Sujata Tibrewala
OneAPI Community Development Manager
Intel

SUJATA.TIBREWALA@INTEL.COM
@SUJATATIBRE
Hello
Alessandro de Oliveira Faria
Founder of the Company
OITI TECHNOLOGIES
Making banking secure via bio metrics application built using oneAPI and DPC++ based on SYCL/C++
Notices and Disclaimers

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Intel technologies may require enabled hardware, software or service activation.

No product or component can be absolutely secure.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
- Searches and compares in more than 20 million in less than 4 seconds
- 100% in cloud
- 0,0000000007 False Accept Rate
- Fight Fraud and Authenticate Users
- Robust Algorithm (low image quality)
Protection of people, financial market and retail from identity fraud

Face capture and document scanning

Useful for mobile application for opening accounts and transactions

Biometrics Cluster

Identifying millions of users in just 3s

Scalable operating power and high availability

Intel Software Innovator
Certiface protects honest people, financial market and retail from fraud

When Certiface finds a linked face with many different documents (Tax ID), in the centralized base with millions of users, the system triggers an alert in real time.
Certiface uses several algorithms (proprietary and opensource) to process a robust solution based on computer vision.

Even with dark, low-quality photos, certiface can process facial biometrics.
Introducing oneAPI

Unified programming model to simplify development across diverse architectures

• Unified and simplified language and libraries for expressing parallelism
• Uncompromised native high-level language performance
• Based on industry standards and open specifications
• Interoperable with existing HPC programming models

These organizations support the oneAPI initiative ‘concept’ for a single, unified programming model for cross-architecture development. It does not indicate any agreement to purchase or use of Intel’s products.
oneAPI Industry Initiative

- oneAPI Industry Specification
  - A standards based cross-architecture language, DPC++, based on C++ and SYCL
  - Powerful APIs designed for acceleration of key domain-specific functions
  - Low-level hardware interface to provide a hardware abstraction layer to vendors
  - Enables code reuse across architectures and vendors
  - Open standard to promote community and industry support
- Technical Advisory Board
- oneAPI Industry Brand

Some capabilities may differ per architecture and custom-tuning will still be required. Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products.
Data parallel C++
Standards-based, Cross-architecture Language

Language to deliver uncompromised parallel programming productivity and performance across CPUs and accelerators
DPC++ = ISO C++ and Khronos SYCL and Extensions
Allows code reuse across hardware targets, while permitting custom tuning for a specific accelerator
Open, cross-industry alternative to single architecture proprietary language

Based on C++
Delivers C++ productivity benefits, using common and familiar C and C++ constructs
Incorporates SYCL* from the Khronos Group to support data parallelism and heterogeneous programming

Community Project to drive language enhancements
Extensions to simplify data parallel programming
Open and cooperative development for continued evolution
DPC++ extensions including Unified Shared Memory are being incorporated into upcoming versions of the Khronos SYCL standard.

Some capabilities may differ per architecture and custom-tuning will still be required. Other accelerators to be supported in the future.


Intel® oneAPI Products *(beta)*

Distributed through a core toolkit and a complementary set of add-on domain-specific toolkits

Includes DPC++ compatibility tool for code migration along with advanced performance analysis and debug tools

Beta Available Now

Visit software.intel.com/oneapi for more details
oneAPI Specification

Key domain-specific functions to accelerate compute intensive workloads

Custom-coded for supported architectures

Refer to http://software.intel.com/en-us/articles/optimization-notice for more information regarding performance and optimization choices in Intel software products
Intel® oneAPI Video Processing Library

Accelerated Video Processing with a Unified Programming API

Jump to: Documentation & Code Samples | Key Specifications | Get Help

State-of-the-Art Video Codecs

The Intel® oneAPI Video Processing Library lets developers add high-speed, real-time transcoding, decoding, and encoding to their applications. Its single video API provides direct access to advanced Intel® CPU and GPU instructions, and gives you total control of the video hardware for their processing needs.

The library is perfect for applications spanning broadcasting, OTT and VOD, in-cloud gaming, and remote desktop solutions.

