Design and Development of an OpenCV Based Parallelised Optic Flow Algorithm

Student's Name	**Punit B. Yadravi**	**RTES (FT-2012)**
Academic Supervisor(s)	N. D. Gangadhar	
Industrial Supervisor(s)		

Punit B. Yadravi

punit.b.yadravi@gmail.com
Ph. No: 0 96634 47876

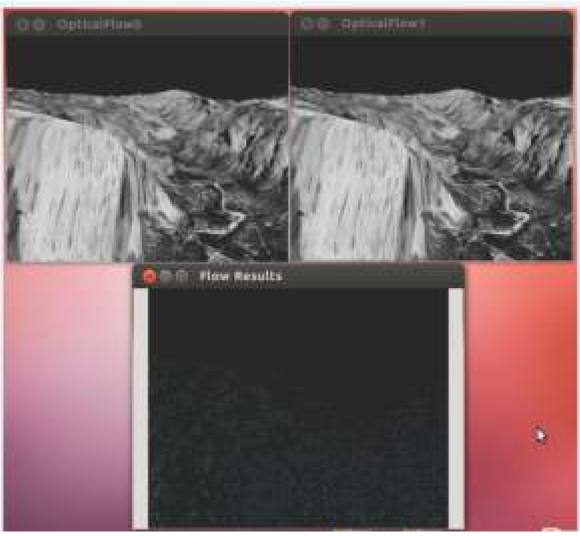
Keywords: Optic Flow, Horn and Schunk, Sobel Operation, Veclocity Flow Vectors, Object Detection

Abstract:

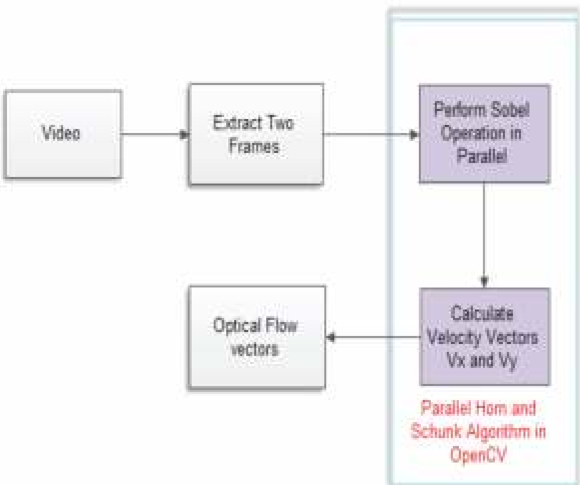
Optical Flow is a popular technique for motion estimation, obstacle detection and avoidance essential for camera based navigation of autonomous vehicles and robots. Optical Flow vectors represent the movement of a pixel from one frame to the next which involves the identification of the presence of an object in successive frames. Embedded applications, such as micro air vehicles with limited computational power and battery power requirements, demand efficient real time estimation of the Optical Flow. Horn and Schunk differential method for Optic Flow has remained one of the most used algorithms for Optical Flow computation. The OpenCV computer vision library comes with an implementation of Horn and Schunk Optical Flow implementation. All Optical Flow based applications can benefit from an improved efficiency of the OpenCV implementation via parallelisation.

This project involves the design and development of an OpenCV based parallel Horn and Schunk algorithm for Optical Flow computation in order to achieve better computational performance. The program flow of the existing Horn and Schunk implementation is analysed and regions of parallelisation identified. At region of computation, Data Parallel approach is employed by dividing the image frames into multiple units of computation to be handled by independent threads. Multithreaded computation is programmed using the OpenMP library after integrating it with OpenCV. The shared common state variables are converted to thread specific local variables to achieve complete thread independence. The number of threads is a parameter of the parallelised implementation which the application can be optimally set for maximum efficiency based on the number of hardware threads.

The efficiency of the parallelised Horn and Schunk algorithm implementation is validated against the original non-parallel implementation for its computation performance using standard test video sequences. The average speedup is measured for multiple frames of the test sequences for different number of threads. While the developed algorithm provides speedup in the execution time when compared to the non-Parallel code for all choices of the number of threads, the optimum performance is achieved when the later matches the number of hardware threads of the CPU. Overall an average speedup of 1.18 is measured to have been achieved. The developed Parallel Horn and Schunk algorithm can be further optimised based on the target processor for improved real time application performance.



Optic flow vectors for yosemite image



Block diagram of motion estimation algorithm