NVIDIA Optical Flow Engine-Assisted Object Tracker

NVOF Tracker

vNVOFA_DA-09915-001 | February 2023

Table of Contents

Chapter 1. Introduction

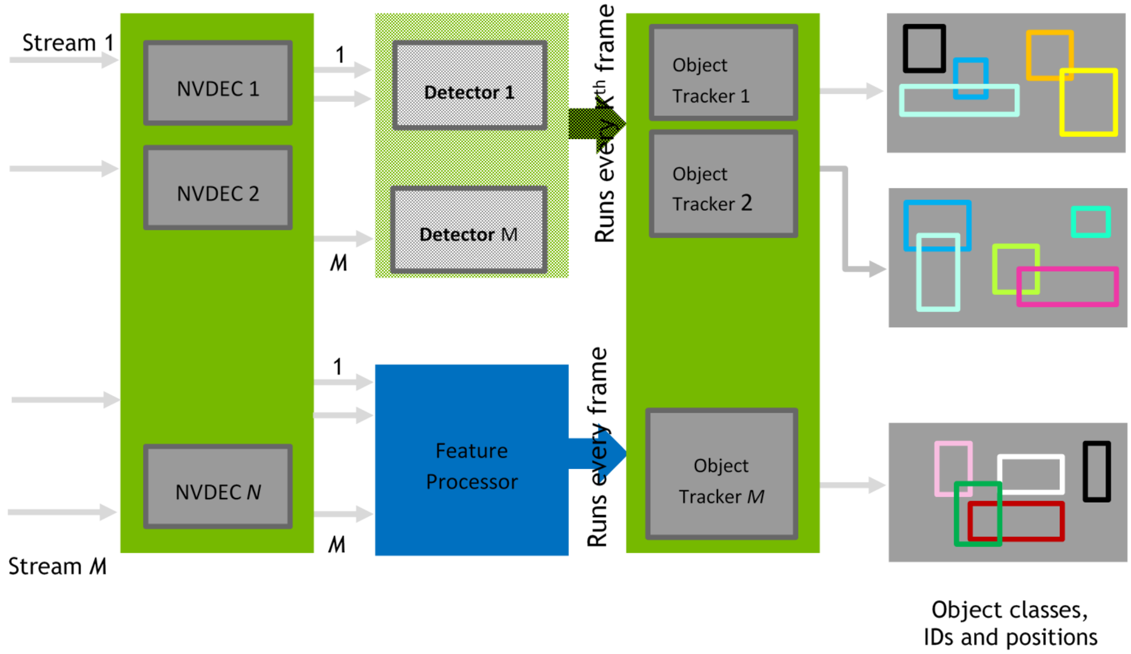
1.1. What is Object Tracking?

Given a set of objects in contiguous frames, the challenge of object tracking is to devise an algorithm that can accurately track each identified object from frame to frame. A typical object tracking algorithm achieves this tracking by assigning a unique ID to each detected object and returning the position of each object in successive frames, as long as the objects are within the frame. While there are several techniques to achieve this tracking, this document describes an efficient and highly accurate object tracking algorithm based on the NVIDIA® Optical Flow hardware accelerator.

1.2. A Classic Object Tracking Solution

Figure 1 represents a classic object tracking solution, in a typical intelligent video analytics scenario. Field cameras generate multiple streams that are fed to a video decoder (NVDEC in this case), and the decoded frames are passed through an object detector and a feature extractor/processor. The tracker, with the combined knowledge of object boundaries and object features, tracks the objects of interest across frames.

