Image Special Interest Group

Session 3: oneIPL spec v0.7 APIs overview
December 14, 2023

The Image SIG discussion rules

DO NOT share any confidential information or trade secrets with the group

DO keep the discussion at a High Level

- Focus on the specific Agenda topics
- We are asking for feedback on features for the oneIPL specification (e.g., requirements for functionality and performance)
- We are NOT asking for the feedback on any implementation details

Please submit the feedback in writing on GitHub per Contribution Guidelines at spec.oneapi.io. This will allow Intel to further upstream your feedback to other standards bodies, including The Khronos Group SYCL specification.

oneIPL – oneAPI interface for image processing

- SYCL 2020 – based on C++17
- oneIPL primitives – classes for data abstractions + functional API
- API shall be compatible with SYCL 2020 compliant compiler implementation
- oneIPL provides SYCL API for image processing functionality working on XPU.
- oneIPL API provides C++ abstraction over image data, which maps to the most accelerated memory available for format and data type.
- oneIPL provision spec v0.7 is published. (extra updates might be done).

oneIPL – oneAPI interface for image processing

- API was simplified compared to the one discussed a year ago.
- src and dst arguments are of class Image type and defining either image or region of interest (ROI).
- spec argument defines Image-independent parameters like filter kernel size, scalar factors, etc.
- Image-dependent arguments follows before dependencies (e.g., border_val defines the border pixel value required for constant borders in algorithms supporting such border)
- Dependencies argument defines a vector of events of the kernels which shall be completed before the execution of the algorithm

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event filter_box(sycl::queue&                          queue,
                       SrcImageT&                            src,
                       DstImageT&                            dst,
                       const gaussian_spec<ComputeT>&        spec,
                       const typename SrcImageT::pixel_t& border_val   = {},
                       const std::vector<sycl::event>&    dependencies = {})
```

oneIPL spec updates plan

oneIPL Spec v0.6

Transformations:
- Resize bilinear
- Resize bicubic
- Resize lanczos
- Resize supersampling
- Horizontal mirror

Filters:
- Sobel 3x3
- Gaussian

Conversions and other operations:
- gray<->rgb(a)
- I420<->rgb(a)
- nv12<->rgb(a)
- rgbp<->rgb(a)
- rgb<->rgba
- Convert
- Copy
- Normalize

oneIPL Spec v0.7

Batch operations:
- Batch mirror
- Batch resize bilinear

Transformations:
- Resize nearest
- Warp nearest
- Transposition

Filters:
- Sobel generic
- Bilateral
- Box
- Convolution
- Separable
- Median

Conversions and other operations
- Color twist
- Swap channels
- Magnitude

To be Added in next versions

Batch operations:
- Batch resize nearest
- Batch warp nearest
- Batch color conversions

Transformations:
- Warp bilinear
- Warp bicubic

Filters:
- Erode
- Dilate

Other operations:
- Histogram
- Threshold
- Gamma correction

Batch APIs in oneIPL spec 0.7

```
template <typename SrcBatchT,
          typename DstBatchT>
sycl::event mirror_batch(sycl::queue&                    queue,
                         SrcBatchT&                      src,
                         DstBatchT&                      dst,
                         const mirror_spec&              spec         = {},
                         const std::vector<sycl::event>& dependencies = {})
```

```
template <typename ComputeT = float,
          typename SrcBatchT,
          typename DstBatchT>
sycl::event resize_bilinear_batch(sycl::queue&                    queue,
                                  SrcBatchT&                      src,
                                  DstBatchT&                      dst,
                                  const resize_bilinear_spec&     spec         = {},
                                  const std::vector<sycl::event>& dependencies = {})
```

oneIPL batch API usage example

Image data pointers (src_ptrs and dst_ptrs) and image descriptors pointers should be accessible from the device.

```
// Allocate shared memory for batch images metadata
auto src_image_descriptors = sycl::malloc_shared<image_descriptor<layouts::channel4, std::uint8_t>>(batch_size, queue);
auto dst_image_descriptors = sycl::malloc_shared<image_descriptor<layouts::channel4, std::uint8_t>>(batch_size, queue);

// Fill batch images metadata
for (std::size_t i{ 0U }; i < batch_size; ++i) {
    src_image_descriptors[i] =
        image_descriptor<layouts::channel4, std::uint8_t>{ src_ptrs[i], src_pitches[i], src_sizes[i], src_roi_rects[i] };
    dst_image_descriptors[i] =
        image_descriptor<layouts::channel4, std::uint8_t>{ dst_ptrs[i], dst_pitches[i], dst_sizes[i], dst_roi_rects[i] };
}

