日程安排

9:30 -10:15　英特尔：oneAPI 生态与现状介绍

10:15-10:20　互动有礼

合作伙伴分享

10:20-10:40	汇川视觉：oneAPI 开发套件在 CV 视觉领域应用案例分享
10:40-11:00	卫宁健康：人工智能实验室医疗 AI 探索和实践
11:00-11:20	恒安嘉新：Intel® oneAPI 加速基于 AI 的公民证件保护应
11:20-11:40	复旦大学：oneAPI 驱动机器学习加速
11:40-11:45	互动有礼
11:45-13:00	午餐
13:00-13:05	互动有礼

黑客松大赛

13:05-13:20	英特尔：oneAPI 黑客松大赛介绍
13:20-13:30	南开大学：基于 oneAPI 的大规模图计算异构加速框架设计
13:30-13:40	中国科学技术大学：基于T2S的oneAPI张量计算后端
13:40-13:50	兴业数字金融：基于机器学习的欺诈风险识别
13:50-13:55	互动有礼

技术培训

13:55-15:25	英特尔：oneAPI 异构计算编程模式
15:25-15:30	互动有礼
15:30-17:00	英特尔：使用 Intel® 优化的AI框架，释放 Intel® XPU 的动能，加速 AI
17:00-17:05	问卷 抽奖

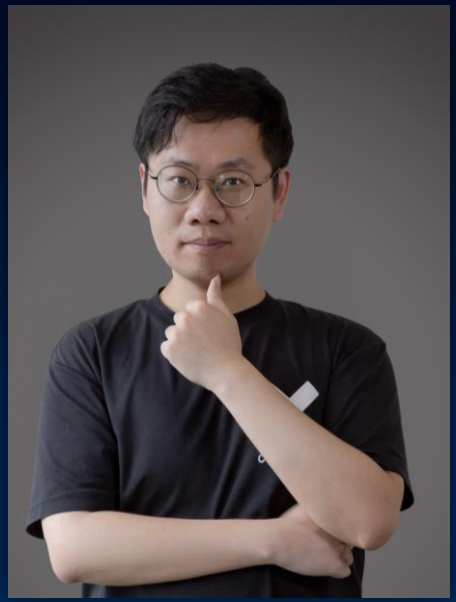
葛锦洲

CV 算法工程师

南京汇川视觉

oneAPI 开发套件在 CV 视觉领域应用案例分享

- 使用 DPCT 工具实现光度立体算子 CUDA 代码到 DPC++ 核显代码的迁移，快速完成 OCR 场景图像预处理的异构部署；
- 使用 vTune 工具协助分析性能瓶颈，实现 Sobel 算子的异构优化；
- 使用 Intel® 高性能库 IPP 优化 dft 算子的效率性能；
- 使用 Intel® 编译器提升形状匹配算子效率性能，提升工业场景缺陷检测预处理、标定对位场景定位效率性能



葛锦洲，南京汇川视觉 CV 算法工程师，有较丰富的视觉算法应用设计及性能优化实战经验

oneAPI 开发套件在 CV 视觉领域应用案例分享

汇报人：葛锦洲

INOVANCE

INOVANCE

1项目概述

Jinovision高效、开放、智能的工业视觉平台软件

- 高效
  - 直观易用的流程图式开发模式
  - 组态式界面开发
  - 轻量级配方式一键换型
- 开放
  - 支持自定义元件、插件
  - 支持元件级、任务级API调用
  - 后端服务、前端界面分体式设计
  - 与汇川体系控制产品底层数据互通
- 智能
  - 开箱即用的OCR基模型
  - 基于预训练模型的AI无监督算法
  - Intel CPU/iGPU指令集加速技术