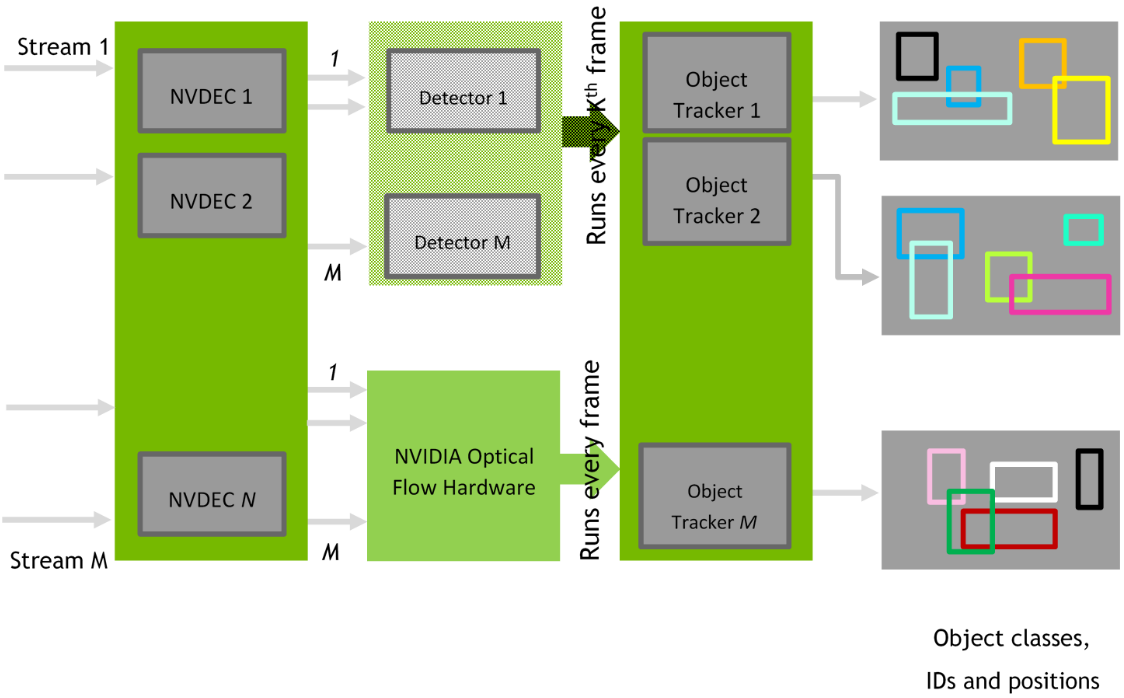
Figure 1. Classical Object Tracking Solution



1.2.1. Optical Flow based Object Tracking Solution

NVIDIA Turing™ and later GPUs have a dedicated hardware accelerator to calculate the optical flow between frames. Figure 2 represents our proposed solution where the feature tracker/extractor part is replaced with an algorithm that is based on the optical flow vectors that are returned by the GPU's optical flow engine. In classic techniques, the feature extractor/processors runs on CPU or in some cases on GPU (for example, by using CUDA). In this solution, this step is done on the Optical Flow Engine, which frees the CPU/GPU from a major computationally intensive operation and leaves the CPU/GPU free for other tasks.

Figure 2. Optical Flow-Based Object Tracking Solution



1.2.2. Object Tracker in Optical Flow SDK

The NVIDIA Optical Flow SDK contains an end-to-end object tracking application and a library that can be easily integrated into your custom application. For ease of integration, an API is provided in the `NvOFTracker.h` file. The source code for the application and the library is also available in the SDK.

The application uses the GPU's NVDEC to decode the input video bitstream, a yolov3-based object detector that runs on the GPU to detect objects of interest, an optical flow hardware accelerator to compute the optical flow between successive frames, and a combination of various algorithms to track the objects through the successive video frames.

