NVIDIA OPTICAL FLOW SDK

Application Note

vNVOFA_DA-09418-001_v03 | June 2021

Table of Contents

Chapter 1. NVIDIA Optical Flow Accelerator

1.1. Introduction

NVIDIA® GPUs, starting with the NVIDIA Turing™ generation, contain a hardware accelerator for computing optical flow and stereo disparity between frames (referred to as NVOFA in this document), which works independently of graphics/NVIDIA CUDA® cores. With end-to-end optical flow calculation offloaded to NVOFA, the graphics/CUDA cores and the CPU are free for other operations.

Optical flow vectors are useful in various use-cases such as object detection and tracking, video frame rate up-conversion, depth estimation, stitching etc. It is also observed that using flow vectors for object detection also increases inference accuracy[1].

The hardware capabilities of NVOFA are exposed through APIs referred to as NVOF APIs.

1.2. NVOFA Capabilities

NVOFA engine can operate in two modes:

- **Optical Flow Mode**: In this mode, the engine generates flow vectors between two given frames, returning both X and Y components of the flow vectors.
- **Stereo Disparity Mode**: In this mode, the engine generates flow vectors in X direction only. This mode is useful in use-cases in which the Y-component of the vectors is not required, or it is known a priori that it is zero (e.g. finding disparity between the left and right images of a stereo capture). Stereo Disparity mode will be deprecated in future SDK releases. Client applications can alternatively use the X component of the flow generated in Optical Flow mode.

The hardware generates flow vectors block-wise, one vector for each block of 4 × 4, 2 x 2 and 1 x 1 pixels (referred to as grid). The generated vectors can be further post-processed in software to improve accuracy; up sampled to produce dense flow map.

[1] Refer http://openaccess.thecvf.com/content_cvpr_2018/papers/Yang_Making_Convolutional_Networks_CVPR_2018_paper.pdf and http://openaccess.thecvf.com/content_cvpr_2017/papers/Carreira_Quo_Vadis_Action_CVPR_2017_paper.pdf.

NVOFA hardware natively supports multiple hardware contexts with negligible context-switching penalty. As a result, subject to the hardware performance limit and available memory, an application can generate motion vectors for multiple contexts simultaneously.

The NVOFA hardware is supported for all Turing GPUs (except TU117) and above.

1.3. NVOF API

Capabilities of NVOFA are exposed via NVOF APIs. NVOF API includes three types of software interfaces:

- CUDA: Cross-platform API, works on Linux and Windows.
- DirectX 11: Works on Windows 8 and above.
- DirectX 12: Works on Windows 10 20H1 and above.

Refer to the sample applications included in the Optical Flow SDK for more details.

Table 1 and Table 2 summarize the capabilities of the NVOFA hardware and the new features exposed through NVOF APIs in Optical Flow SDK 3.0 respectively.

Table 1. NVOFA Hardware Capabilities

Hardware Features	Turing	Ampere
Optical flow and stereo mode	**Y**	**Y**
Support for external hints	**Y**	**Y**
4x4 grid size	**Y**	**Y**
2x2 and 1x1 grid size	**N**	**Y**
Hardware cost	**N**	**Y**
Region of interest (ROI) optical flow calculation	**N**	**Y**
Maximum supported resolution	4096x4096	8192x8192

- **Y**: Supported, **N**: Unsupported

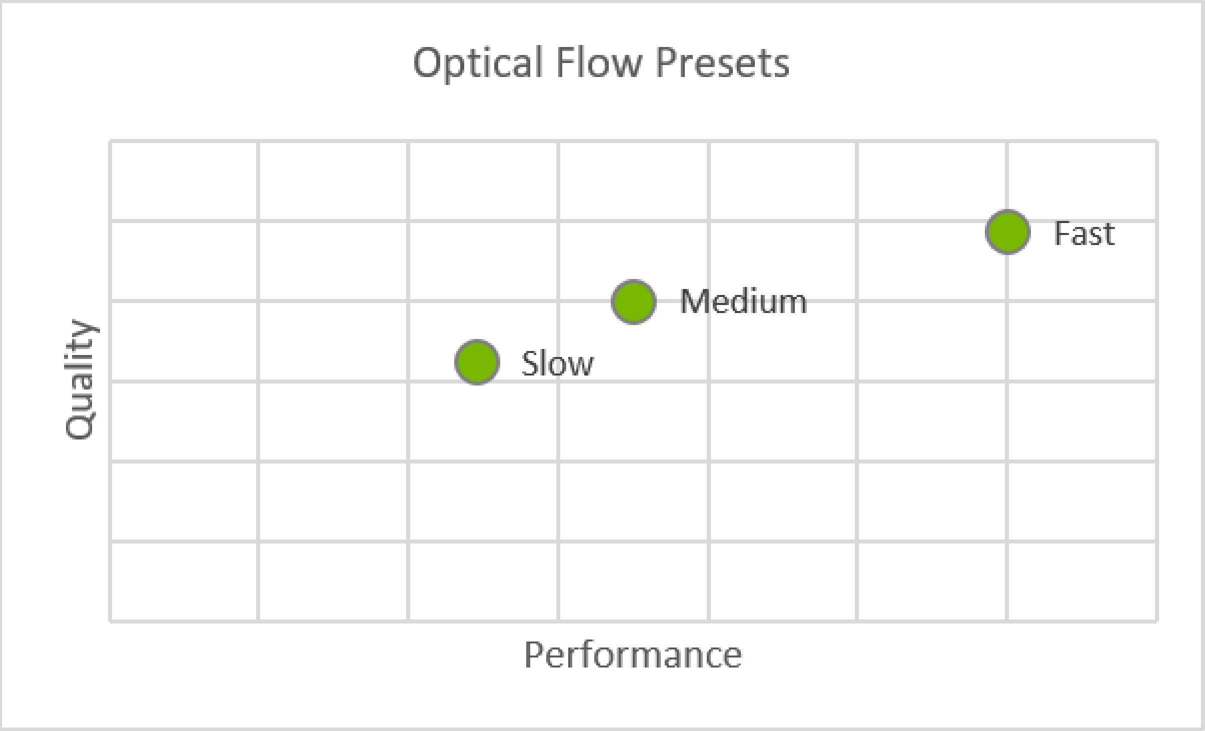
Table 2. New features exposed through NVOF SDK 3.0