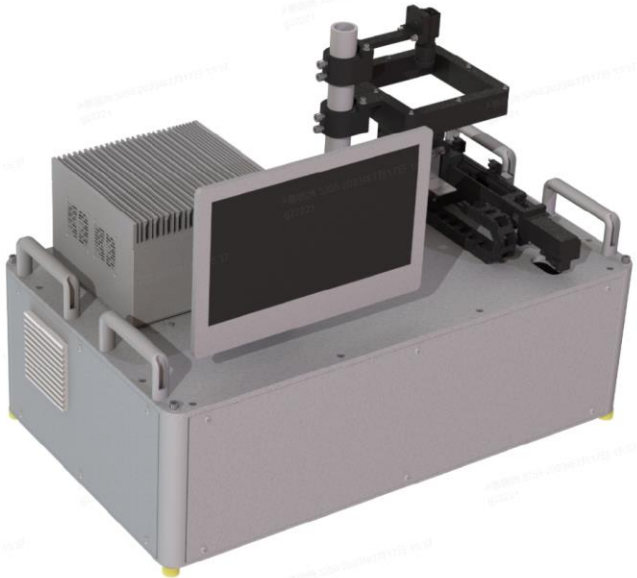
INOVANCE

1项目概述

光度立体OCR字符识别方案

采用光路立体方案，消除复杂纹理背景，凸显钢印字符信息，识别字符内容 。

- 核心技术
  - 光度立体
  - OCR基模型
- 性能、功能亮点
  - 规范字符场景模型开箱可用
  - 光度立体去除复杂纹理背景
- 可扩展场景
  - 汽车行业的铸造件检测
  - 医药行业线的钢印喷码检测



INOVANCE

1项目概述

自动对位&精度复测设备+PAC控制器
视觉运控一体控制器产品形态

基于PAC视控一体控制器的典型对位方案。

- 核心技术
  - 视控一体
  - 精度解耦
- 性能、功能亮点
  - 视控一体化编程
  - 通过精度解耦快速交付
- 可扩展场景
  - 各种对位类应用

INOVANCE

INOVANCE

2核心竞争力分析

产品名称	关键特性描述	优势
睛麟视觉平台	深度优化后的视觉算子，强劲性能表现	oneAPI开发套件中vTune工具分析算子运行瓶颈，IPP等高性能计算库深度优化算子运行性能
光度立体字符识别系统	光度立体去除复杂纹理，异构部署	oneAPI开发套件中DPCT工具实现光度立体算子CUDA代码到DPC++核显代码的快速迁移
对位视控一体系统	高精亚像素边缘提取，视控一体流畅性	oneAPI开发套件中sycl实现Sobel算子深度优化，intel C++编译器优化形状匹配算子效率

总结：intel oneAPI开发套件助力视觉平台算子性能提升。

INOVANCE

INOVANCE

Sobel算子优化

```cpp
auto start_time = std::chrono::steady_clock::now();
for (int i = 0; i < num_iters; i++)
{
    cv::Sobel(src, grad_x, CV_32F, 1, 0, ksize, scale, delta, cv::BORDER_REPLICATE);
    cv::Sobel(src, grad_y, CV_32F, 0, 1, ksize, scale, delta, cv::BORDER_REPLICATE);

    abs_grad_x = cv::abs(grad_x);
    abs_grad_y = cv::abs(grad_y);

    cv::add(abs_grad_x, abs_grad_y, dst);
}
auto end_time = std::chrono::steady_clock::now();
```

OpenCV在CPU上的部署代码

```cpp
for (int i = 0; i <= num_iters; i++)
{
  // waitForKernelCompletion();
   start_ct1 = std::chrono::steady_clock::now(); // start timer
   // apply sobel kernels
   q_ct1.submit([&](sycl::handler &cgh)
                { cgh.parallel_for<class compute_kernel>(
                      sycl::nd_range<3>(sycl::range<3>(1, 1, num_blks) *
                                            sycl::range<3>(1, 1, thd_per_blk),
                                        sycl::range<3>(1, 1, thd_per_blk)),
                      [=](sycl::nd_item<3> item_ct1){
                          apply_sobel_filter_sum(gradient_pixels, in,
                                                 scale, image_width, image_height,
                                                 sobel_kernel_x_gpu, sobel_kernel_y_gpu,
                                                 item_ct1);
                      }); })
       .wait();

  // waitForKernelCompletion();
   stop_ct1 = std::chrono::steady_clock::now(); // end timer

   if (i > 0)
       elapsed += std::chrono::duration<float, std::milli>(stop_ct1 - start_ct1)
                      .count();
}
```

