matching on two opposite directions by switching the reference image and non-reference image. The results are displayed in (b) and (c). We then do a Cross-Checking [2] by comparing these two disparity map. This is an effective way to detect most matching errors and improve the matching accuracy, as indicated in (d).

Experimental Results



Reference:

[1] R. Yang and M. Pollefeys. Improved Real-Time Stereo on Commodity Graphics Hardware. In Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR) workshop on Real-time 3D Sensors and Their Use, 2004

[2] S. D. Cochran and G. Medioni. 3D Surface Description from Binocular Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 14(10), pages 981-994, 1992.