Geekbench 5
Compute Workloads

Introduction

This document outlines the workloads included in the Geekbench 5 Compute Benchmark suite.

Compute Benchmark scores are used to evaluate and optimize GPU Compute performance using workloads that include image processing, computational photography, computer vision, and machine learning. Performance in these workloads is important for a wide variety of applications including cameras, image editors, and real-time renderers.

Platform Support

Platform	Minimum Version	Comment
Android	Android 7 “Nougat”	
iOS	iOS 12	
Linux	Ubuntu 16.04 LTS	
macOS	macOS 10.13	
Windows	Windows 10	

API Support

Geekbench 5 supports the following GPU Compute APIs:

API	Version	Comment
CUDA	CUDA 9.0	Compute Capability 3.0 or later.
Metal	Metal 2.0	Metal 2.1 if available.
OpenCL	OpenCL 1.1	
Vulkan	Vulkan 1.0	

Runtime

Geekbench 5 runs Compute workloads in the order listed here as the Compute Benchmark. Each workload is run for 20 iterations by default.

Scores

Geekbench 5 provides one overall score for the Compute Benchmark. The overall score is the geometric mean of the scores of the individual Compute workloads.

Comparing Scores

Each Compute workload has an implementation for each supported Compute API. While it is possible to compare scores across APIs (e.g., a OpenCL score with a Metal score) it is important to keep in mind that due to the nature of Compute APIs, the performance difference can be due to more than differences in the underlying hardware (e.g., the GPU driver can have a huge impact on performance).

Compute Workloads

Sobel

The Sobel operator is used in image processing and computer vision for finding edges in images.

The Sobel workload converts an RGB image to greyscale and computes the Sobel operator for the greyscale image. The operator uses two integer convolutions (one for horizontal edges, and one for vertical edges) to compute the final image.

Canny

Like the Sobel operator, the Canny algorithm is an edge detector. However, Canny is significantly more complex as it uses a multi-stage algorithm to find edges in images:

1. Apply a Gaussian Blur to the image to remove noise.
2. Calculate the gradients of the image.
3. Apply non-maximum suppression to remove spurious edges
4. Finalize edges by removing uncertain edges that are not connected to certain edges.

Stereo Matching

The Stereo Matching workload constructs a depth map from a pair of images taken from the same scene.

The Stereo Matching workload uses a block matching algorithm to find each pixel's disparity and generate the map. This algorithm matches each block of pixels in one image, to the closest block in the second image, where a Sum of Absolute Differences (SAD) is used as a metric for how close two blocks of pixels are from one another.

Histogram Equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. The Histogram Equalization workload performs this adjustment on a 2576×3872 image.

Gaussian Blur

The Gaussian Blur workload blurs an image using a Gaussian spatial filter. Gaussian blurs are widely used in software—both in operating systems to provide interface effects, and in image editing software to reduce detail and noise in an image. Gaussian blurs are also used in computer vision applications to enhance image structures at different scales.

The Gaussian Blur workload blurs an image using a Gaussian spatial filter. While the workload's implementation supports an arbitrary sigma, the workload uses a fixed sigma of 3.0f. This sigma translates into a filter diameter of 25 pixels by 25 pixels.

Depth of Field

The Depth of Field workload computes a lens blur "depth of field" image given two images — a depth image and a colour image. It accomplishes this by applying a blur to each pixel of the colour image that is proportional to the distance of the pixel from the focal point (as determined by the depth image).

Face Detection

Face detection is a computer vision technique that identifies human faces in digital images. One application of face detection is in photography, where camera applications use face detection for autofocus.

The Face Detection workload uses the algorithm presented in "Rapid Object Detection using a Boosted Cascade of Simple Features" (2001) by Viola and Jones. The algorithm can produce multiple boxes for each face. These boxes are reduced to a single box using non-maximum suppression.

Horizon Detection

The Horizon Detection workload searches for the horizon line in an image. If the horizon line is found, the workload rotates the image to make the horizon line level.

The workload first applies a Canny edge detector to the image to reduce details, then detects lines in the image using the Hough transform, and finally picks the line with the maximum score as the horizon.

Feature Matching

The Feature Matching workload finds a match of keypoints (or features) between two images using the ORB algorithm. ORB is used by other algorithms (such as Structure from Motion) to find the matching keypoints across the two images that are then used to generate the 3D map of those points.

Particle Physics

Particle physics simulations are used for many applications including the simulation of fluids and smoke for games.

The Particle Physics workload implements a simulation where particles interact with one another and their environment via elastic collisions. Other particle-particle forces are ignored. The Particle Physics workload uses 4,096 particles in its simulation.

SFFT

FFT (Fast Fourier Transform) decomposes an input signal into a linear combination of a basis of trigonometric polynomials. FFT is a core algorithm in many signal-processing applications.

The FFT workload executes an FFT on a 32MB input buffer operating in 1KB chunks. This is similar to how FFT is used to perform frequency analysis in an audio processing application.