Accelerated SIFT

Author: Thomas Twanlee Tan
Supervisor: Dr. Wai Ho Li
ECSE Final Year Project

What is SIFT?

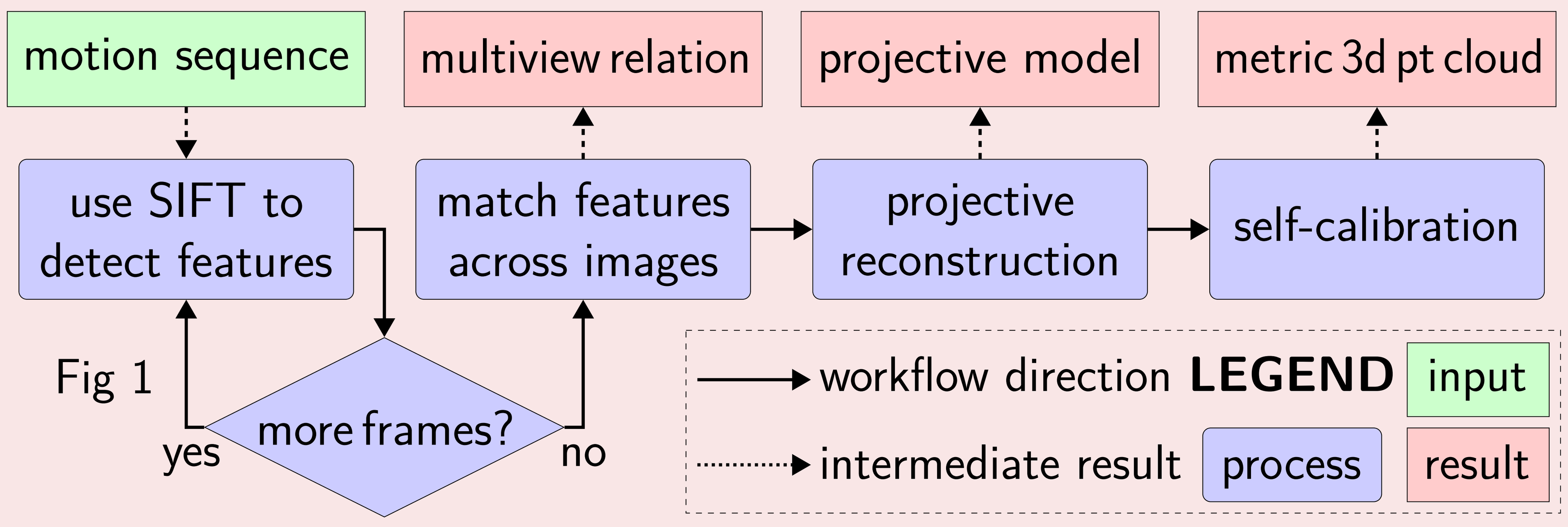
The Scale-Invariant Feature Transform, commonly known as just SIFT, is a technique in computer vision used to detect and describe local features in an image for applications in object recognition, robot navigation, image stitching, 3d reconstruction and augmented reality.

Uses of SIFT

SIFT offers application benefits over traditional descriptors such as Harris' corners or the Canny edge detector by providing robustness to scale, illumination, rotation and other affine distortions [1]. Despite many competitors, GLOH, Speeded-Up Robust Features and the Level-Line-Descriptor to name a few, studies [3] show the outperformance of SIFT provided its computational costs are acceptable [2].

Application – Structure from Motion (SfM)

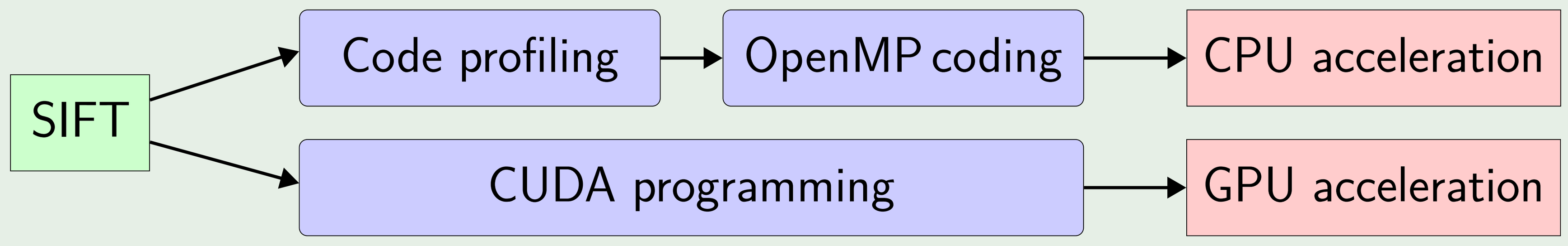
A model's 3d structure is a useful starting point for many applications including navigation in robotics or mesh modeling in computer graphics. Figure 1 shows the application of SIFT to Structure from Motion (SfM). The structure obtained is in the form of points in 3d space. Of significance in this result is the structure being obtained from a public image sequence with no physical possession of the original dinosaur object, highlighting the ability to achieve a 3d model even when physical possession may not be possible. A 3d point cloud can be upgraded to a water-tight surface model with software such as Cocone. See Figure 2 for results.



