## Project Motion Tracking

The scope of this project is to blend the available technology, Cuda, into tracking the motion from a set of pictures which together form a video of an object in motion. We will be using OpenCV and several edge detection libraries in order to fully differentiate the object from the surrounding.

Therefore in the multiple available sources focused on extracting the essence of GPU-based parallel programming, we notice that the Vector Coherence Mapping, we address the viability of GPUs for general purpose vision processing by showing that emerging programming frameworks support efficient mapping of visionalgorithms onto graphics hardware. To this end, we implemented a computation and data-intensive algorithm for motion vector extraction on various GPUs and compare the performance against a state-of-the art CPU.

In the other files except motion\_tracking.cu, we deal with the convolution of the image, image storing in the GPU processor, applying the **Sobel filter** for emphasizing the edges of the object (edge\_detect.cu), detecting mismatching pixels (edge\_detect.cu and motion\_tracking.cu), determining the density map of a specific frame (motion\_tracking.cu).

By maintaining a history of vectors extracted in previous frames, VCM is able to estimate the termination point of the next vector by applying a constant acceleration assumption. This ‘temporal estimate’ is used to bias the vote to pick a vector that maximizes both spatial and temporal coherence of the resulting vector field. VCM is composed of three main phases: Interest Point (IP) extraction, NCM computation and VCM computation.

We segment the video image into 16×16 sub-windows for IP extraction. A set of CUDA blocks process these sub-windows until all the sub-windows are completed. Within the block, a 16×16 ‘result array’ each processing thread is responsible for a pixel, computing the Sobel gradients, image difference, and fuzzy-And operation. The resulting spatio-temporal (s-t) gradient is entered into a 16×16 array in shared memory. Since we keep the source video images in texture memory, most of the memory access to the images are cache hits.