AGENDA

Optical Flow in Turing GPUs
NVIDIA Optical Flow SDK
Benchmarks
End-to-end applications
Roadmap
BACKGROUND
ESTIMATING PIXEL MOTION

➢ “Video” motion vectors
  ➢ Minimize encoding cost
  ➢ SAD, SATD, RDO, intra modes, partitions

➢ Optical flow vectors
  ➢ Visual motion
  ➢ Current and surrounding pixels/blocks
ESTIMATING PIXEL MOTION USING NV GPUS

- ME-only mode - Maxwell, Pascal, Volta
  - Optimized for encoding - up to 8×8 granularity motion vectors
  - Video Codec SDK 7.0+
- Optical flow (OF) - Turing & beyond
  - New hardware in NVENC
  - Optical flow and stereo disparity
  - Optical Flow SDK 1.0 (released Feb 2019)
OPTICAL FLOW ENGINE

Capabilities

• Hardware
  • Up to 150* fps at 4K
  • $4 \times 4$ pixel granularity
  • $\frac{1}{4}$ pixel resolution
  • Accuracy comparable to best DL methods
  • Advanced algorithms to find true flow vectors

• Software
  • SDK (Windows, Linux, CUDA, DirectX)

*Dependent on device clock speed
**INTENSITY DIFFERENCES**

Optical flow must be insensitive to intensity

<table>
<thead>
<tr>
<th></th>
<th>136</th>
<th>118</th>
<th>26</th>
<th>31</th>
<th>39</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>115</td>
<td>33</td>
<td>40</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>102</td>
<td>78</td>
<td>67</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>57</td>
<td>23</td>
<td>221</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>86</td>
<td>99</td>
<td>155</td>
<td>200</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>70</th>
<th>62</th>
<th>14</th>
<th>16</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>58</td>
<td>59</td>
<td>17</td>
<td>20</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>56</td>
<td>40</td>
<td>33</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>29</td>
<td>12</td>
<td>112</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>43</td>
<td>55</td>
<td>78</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Turing Optical Flow</td>
<td>Pascal/Volta Motion Vectors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------</td>
<td>----------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Granularity</strong></td>
<td>Up to 4x4</td>
<td>Up to 8x8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Algorithm used</strong></td>
<td>Visual motion optimization</td>
<td>Encoding cost optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td>Robust to intensity changes</td>
<td>Sensitive to intensity changes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>Close to true motion</td>
<td>May deviate from true motion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Low average EPE (end-point error)</td>
<td>Higher EPE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NVIDIA OPTICAL FLOW SDK
NVIDIA OPTICAL FLOW SDK

- New Optical Flow C-API
- Scalable, accommodates needs of future hardware
- Linux, Windows 8.1, 10, server, ...
- DirectX, CUDA interoperability
- OpenCV
- Public released - Feb 2019
- Legacy ME-only mode API continues to be supported
OPTICAL FLOW API

Basic functionality

Main Functionality (nvOpticalFlowCommon.h)

NV_OF_STATUS(NVOFAPI* PFNNVOFINIT) (NvOFHandle hOf, const NV_OF_INIT_PARAMS *initParams);

NV_OF_STATUS(NVOFAPI* PFNNVOFEXECUTE) (NvOFHandle hOf, const NV_OF_EXECUTE_INPUT_PARAMS *executeInParams,
   NV_OF_EXECUTE_OUTPUT_PARAMS *executeOutParams);

typedef NV_OF_STATUS(NVOFAPI* PFNNVOFDESTROY) (NvOFHandle hOf);

CUDA and DirectX buffer management

nvOpticalFlowCuda.h & nvOpticalFlowD3D11.h
REUSABLE CLASSES

NvOF
Base class for all core functionality

NvOFCUDA
Input and output in CUDA buffers

NvOFD3D11
Input and output in DirectX buffers
USE VIA OPENCV

Mat frameL = imread(pathL, IMREAD_GRAYSCALE);
Mat frameR = imread(pathR, IMREAD_GRAYSCALE);
GpuMat d_flowL(frameL), d_flowR(frameR), d_flow;
Mat flowx, flowy, flowxy;
int gpuId = 0;
int width = frameL.size().width, height = frameL.size().height;

Ptr<cuda::NvidiaOpticalFlow> OpticalFlow = cuda::NvidiaOpticalFlow::create(performance);
OpticalFlow->calc(frameL, frameR, d_flow);
d_flowL.download(flowxy);