How can SIFT be accelerated?

Our work focused on an accelerated version of SIFT and its applications. From an existing code base, two separate versions were created using different techniques:

- Traditional parallelization with OpenMP,
- Code execution on an Nvidia Graphics Processing Unit (GPU) running CUDA.



Code Acceleration (Traditional Methods)

The traditional method of acceleration takes advantage of all CPUs in the architecture. Code profiling was enabled and GNU's gprof was used on the original build to reveal in which functions of code most execution time was spent. It is presumed that accelerations are most beneficial in these portions of code.

Table 2 lists functions profiled and OpenMP accelerated with their speedups.

GPU Acceleration

Acceleration using the Nvidia GPU architecture is a much more involved process. Code must be re-written to account for memory hierarchy, resource occupancy, data movement and the division of code to separately target a "host" and "device".

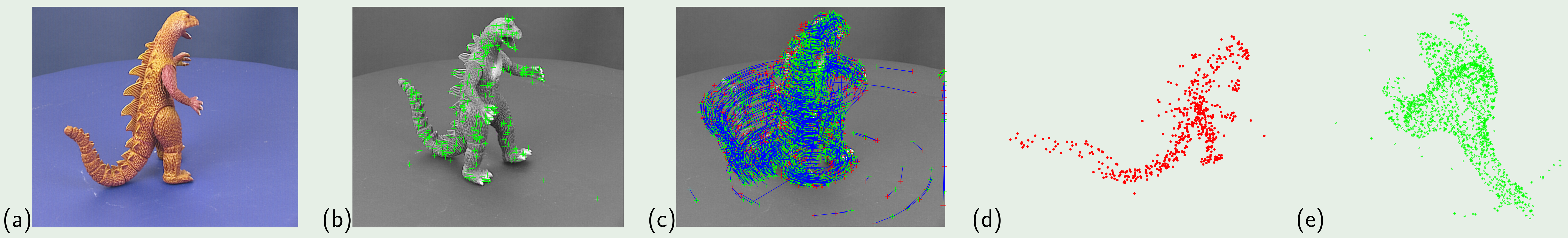
See Table 1 for results, including the impact of data movements.

Results – GPU Acceleration

	Execution time	Data transfer overhead	Total time
CPU	4381 ms	N/A	4381 ms
GPU	13.2 ms	178 ms	191 ms

Table 1: comparison of execution time required to perform convolution.

Results – Structure from Motion



(a) (b) (c) (d) (e)

Figure 2: (a) The first of 37 frames of motion used as the input sequence to compute the dinosaur's structure from motion. (b) The first step is to detect interesting features in the image with SIFT, which requires the image to be in greyscale. The location of detected features have been marked with green crosses. (c) After tracking features through several frames, the camera and object's relative location becomes apparent by applying mathematical constraints. (d) The constraints provided by the Fundamental matrix (point-to-line correspondences), homographies (point-to-point correspondences) and metric tensors allow a preliminary reconstruction up to projective ambiguity to be formed. (e) By applying auto-calibration and a combination of subsequence merging and bundle adjustment, the projective ambiguity can be removed. The refined 3d model allows views from angles not present in the original sequence, all without the physical possession of the dinosaur object. Further work to upgrade this model to remove phantom points and create a surfaced water-tight mesh are deferred to the field of computer graphics.

References

[1] N. Govender. Evaluation of Feature Detection Algorithms for Structure from Motion. In *3rd Robotics and Mechatronics Symposium (ROBMECH 2009)*, Pretoria, South Africa, November 2009.

[2] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[3] J. M. Morel and G. Yu. On the Consistency of the SIFT Method. *American Institute of Mathematical Sciences, Inverse Problems and Imaging*, 26, 2008.

Results – Code Acceleration

Time %	Time (sec)	# calls	Function name	Speedup
67.73	0.49	1266042	GenerateHistogram	
12.90	0.25		GaussianBlur	2.36
9.68	0.28	31	ConvolutionVertical	2.36
3.23	0.29	31	ConvolutionHorizontal	2.36
3.23	0.30	7	FindExtrema	2.16

Table 2: profiling data obtained of the most time consuming code, and the speed up obtained from using OpenMP.

MONASH University Engineering

Department of Electrical and Computer Systems Engineering

GO Boldly.