**Parallelization and Optimization of SIFT on GPU**

**Using CUDA**

El artículo se enfoca a paralelizar SIFT (Scale-invariant feature transform) en un GPU. SIFT es un algoritmo de reconocimiento de patrones, con el cual se extraen característicos de imágenes. Los autores presentan estrategias de cómputo parlelo, para implementar y optimizar el algoritmo SIFT con el modelo de programación CUDA. Los resultados experimentales demuestan que la implementación del artículo presente es 2.5 veces más rápida que implementaciones anteriores. Con esta implementación se demuestra una velocidad de 20 frames por segundo en imágenes de 1280x960 pixeles; con imágenes de 1920 x 1440 la velocidad fue 11 frames por segundo.

Los autores muestran como la el la arquitectura no afecta en la cantidad de clasifcaciones correctas. Tanto la implementación en CPU, y la implementación en GPU obtienen el mismo porcentaje de casos correctamente clasificados. Además, muestran como con forme aumentan la resolución de la imagen, el “speedup” aumenta también. De tal manera que con imágenes de 320x240 el speed up es de 10x, pero para imágenes de 1920x1440 el speed up es de 50x.

El experimento lo corrieron con un Intel i7 2600K CPU corriendo a 3.4GHz y con 16GB de memoria. Por otro lado el GPU era una NVIDIA Tesla C2050 la cual tiene 448 shadders corriendo a 1.15GHz con 3GB de memoria.

Los autores comentan que la forma en la que alcanzaron los speedups motrados son con el uso efectivo de los bloques e hilos en el GPU. Además mencionan que uno de los mejores resultados fue al mejorar la ocupación de los SM (Streaming multiprocessors). De tal manera que optimizaron la carga de imbalanseada y usaron paralelización de corrientes que son significativas para mejorar el rendimiento del SIFT paralelizado en el GPU.

Yonglong, Z., Kuizhi, M., Xiang, J., & Peixiang, D. (2013, November). Parallelization and Optimization of SIFT on GPU Using CUDA. In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC\_EUC), 2013 IEEE 10th International Conference on* (pp. 1351-1358). IEEE.