SYCL在iGPU上的部署代码

INOVANCE

Sobel算子优化

Grouping: Computing Task

Computing Task	Work Size		Computing Task					Data Transferred		EU Array		
	Global ▼	Local	Total Time	Average Time	Instance Count	SIMD Width	SVM Usage Type	Size	Total, GB/sec	Active	Stalled	Idle
sobelDpcppDB::{lambda(sycl::_V1::handler&)#1}::operator	32000000 x 1 x 1	256 x 1 x 1	35.503ms	35.503ms	1	16		0 B	0.000	68.2%	10.0%	21.8%
clEnqueueMemcpyINTEL			23.553ms	7.851ms	3			288 MB	12.228	7.4%	70.7%	21.9%
[Outside any task]			0ms	0ms	0			0 B	0.000	16.3%	35.2%	48.6%

瓶颈分析：

1、线程调度达到32000000；

2、运行位宽为128位；

3、数据搬运量为288MB。

优化方案：

1、数据分块，优化线程调度开销；

2、增大运行位宽；

3、使用SLM优化数据搬运总量。

INOVANCE

Sobel算子优化

Grouping: Computing Task

Computing Task	Work Size		Computing Task					Data Transferred		EU Array		
	Global ▼	Local	Total Time	Average Time	Instance Count	SIMD Width	SVM Usage Type	Size	Total, GB/sec	Active	Stalled	Idle
sobelDpcppDB::{lambda(sycl::_V1::handler&)#1}::operator	32000000 x 1 x 1	256 x 1 x 1	35.503ms	35.503ms	1	16		0 B	0.000	68.2%	10.0%	21.8%
clEnqueueMemcpyINTEL			23.553ms	7.851ms	3			288 MB	12.228	7.4%	70.7%	21.9%
[Outside any task]			0ms	0ms	0			0 B	0.000	16.3%	35.2%	48.6%

Grouping: Computing Task

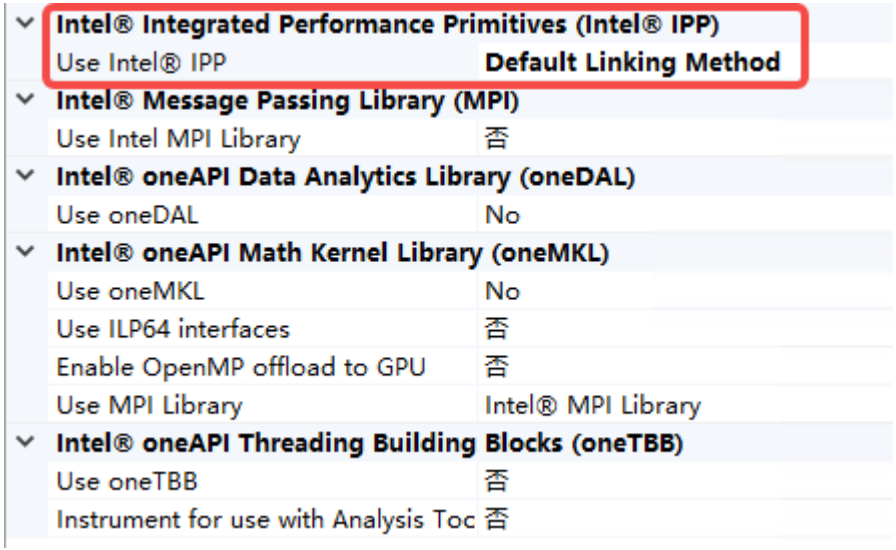
Computing Task	Work Size		Computing Task					Data Transferred		EU Array			Peak E
	Global ▼	Local	Total Time	Average Time	Instance Count	SIMD Width	SVM Usage Type	Size	Total, GB/sec	Active	Stalled	Idle	
main::{lambda(sycl::_V1::handler&)#1}::operator()(sycl::_V	8544 x 1456	32 x 8	27.617ms	27.617ms	1	32		0 B	0.000	75.6%	2.6%	21.8%	
clEnqueueMemcpyINTEL			2.608ms	2.608ms	1			32 MB	12.272	7.3%	69.3%	23.5%	
[Outside any task]			0ms	0ms	0			0 B	0.000	9.6%	15.3%	75.1%	

