“REAL TIME DISTANCE ESTIMATION USING GPU”

PROJECT REFERENCE NO. : 37S0754

COLLEGE : HKBK COLLEGE OF ENGINEERING, BANGALORE
BRANCH : COMPUTER SCIENCE AND ENGINEERING
GUIDE : DR. WASEEM AHMED
STUDENTS : VINAY KUMAR BANSAL
AZHAR LIAQUAT
SUHAID JAVIAD
WAQAR AHMED

Keywords: CUDA programming, stereo vision, distance estimation, real time, parallelization, etc.

Introduction:

A Stereo Vision is a device that is designed to extract depth information, two images of the same scene on a card differing only in horizontal placement of the camera. When the stereo vision is used, the left and right eyes are presented with the two different views of the scene. The two images are blended together as if the user was actually viewing the scene, with all the lifelike depth information present. From this, modern stereo vision, as performed by a computer system utilizing differing views of a scene taken from digital cameras, was developed.

Real Time Distance Estimation using GPU is an application which uses the Graphics processing unit to generate the pixels in real time and detect the object and estimate the distance between the object and the observer. This application generates the disparity map and, if the difference between the calculated average disparity of infinity points and the expected disparity is below a certain threshold, than the system is considered to have been unchanged.

There has been a lot of work related to stereo vision. The Tyzx DeepSea G2 Vision System [1], describes a taskable, Embedded Stereo Camera build vision systems suitable for deployment in devices that operate in demanding dynamic, variably lit, real-world environments. In [2], a real-time stereo vision applications have been described which
include SAD-based Implementations, phase-based Implementations and disparity Map Refinement. In [3], work on Stereo Vision like epipolar constraints, disparity and depth, correspondence problems and various phases of Stereo Vision System have been implemented. High speed stereo vision sensors [4] are also available for High-speed video, 3D dynamic shape, stereo vision, motion, and tracking. These sensors use very high speed calibrated cameras (500fps) to capture fast moving high intensity images. And it needs very high computation speed. 3D Acquisition is very difficult task as very high resolution images are being captured and processed. In [5] distance estimation is done using stereo vision where two Philips SPC 900NC webcams are used to form stereo vision. However, all the systems described above calculate distance for static objects from stored images. This work attempts to estimate the distance on streaming images (or video) dynamically in real time.

**Objectives:**

For automobiles operated by physically challenged people, a system that assists them in collision detection and avoidance is essential. One of the components of such a system is distance estimation of objects in the front of the automobile. Although distance sensors can be utilized for accurately finding the distance of objects, many such sensors will be needed for a real world operation considering the height and width of the automobile. Another more effective way to achieve this would be to use computer vision to estimate the distance of objects in the scene. However, the computations involved in achieving the result consume a lot of computation time. The results from this approach, in most cases, cannot be obtained in real time. As the number of frames captured by even a low-end camera exceeds 30 per second, the processing needs to be extremely fast to synchronize with the incoming streams of pixels. The objective of the project is to achieve in real-time the estimation of the distance of moving objects in a scene. The real-time aspect is achieved by accelerating the stereo vision algorithms used for distance estimation using a multi-core multithreaded Graphics Processing Unit (GPU) using parallelization.
1. Evaluating existing stereo vision algorithms based on complexity, performance availability and understandability.
2. Calculating the position and baseline distance between the two cameras based on the selected algorithm and the scene.
3. Implementing the algorithm to determine computation time. Tabulate the results.
4. Identifying the bottlenecks in the code by static and dynamic profiling.
5. Parallelizing the bottlenecks for faster execution on the GPU.
6. Repeating step 5 with different combinations of code transformations
7. Evaluate the alternatives to find the best (fastest) transformation in the set from step 6.
8. Test the system on real scenes and document the results.
9. Compare results from 3 and 8.
Results and conclusions:

Though the existing program consumed 0.47 seconds, after writing CUDA kernel Our CUDA kernel has improved the Performance by 400%

We succeeded in achieving our objective as the object detection is done in real time. It is proved in our project that the time complexity shows the object is detected in 0.16 second, almost 4 time the existing system. The graphics processing unit make use of its cores and generate pixel in real time.

We also figured the main bottle neck in the program stereoSavePointCloud(). Here each pixel of the picture was being processed, and we actually wanted this. And this was the most time consuming part of the program and hence we improved the time complexity.

Future Enhancement:

As we concentrated in reducing the speed of detection of object, the algorithm speed is improved, but distance estimation part can be improved.

Scope for future work:

The real time distance estimation system to be developed as part of this project has various uses

- Can be installed as a subsystem in automobiles to aid physically challenged people in detecting and avoiding collision. Based on this subsystem speed and distance of moving objects can be estimated and based on this necessary action like applying the brakes, popping up the air bag, etc. can be performed by the other sub systems.
- This enables automobiles in heavy traffic to move closely as this system helps in avoiding collision by indicating or applying brakes whenever the distance between automobiles becomes lesser than a set threshold.

In robotics and automation, proper distance estimation is important in a variety of applications for example obstacle avoidance for mobile robots, object grasping, etc.