GSoC Ideas 2023

Projects that we maintain:
ffmpeg-vaapi, fmpeg-qsv, ffmpeg-nn: QSV and DNN in https://ffmpeg.org/
gstreamer-vaapi: https://gitlab.freedesktop.org/gstreamer/gstreamer.git (subprojects/gstreamer-vaapi)
gstreamer-msdk: https://gitlab.freedesktop.org/gstreamer/gstreamer.git (subprojects/gst-plugins-bad/sys/msdk/)
gst-va: https://gitlab.freedesktop.org/gstreamer/gstreamer.git (subprojects/gst-plugins-bad/sys/va/)
libxcam: https://github.com/intel/libxcam

Projects where we actively participate:
ffmpeg filters
Gstreamer plugins

LIST OF PROJECT IDEAS:

GPU accelerated PSNR filter on FFmpeg

Description:
PSNR check is one of Video testing method. FFmpeg has CPU based PSNR filter, but its' performance is limited. When it is used in heavy video CI testing, it is always the performance bottleneck. PSNR algorithm is highly parallel, which is suitable for optimizing with GPU. If we are testing hardware codec, we can build full gpu piple without data transmission between device memory and system memory, and it makes testing even faster.
We will add PSNR filter based on OpenCL to ffmpeg. FFmpeg already has OpenCL and Vulkan filter, so we can reuse their basic framework to add a new filter.
Difficulty: Medium
Hardware Required: computer with GPU
Skill Required: C programming, OpenCL, and FFmpeg experience.
Possible mentor: Wenbin Chen (wenbin.chen@intel.com)

Panoramic segmentation for libXCam 360-degree surround view stitching images

Description:
Panoramic segmentation is to detect and segment **all objects** in the picture, including the **background**, and **distinguish different instances**. In the process of autonomous driving, panoramic segmentation provides information about the trafic condition on the road, as well as the detection of landmark and traffic signs. These information can provide a complete surrounding understanding for autonomous driving system.
The result of panoramic segmentation is to transform the 360-degree field-of-view image into several color blocks, and each color block represents one part of the image.
Difficulty: Medium
Hardware Required: GPU/CPU
Skill Required: C++ programming, DNN, Image segmentation
Possible mentor: Zong Wei (wei.zong@intel.com)

H.266/VVC bit writer implementation in Gstreamer

Description:
The Versatile Video Coding, also known as H.266, is the next generation video compression standard finalized on 6 July 2020. It will be and already is the new hot topic for the video related

industry. Gstreamer as a widely used media framework, it wants to support all kinds of codec types, and so a VVC bit writer is mandatory. In Gstreamer, the bit writer can help and ease the encoder implementation a lot. As a bit writer, it needs to generate the bit stream which conforms to the SPEC based on the according SPS/PPS/SLICE information, which is the exactly inverse job of VVC parser (already implemented in Gstreamer).

This bit writer will be added as a helper library in Gstreamer. Given the SPS, PPS or other information, user should be able to use these helper functions to generate the VVC bit stream. And then the new bit stream can also be parsed correctly again and the parsed information should be identical to the original information.

Difficulty: Easy

Hardware Required: Any computer

Skill Required: C programming and Gstreamer experience.

Possible mentor: He Junyan (junyan.he@intel.com)

DNN model based frame interpolation filter in FFmpeg

Description:

Video frame interpolation is using the consecutive frames to generate intermediate(non-existent) frame between them. Now FFmpeg provide a filter, called framerate, to simply add two frames after multiplying factors to produce the intermediate one. The outputs of which is quite rough. As we investigated, the related DNN algorithm would promise output frame with much better quality than traditional algrithms. Therefore, we would like introduce DNN method in FFmpeg filter to generate more satified results.

The model should be an OpenVINO backend model. We support OpenVINO as one of FFmpeg inference backend already. The framerate filter can also be a reference for adding such filter. The mainly work would be in model choosing and conversion, filter adding/modifying, model inference supporting.

Difficulty: Medium

Hardware Required: Any computer

Skill Requires: C/C++ programming, DNN, Python, FFmpeg experience.

Possible mentor: Ting Fu (ting.fu@intel.com)