INOVANCE

dft算子优化

```
CV_EXPORTS_W void dft(InputArray src, OutputArray dst, int flags = 0, int nonzeroRows = 0);
```

OpenCV相关API




```
#include<ipps.h>
#include<ippi.h>
#include<ippcore.h>
```

```
IPPAPI (IppStatus, ippiDFTFwd_RToPack_32f_C1R,
                   ( const Ipp32f* pSrc, int srcStep,
                     Ipp32f* pDst, int dstStep,
                     const IppiDFTSpec_R_32f* pDFTSpec,
                     Ipp8u* pBuffer ))
```

IPP相关API

光度立体算子异构部署

DPCT工具将CUDA转换至DPC++部署方法

1、准备所需迁移CUDA代码。

从一个可以构建和运行的正在运行的 CUDA 项目开始。英特尔 DPC++兼容性工具寻找 CUDA 头文件，需确保头文件可以被工具访问。

2、迁移项目

要生成SYCL代码，将原始代码作为工具输入。

对于简单的项目，可以使用文件到文件迁移，可以选择一次迁移所有文件或一个一个地迁移文件。

对于复杂的项目，可以使用 Microsoft Visual Studio项目文件或 Make/Cmake 文件来构建编译数据库，用于迁移完整的项目。

3、检查转换后的代码

输出文件的注释包含帮助迁移无法自动迁移的任何剩余代码。检查转换后的代码，查看注释以帮助手动转换未迁移的代码，并寻找潜在的代码改进。

4、使用 Intel oneAPI DPC++/C++编译器构建项目

确保新迁移的项目使用 Intel oneAPI DPC++/C++编译器成功编译。

INOVANCE

光度立体算子异构部署

1、CUDA与DPC++的基本对应关系：

CUDA to SYCL dictionary

CUDA	SYCL
Block	Work group
Thread	Work item
Grid	ND-range
Kernel	Command group
CUDA Stream	Queue
Shared memory	Local memory
Cooperative groups	Subgroups
Unified memory	Unified shared memory(USM)
Graphs	tf::syclflow in Taskflow

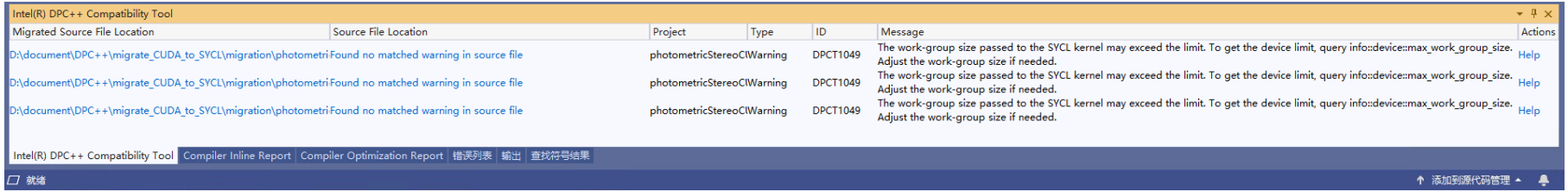
2、部分CUDA API目前仍然不支持，可通过以下链接查阅
https://www.intel.com/content/www/us/en/docs/dpcpp-compatibility-tool/developer-guide-reference/2023-1/cuda-api-migration-support-status.html

3、对于基于cuda封装的库，如OpenCV中gpuMat 数据结构，手动做数据类型转换。

INOVANCE

光度立体算子异构部署

4、



以上图的Warning为例，应核对work-group size是否超过限制。
如果是NVIDIA独显到Intel核显的转换，直接转换的work-group size大概率会超出限制。
对应的警告和错误ID和通过https://www.intel.com/content/www/us/en/docs/dpcpp-compatibility-tool/developer-guide-reference/2023-1/diagnostics-reference.html查询具体细节。

