

GSoC 2021 Ideas

Projects that we maintain:

FFmpeg-QSV, FFmpeg-NN: QSV and DNN in <https://ffmpeg.org/>

gstreamer-vaapi: <https://cgit.freedesktop.org/gstreamer/gstreamer-vaapi/>

gstreamer-MediaSDK: <https://cgit.freedesktop.org/gstreamer/gst-plugins-bad/tree/sys/msdk>

libxcam: <https://github.com/intel/libxcam>

Projects where we actively participate:

FFmpeg vaapi and filters

Gstreamer plugins

Async support for TensorFlow backend in FFmpeg

Description: FFmpeg DNN (deep neural network) module is to enable deep learning based FFmpeg filters with OpenVINO, TensorFlow and Native as its backends. The OpenVINO backend has supported async+batch execution for model inference with commit e67b5d and 64ea15 etc, it shows good performance gain. This project focuses on the async support for TensorFlow backend which invokes TensorFlow C library for model loading and inference. The main difference between the two backends is that OpenVINO API has async execution function while TensorFlow C API does not expose async function, so we need to support async execution by multiple threads within TensorFlow backend with sync function TF_SessionRun. (it is a plus if batch mode is also supported.)

Difficulty: Medium

Skill Required: C, multiple threads, DNN knowledge, git

possible mentor:

Guo, Yejun (yejun.guo@intel.com)

Fu, Ting (ting.fu@intel.com)

VVC codec parser in Gstreamer

Description: codec parsers are a core part of the gstreamer hardware decoder. They are used in gstvappi and gstcodecs to split frames, get slice level information. Since the vvc spec is finalized on 6 July 2020.

It's a good time to start work on the vvc parser for gstreamer. You need ref both vvc

spec(<https://www.itu.int/rec/T-REC-H.266/en>) and gstreamer h265

parser(<https://gitlab.freedesktop.org/gstreamer/gst-plugins-bad/-/blob/master/g...>) to implement it.

Difficulty: Medium

Skill Required: gstreamer, C, git, media codec

possible mentor:

He, Junyan (junyan.he@intel.com)

Xiang, Haihao (haihao.xiang@intel.com)

Equirectangular Projection (ERP) and Cubemap Projection (CMP) conversion

Description: High resolution 360-degree videos provide viewers with an immersive experience in virtual reality (VR) and augmented reality (AR) technologies. The 360-degree stitched videos normally represented in a native projection format, e.g., equirectangular (ERP), to compress and deliver such video content requires much higher bandwidth. It's useful to convert ERP into another projection format, e.g., cubemap (CMP) before encode 360-degree stitched frame by video codecs, the CMP projection format would potentially improve the representation efficiency and coding performance.

CMP is very popular and has been widely used. The goal of this project is to implement and optimize ERP and CMP conversion algorithm by GLES.

Difficulty: Medium

Skill Required: : C++, GPU parallel programming, GLES

possible mentor:

Zong, Wei (wei.zong@intel.com)

Liu, Yinhang(yinhang.liu@intel.com)