- Includes high-performance, hardware-accelerated AVC, HEVC, and AV1 codecs
- Supports deployment on CPUs and GPUs
- Flexible API enables developers to maximize application exposure to Intel® hardware

Develop, Test, and Run Your oneAPI Code in the Cloud

Get what you need to build and optimize your oneAPI projects for free. With an Intel® DevCloud account, you get 120 days of access to the latest Intel® hardware—CPUs, GPUs, FPGAs—and Intel oneAPI tools and frameworks. No software downloads. No configuration steps. No installations.

Get Access

Download Intel oneAPI Video Processing Library as Part of the Intel® oneAPI Base Toolkit

Get It Now
Documentation & Code Samples

Get Started

- Intel® oneAPI Video Processing Library\(^{(\text{Beta})}\)

Documentation

- Release Notes
- System Requirements

View All Documentation

Code Samples

Learn how to access oneAPI code samples in a tool command line or IDE.

- Simple Encode
- Simple Decode
- Decode with an Accelerator Selection
- Decode with Video Post-Processing
- Demux and Decode
- Memory Integration
Fast Decoder in C++ with libVPL

```c++
uint8_t *pbs = new uint8_t[BUFFER_SIZE];
FILE* fInput = fopen(imageVideo.c_str(), "rb");
if (!fInput)
{
    printf("Error: could not open input file \"%s\", imageVideo.c_str());
    return 1;
}

vplmem* image = nullptr;
bool bdrain_mode = false;
vplWorkstreamState decode_state = VPL_STATE_READ_INPUT;
int frameCount = 0;
double elapsedTime = 0.0;
for (; decode_state != VPL_STATE_END_OF_OPERATION && decode_state != VPL_STATE_ERROR; decode_state = decoder.GetState())
{
    printf("> Frame %d \n", frameCount);
    uint32_t bs_size = 0;
    if ((decode_state == VPL_STATE_READ_INPUT) && (!bdrain_mode))
    {
        bs_size = (uint32_t)fread(pbs, 1, BUFFER_SIZE, fInput);
    }
    if (bs_size == 0 || decode_state == VPL_STATE_INPUT_BUFFER_FULL)
    {
        bdrain_mode = true;
    }

    auto decTimeStart = std::chrono::system_clock::now();
    if (bdrain_mode)
    {
        image = decoder.DecodeFrame(nullptr, 0);
    }
    else
    {
        image = decoder.DecodeFrame(pbs, bs_size);
    }
    DisplayOutput(image);
    auto decTimeEnd = std::chrono::system_clock::now();
    std::chrono::duration<double> t = decTimeEnd - decTimeStart;
    elapsedTime += t.count();
    printf("%0.2f sec(s)\n", t.count());
    if (!image) continue;
    frameCount++;
}
fclose(fInput);
delete[] pbs;
```
void DisplayOutput(vpl_mem* img) {
    cv::Mat img_nv12, frame, blob;
    vpl_cpu_image handle = {0};
    vpl_image_info desc;
    unsigned char *data;

    // Read image description (width, height, etc) from vpl memory
    vpl_get_image_info(img, &desc);
    // Access data in read mode
    vpl_status err = vpl_map_image(img, VPLM_ACCESS_MODE_READ, &handle);

    // Need to rearrange data because of stride size
    data = new unsigned char[desc.height * 3/2 * desc.width];

    size_t pitch0 = handle.planes[0].stride;
    size_t pitch1 = handle.planes[1].stride;
    for(size_t y = 0; y < desc.height; y++){
        memcpy(data + (desc.width * y), handle.planes[0].data + (pitch0 * y), desc.width);
    }
    for(size_t y = 0; y < desc.height/2; y++){
        memcpy(data + (desc.width * desc.height) + (desc.width * y), handle.planes[1].data + (pitch1 * y), desc.width);
    }
    img_nv12 = cv::Mat(desc.height * 3/2, desc.width, CV_8UC1, data);
    // Convert NV12 to BGRA format for displaying with OpenCV
    cv::cvtColor(img_nv12, frame, cv::COLOR_YUV2BGRA_NV12);

    if (frame.empty())
    {
        printf("Error in convert frame\n");
        exit(0);
    }