// Create source and destination batches
batch<layouts::channel4, std::uint8_t> src_batch{ src_image_descriptors, batch_size, src_max_roi_size };
batch<layouts::channel4, std::uint8_t> dst_batch{ dst_image_descriptors, batch_size, dst_roi_size };

// Resize batch of images
auto event1 = resize_bilinear_batch(queue, src_batch, dst_batch);
auto event2 = mirror_batch(queue, dst_batch, dst_batch, {event1});
event2.wait()

// Free the allocated memory
sycl::free(src_image_descriptors, queue);
sycl::free(dst_image_descriptors, queue);
```

Transformation APIs in oneIPL spec 0.7

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event resize_nearest(sycl::queue&                    queue,
                           SrcImageT&                      src,
                           DstImageT&                      dst,
                           const resize_nearest_spec&      spec         = {},
                           const std::vector<sycl::event>& dependencies = {})

template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event warp_nearest(sycl::queue&                          queue,
                         SrcImageT&                            src,
                         DstImageT&                            dst,
                         const warp_nearest_spec<ComputeT>&    spec         = {},
                         const typename SrcImageT::pixel_t&    border_val   = {},
                         const std::vector<sycl::event>&       dependencies = {})
```

Transformation APIs in oneIPL spec 0.7

```
template <typename SrcImageT,
          typename DstImageT>
sycl::event transpose(sycl::queue&                    queue,
                      SrcImageT&                      src,
                      DstImageT&                      dst,
                      const transpose_spec&           spec         = {},
                      const std::vector<sycl::event>& dependencies = {})
```

Filtering APIs in oneIPL spec 0.7

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT >
sycl::event bilateral(sycl::queue&                          queue,
                      SrcImageT&                            src,
                      DstImageT&                            dst,
                      const bilateral_spec<ComputeT>&       spec,
                      const typename SrcImageT::pixel_t&    border_val   = {},
                      const std::vector<sycl::event>&       dependencies = {})
```

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event filter_box(sycl::queue&                          queue,
                       SrcImageT&                            src,
                       DstImageT&                            dst,
                       const filter_box_spec&                spec,
                       const typename SrcImageT::pixel_t&    border_val   = {},
                       const std::vector<sycl::event>&       dependencies = {})
```

Filtering APIs in oneIPL spec 0.7

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event filter_convolution(sycl::queue&                                queue,
                               SrcImageT&                                  src,
                               DstImageT&                                  dst,
                               const filter_convolution_spec<ComputeT>& spec,
                               const typename SrcImageT::pixel_t&       border_val   = {},
                               const std::vector<sycl::event>&          dependencies = {})
```

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event median(sycl::queue&                          queue,
                   SrcImageT&                            src,
                   DstImageT&                            dst,
                   const median_spec&                    spec,
                   const typename SrcImageT::pixel_t& border_val   = {},
                   const std::vector<sycl::event>&    dependencies = {})
```

Filtering APIs in oneIPL spec 0.7


```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event separable(sycl::queue&                          queue,
                      SrcImageT&                            src,
                      DstImageT&                            dst,
                      const separable_spec<ComputeT>&       spec,
                      const typename SrcImageT::pixel_t&    border_val   = {},
                      const std::vector<sycl::event>&       dependencies = {})
```

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event sobel(sycl::queue&                          queue,
                  SrcImageT&                            src,
                  DstImageT&                            dst,
                  const sobel_spec&                     spec         = {},
                  const typename SrcImageT::pixel_t&    border_val   = {},
                  const std::vector<sycl::event>&       dependencies = {})
```

Other APIs in oneIPL spec 0.7

```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event magnitude(sycl::queue&                    queue,
                      SrcImageT&                      src,
                      DstImageT&                      dst,
                      const magnitude_spec&           spec         = {},
                      const std::vector<sycl::event>& dependencies = {})
```

```
template <typename SrcImageT,
          typename DstImageT>
sycl::event swapchannels(sycl::queue&                      queue,
                         SrcImageT&                        src,
                         DstImageT&                        dst,
                         const swapchannels_spec&          spec,
                         const typename DstImageT::data_t dst_fill_value =
max_color_v<typename DstImageT::data_t>,
                         const std::vector<sycl::event>&  dependencies   = {})
```

Conversions APIs in oneIPL spec 0.7


```
template <typename ComputeT = float,
          typename SrcImageT,
          typename DstImageT>
sycl::event color_twist(sycl::queue&                          queue,
                        SrcImageT&                            src,
                        DstImageT&                            dst,
                        const color_twist_spec<ComputeT>& spec,
                        const std::vector<sycl::event>&   dependencies = {})
```

Thank you for attending!

- If you have content to post to oneAPI.io, please let us know
- Please feel free to extend invitations to others to join the Image SIG or other oneAPI Community Forums
- Join oneAPI on LinkedIn: https://www.linkedin.com/groups/14241252/