Sr. No.	Feature	Description
1	DirectX 12 interface	This SDK adds support for DX12 interface, where the client application can generate the flow for a pair of DX12 resources.
2	Forward-Backward flow	With single execute call, client application can generate forward and backward flow which can be further used in validating the correctness of the flow vectors.
3	Global flow	The Optical Flow API can also output the global flow vector (for the entire frame). The global flow vector is calculated from the forward flow vectors only.

1.4. NVOFA Quality and Performance

The NVOF API exposes multiple quality and performance levels (which are referred to as *presets*) which the user can choose based on the desired quality and performance requirement. Figure 1 shows the performance/quality trade-off to be expected with the presets.

Figure 1. Performance/Quality Presets exposed in NVOF API



The NVOFA provides real time performance with small CUDA core utilization. Table 3 shows *indicative*[2] performance and quality of NVOF API on KITTI 2105 which is publicly available data set. Users can trade quality vs. performance by choosing the right preset. Note that performance numbers in Table 3 are measured with assumptions listed under the table. The performance varies across GPU classes (e.g. Quadro, Tesla), and scales (almost) linearly with the clock speeds for each hardware.

[2] NVOFA performance depends on many factors, including but not limited to: OFAPI settings, GPU clocks, GPU type, video content type, instantaneous available memory bandwidth etc.

Table 3. Indicative quality and performance

Grid size	Preset		Fl-bg		Fl-fg		Fl-all		Performance	
			Turing	Ampere	Turing	Ampere	Turing	Ampere	Turing	Ampere
4x4	SLOW	NOC	21.75	16.57	24.89	26.77	22.29	18.32	216	227
		OCC	33.58	28.92	27.62	29.32	32.67	28.98		
	MEDIUM	NOC	21.68	19.71	36.59	33.30	24.23	22.04	463	571
		OCC	33.45	31.54	38.89	35.66	34.28	32.17		
	FAST	NOC	22.81	22.43	47.39	39.02	27.01	25.26	900	804
		OCC	34.34	33.86	49.32	41.25	36.64	35.00		
2x2	SLOW	NOC	N/A	16.24	N/A	26.95	N/A	18.07	N/A	101
		OCC		28.67		29.52		28.80		
	MEDIUM	NOC		17.48		33.32		20.18		172
		OCC		29.70		35.67		30.61		
	FAST	NOC		20.17		35.69		22.82		319
		OCC		32.02		38.02		32.94		
1x1	SLOW	NOC		16.57		27.31		18.41		31
		OCC		28.98		29.82		29.10		
	MEDIUM	NOC		17.73		30.22		19.86		48
		OCC		29.83		32.65		30.26		
	FAST	NOC		19.69		34.51		22.22		92
		OCC		31.64		36.82		32.43		

- The above data is generated for Optical Flow mode in NVOF API using .\Samples \AppOFCuda on RTX2070 and GA100, respectively on Windows.
- The performance on Windows (on CUDA interface) is typically lesser than that of Linux due to a known bug inside NVIDIA display driver.
- All measurements are done by setting the video clocks as reported by nvidia-smi at 1755 MHz. The performance should scale according to the actual video clocks for other GPUs. Information on nvidia-smi can be found at https://developer.nvidia.com/nvidia-system-management-interface.
- Resolution/Input format: 1920x1080/YUV 4:2:0
- Software: Optical flow SDK 3.0, NVIDIA display driver: 466.27
- Fl-all = Percentage of vectors having an average EPE > 3 on KITTI 2015.
- Fl -bg = Percentage of vectors in background pixels having an average EPE > 3 on KITTI 2015.
- Fl -fg = Percentage of vectors in foreground pixels having an average EPE > 3 on KITTI 2015.
- Details for KITTI 2015 and the test data set can be found http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow.
- NOC = Non-occluded region

- OCC = Occluded region

1.5. Programming NVOFA

Optical Flow SDK 3.0 is supported on R465 drivers and above. Refer to the SDK release notes for information regarding the required driver version.

Refer to the documents and the sample applications included in the SDK package for details on how to program NVOFA.

1.6. OpenCV Support

OpenCV is one of the most popular libraries in the field of computer vision. OpenCV library contains several CPU-based and CUDA-based algorithms for computing optical flow vectors.

NVOFA can also be used with OpenCV to significantly speed up the optical flow calculation.

Note that OpenCV is an open-source project and their usage is governed by specific licenses and terms and conditions.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgment, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, CUDA Toolkit, cuDNN, DALI, DIGITS, DGX, DGX-1, DGX-2, DGX Station, DLProf, GPU, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NVCaffe, NVIDIA Deep Learning SDK, NVIDIA Developer Program, NVIDIA GPU Cloud, NVLink, NVSHMEM, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, TensorRT Inference Server, Tesla, TF-TRT, Triton Inference Server, Turing, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2018-2021 NVIDIA Corporation. All rights reserved.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com