    // Create a 4D blob from a frame.
    Size inpSize(inpWidth > 0 ? inpWidth : frame.cols, inpHeight > 0 ? inpHeight : frame.rows);
    blobFromImage(frame, blob, scale, inpSize, Scalar(), false, false);

    net.setInput(blob);
    std::vector<Mat> outs;
    net.forward(outs, outNames);
    postprocess(frame, outs, outNames);
}

vpl_unmap_image(&handle);
delete data;
return;
source /opt/intel/inteloneapi/setvars.sh
/usr/bin/c++ -I/opt/intel/inteloneapi/vpl/latest/include \ 
   -o main.cpp.o -c /home/u31713/lab/antispoofing_oneAPI/main.cpp 
/usr/bin/c++ /home/u31713/lab/antispoofing_oneAPI/main.cpp.o \ 
   -o main /opt/intel/inteloneapi/vpl/latest/lib/libvpl.so \ 
   /opt/intel/inteloneapi/vpl/latest/lib/libvplmemory.so \ 
   /opt/intel/inteloneapi/vpl/latest/lib/libopencv_world.so
• I take Linux seriously, researches and works with biometrics and computer vision since 1998.
• SUSE and openSUSE Leap Linux distribution
• openCV optimized for Intel processors
• Based on the Simple Decode example
Create folder, and run cmake command

Build command
Create H.264 DECODER

```cpp
vpl::Workstream decoder(VPL_TARGET_DEVICE_DEFAULT, VPL_WORKSTREAM_DECODE);
```

Use GPU if available

```cpp
decoder.SetConfig(VPL_PROP_SRC_BITSTREAM_FORMAT, VPL_FOURCC_H264);
```

How To Work

```cpp
vplm_mem* image = nullptr;
```

Read FRAME

```cpp
stream.read(reinterpret_cast(buffer.data()), buffer.size());
```

Decoder frame

```cpp
image = decoder.DecodeFrame(buffer.data(), stream.gcount());
```

Access data in read mode

```cpp
vplm_status err = vplm_map_image(image, VPLM_ACCESS_MODE_READ, &handle);
```
Summary

Certiface technology was designed to be used in Banking system to combat fraud and protect honest people by using technology to differentiate between a living person and a recorded video.

Certiface AntiSpoofing use oneAPI and one VPL for fast decode video.

oneVPL Library enables allows the development of real-time transcoding, decoding and encoding, high-speed direct access to advanced Intel CPU and GPU instructions.

oneAPI unifies and simplifies programming across CPUs and accelerators, delivering developer productivity across architectures and vendors.
INNOVATOR PROGRAM: RECOGNIZING YOUR WORK IN THE COMMUNITY (DEVMESS.INTEL.COM -> MEMBER PROGRAMS)

- **Top Tier Experts Who Innovate & Like to Share Work Go Under NDA**
- **Access to Intel HW/SW & Intel Expertise**
- **Early Experiments, Prototypes, Testing & Research**
- **Innovator Spotlight**
  - Shared Projects - Mesh
  - Technical Articles
  - Media Articles
  - Event Demos
  - Devs Trained
  - Product Feedback
- **Innovator Speakerships & Demos at Major Conferences and Public Events**

Projects Innovators are passionate about.
Certiface Anti-Spoofing

Alessandro de Oliveira Faria
Sorocaba, SP

Certiface AntiSpoofing use oneAPI for fast decode video for perform liveness detection with inference. The system is capable of spotting fake faces and performing anti-face spoofing in face recognition systems.

Project status: Under Development

Artificial Intelligence, HPC
Member Programs

Want to Join? Apply Now!
If you'd like to join Intel's exclusive developer community, apply for membership now.

- Learn More About The Intel Software Innovator
- Learn More About The Intel Student Ambassador
- Read About The Applications Process. Learn more here.
- To Apply - First create an account and login to Intel DevMesh
Intel is committing $50M to support healthcare during the COVID-19 pandemic. $40M will be allocated to the Response and Readiness Initiative, which aims to accelerate access to technology at the point of patient care, speed scientific research, and ensure access to online learning for students. The remaining $10M will be used as an innovation fund to enable and fuel new ideas and technologies with external partners and employee-led relief projects to manage or reduce the impact of COVID-19.
THANK YOU

Alessandro de Oliveira Faria
Sujata Tibrewala
SUJATA.TIBREWALA@INTEL.COM
@SUJATATIBRE