INOVANCE

形状匹配算子优化

General	
Windows SDK Version	**10.0 (最新安装的版本)**
Output Directory	$(SolutionDir)$(Platform)\$(Configuration)\
Intermediate Directory	$(Platform)\$(Configuration)\
Target Name	$(ProjectName)
Platform Toolset	**Intel C++ Compiler 2023**
C++ Language Standard	Default
C Language Standard	Default
Configuration Type	**Dynamic Library (.dll)**
Intel Specific	
Base Platform Toolset	v142
Profile-Guided Build Options	Disabled
Code Coverage Build Options	Disabled
Profile Directory	$(IntDir)
Project Defaults	
Whole Program Optimization	

切换至intel C++编译器，使用#pragma omp simd指令及针对支持AVX2指令集的intel Core CPU完成形状匹配算子效率优化。

```
#pragma omp simd reduction(+:score)
```

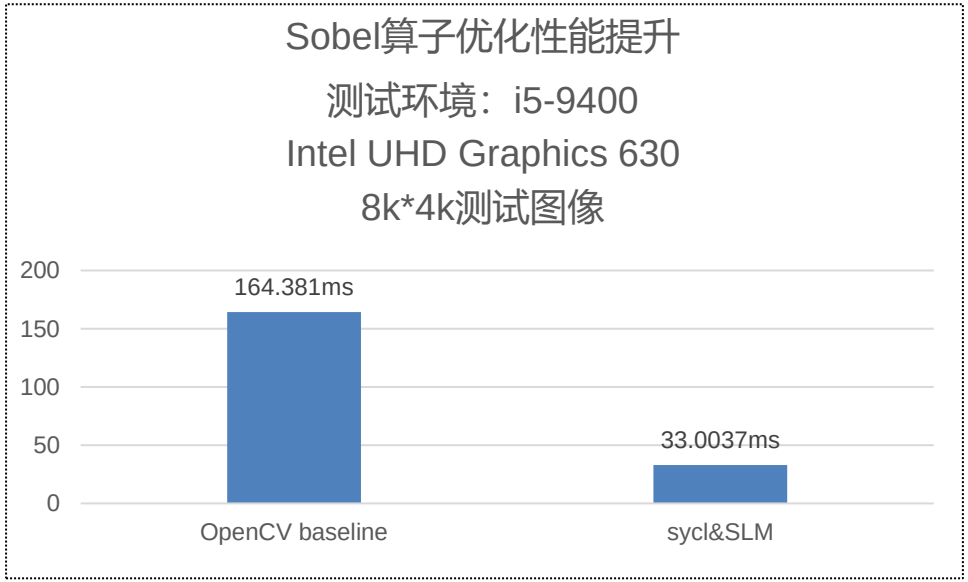
Intel Processor-Specific Optimization: **Intel(R) Advanced Vector Extensions 2 (Intel(R) AVX2) (/QxCORE-AVX2)**

- Intel(R) processor or microarchitecture code name Rocket Lake (/Qxrocketlake)
- Intel(R) processor or microarchitecture code name Sapphire Rapids (/Qxsapphirerapids)
- Intel(R) processor or microarchitecture code name Skylake (/Qxskylake)
- Intel(R) processor or microarchitecture code name Tiger Lake (/Qxtigerlake)
- Intel(R) processor or microarchitecture code name Tremont (/Qxtremont)
- Intel(R) processor or microarchitecture code name Whiskey Lake (/Qxwhiskeylake)
- Intel(R) Advanced Vector Extensions 512 (Intel(R) AVX-512) common for Intel(R) Xeon(R) and Intel(R) Xeon Phi(TM) processors (/QxCOMMON-AVX512)
- Intel(R) Advanced Vector Extensions 512 (Intel(R) AVX-512) for Intel(R) Xeon Phi(TM) processors (/QxMIC-AVX512)
- Intel(R) Advanced Vector Extensions 512 (Intel(R) AVX-512) for Intel(R) Xeon(R) processors (/QxCORE-AVX512)
- Intel(R) Advanced Vector Extensions 2 (Intel(R) AVX2) (/QxCORE-AVX2)