Ptr<cuda::FarnebackOpticalFlow> OpticalFlow = cuda::FarnebackOpticalFlow::create();
OpticalFlow->calc(d_flowL, d_flowR, d_flow);
d_flowL.download(flowxy);
d_flow.download(flowxy);
BENCHMARKS
OPTICAL FLOW QUALITY
Evaluation Methodology

- **Objective quality**
  - KITTI 2012/2015, Sintel, Middlebury
  - Average end point error (EPE)
  - Percentage of outliers - background, foreground and all

- **Subjective quality**
  - Flow maps
  - Frame-rate-up-conversion (video interpolation)
OPTICAL FLOW QUALITY

EPE - KITTI 2015

Avg. EPE - Lower is better

- EPE = End-point error = Euclidian distance between OF vector & ground truth
- Non-occluded EPE
- Occluded EPE higher but same trend
- KITTI 2012 EPE = 2.31
- Sintel EPE = 8
## OPTICAL FLOW QUALITY

### Outliers - KITTI 2015

<table>
<thead>
<tr>
<th>Method</th>
<th>Background Outliers (%)</th>
<th>Foreground Outliers (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEGACY ME-ONLY MODE</td>
<td>31.09%</td>
<td>43.01%</td>
</tr>
<tr>
<td>OF RAW</td>
<td>21.33%</td>
<td>36.37%</td>
</tr>
<tr>
<td>OF POST-PROCESSED</td>
<td>16.76%</td>
<td>27.57%</td>
</tr>
<tr>
<td>PWC-DC</td>
<td>21.21%</td>
<td>42.29%</td>
</tr>
<tr>
<td>FLOWNET2</td>
<td>21.08%</td>
<td>23.57%</td>
</tr>
</tbody>
</table>

**Outlier** = Euclidian distance > 3 between OF vector and ground truth
OPTICAL FLOW QUALITY

Subjective Quality

• NVIDIA frame-rate-up-conversion
  • Video frame interpolation
  • ME-only mode (8×8), optical flow (4×4), optical flow with post-processing (1×1)
  • Subjective and objective quality comparison

• Results
  • Raw optical flow (4x4) based video interpolation better than ME-only mode (8x8) interpolation
  • Some video quality improvement with OF-post-processed (1x1) - content-dependent
VIDEO FRAME INTERPOLATION

Original 30 fps video
VIDEO FRAME INTERPOLATION

Upconverted 60 fps video
PERFORMANCE

- 3 presets
- Fast/Medium - no CUDA processing
- Slow - pre/post-processing in CUDA
- Performance scales with resolution
- Cost calculation in CUDA (enable only if needed)

Optical Flow quality vs performance

![Graph showing performance and average EPE.png](image-url)
NV OF time per frame: 2ms
GPU Utilization: 7%

Farneback's OF time per frame: 8ms
GPU Utilization: 13%

LK's OF time per frame: 25ms
GPU Utilization: 10%
END-TO-END APPLICATIONS
END-TO-END USE-CASES

Applications

• Video comprehension/classification
  • 2x better accuracy compared to no optical flow with UCF-101
  • Makes OF-assisted-video-comprehension usable
• Optical-flow-assisted video inter/extrapolation
  • Objective and subjective quality comparable to FlowNet2
  • Turing enables real-time optical-flow-assisted video interpolation
OPTICAL FLOW-ASSISTED VIDEO CLASSIFICATION
VIDEO CLASSIFICATION
Enables world class classification accuracy with real time performance

- Turing hardware:
  - Optical Flow reduces error rates by 2x
  - 20+ streams 720p inference

Image only video classification has high error rates

Optical flow significantly reduces error rates, but DL based OF is unusably slow
TURING OPTICAL FLOW

High quality video frame interpolation at 4K in real-time

Video Interpolation
Performance vs quality – 2160p streams

- 60 fps ➔ 120 fps at 4K in real-time
- 7x perf vs FlowNet2
- 1 dB better objective quality (PSNR) than FlowNet2-assisted interpolation
- Similar visual quality as FlowNet2-assisted interpolation
ROADMAP
ROADMAP
Optical Flow SDK 1.1

➢ Q3 2018

➢ Improved quality via post-processing

➢ 1x1 flow vectors

➢ Integration into DALI, Pytorch and other DL frameworks
RESOURCES


Support: video-devtech-support@nvidia.com


Connect with Experts (CE9103): Wednesday, March 20, 2019, 3:00 pm