Chapter 2. NvOFTracker: Algorithm and Internals

1. Decode the incoming bitstream into frames.
2. Feed the decoded frames to an object detector. We use a yolov3-based object detector in the application, but this object can be replaced with any object detector you choose.
    - This step detects Regions of Interest (ROI) and their respective classes.
    - The object detector runs every K^{th} frame where $0 \leq K \leq 4$.
3. Feed decoded frames to the Optical Flow Engine.
    - This step provides flow vectors between the frames, for example, frame *(P-1)* and frame *P*.
4. The tracker completes the following tasks:

    a). Maintains an ROI store. This store is constantly updated with detector ROIs.

    b). Calculates a Representative Flow for each of the ROIs.

    c). Warps the ROIs that are present in ROI store with the representative flow.

    d). Finds the best match for the warped ROI among the current frame ROIs, based on following algorithm:

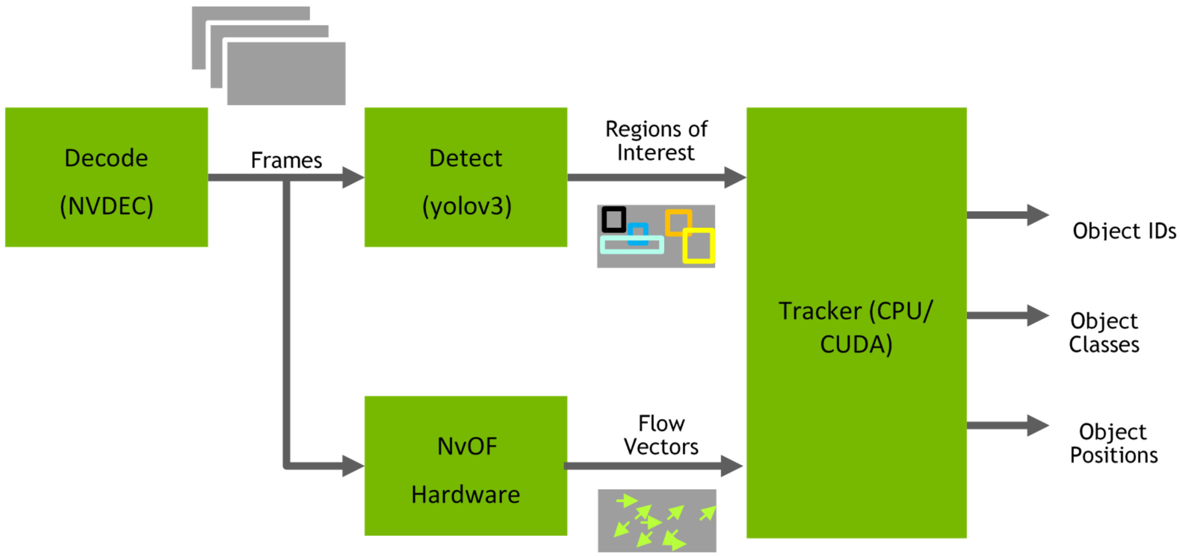
    - Calculate the centroid distance between the current frame ROI and the warped ROI.
    - Calculate the Intersection Over Union (IOU) of the current frame ROI and the warped ROI.
    - Use the weighted average of the above as the cost in the Hungarian Algorithm to find the ROI matching between frames.
    - Add the ROIs to output list.

    e). If there is no match, the tracker warps the ROI in ROI store and adds it to output list with the existing ID.

    f). ROIs in current frame that do not find a match are given new IDs and added to output list

    g). The ROI store is subsequently updated accommodating the current frame ROIs
5. Repeat Steps 1-4 for as many frames as there are in the bitstream.

Figure 3. High-Level Overview of Sample Tracking Pipeline with NVOFTRACKER



2.1. Deriving the ROI Representative Flow

As mentioned in step 4b on on page 4, NvOFTracker computes the Representative Flow for each ROI in the ROI store. The flow that is generated by NVIDIA Optical Flow engine needs to be converted to a sparse single flow vector for each of the ROIs. NvOFTracker segments the ROI in the flow vector domain into regions of similar flow based on certain thresholds and uses the median of the dominating region as the ROI representative flow. For more information, refer to the `CConnectedRegionGenerator` class in the `CConnectedRegionGenerator.h` file.

Chapter 3. 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 acknowledgement, 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.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

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

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