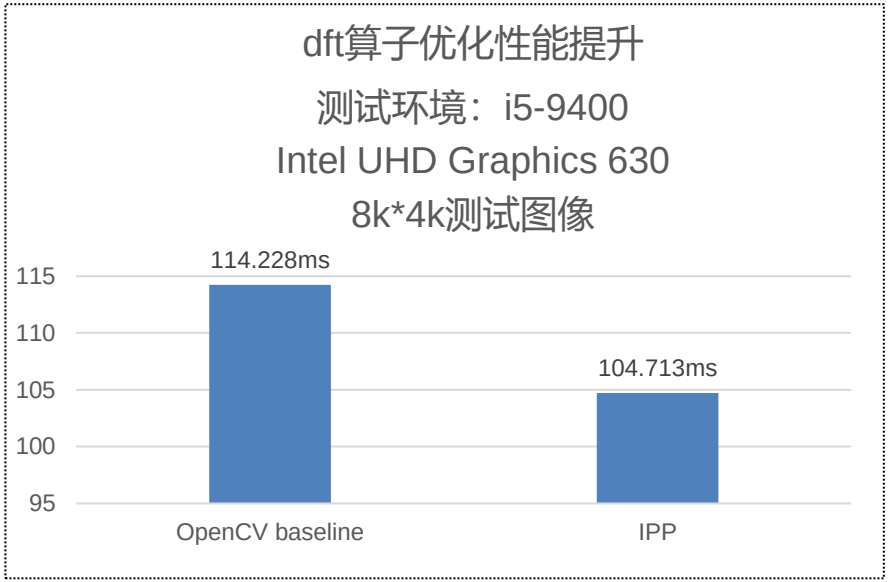
INOVANCE

INOVANCE

性能提升



提升4.98倍



提升1.09倍

INOVANCE

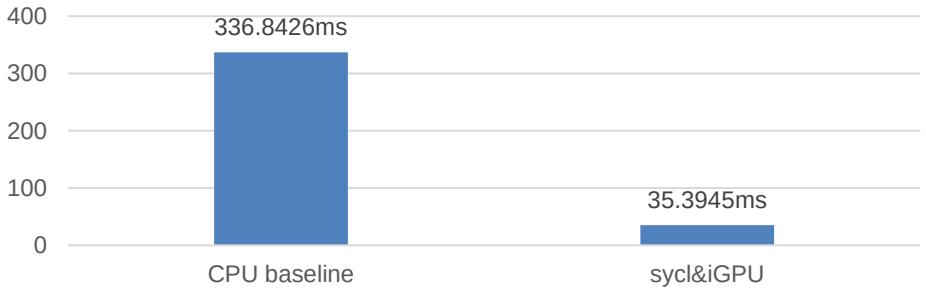
性能提升

光度立体CUDA转DPC++快速部署

测试环境：i5-9400

Intel UHD Graphics 630

3072*2048测试图像

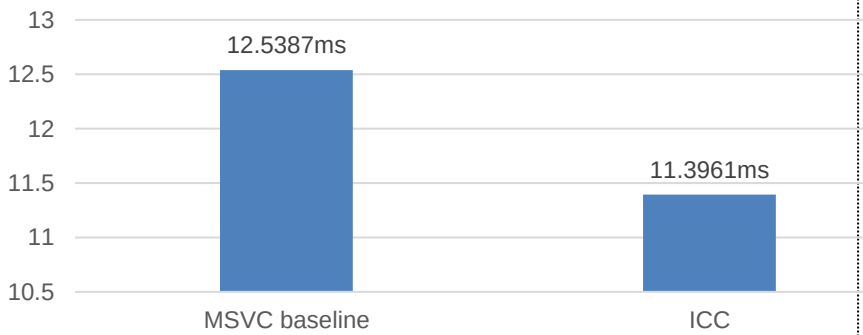


快速部署并获取与硬件算力匹配的效率

形状匹配算子优化

测试环境：i5-9400

1280*1024测试图像



提升1.10倍

INOVANCE

Forward Always Progressing