【报修开发】缝针缝线市一南院 - Task #19214

opencv GPU加速调研

2024-11-22 09:16 - 尹駉（Antoine）

状态:	Feedback	**开始日期:**	2024-11-25
优先级:	Normal	**计划完成日期:**	2024-11-29
指派给:	蔡旭庆（Wendell）	**% 完成:**	0%
类别:		**预期时间:**	0.00 小时
目标版本:		**耗时:**	45.00 小时
客户名称:		**现场FSW统计:**	是
关键字:		**产品名称:**	缝针缝线柜

描述

历史记录

#1 - 2024-11-22 09:21 - 尹駉（Antoine）

我们安卓使用的3588应该支持OpenCL。看看是否能启用OpenCV的GPU加速。要研究加速的效果是否明显。目前可针对native.cpp中，Java_com_sunray_ocr_keyword_OCRPredictorNative_StartSingleModelTest中的“ SURF算法 ”对比启用GPU和不启用的速度差异。大致代码段可搜索如下：

```
cv::Ptr&lt;cv::xfeatures2d::SURF&gt;  detector;
                    detector  =  cv::xfeatures2d::SURF::create(tagInfo.ModelHessian);
                    std::vector&lt;cv::KeyPoint&gt;  kp;
                    detector->detect(mtTemplateGray,  kp,  mtMask);

cv::Ptr&lt;cv::xfeatures2d::SURF&gt;  extractor  =  cv::xfeatures2d::SURF::create();
                    cv::Mat  mtDesriptorData;
                    extractor->compute(mtTemplateGray,  kp,  mtDesriptorData);

m_tagOCR->m_parameters.TagInfoList[nModelOrder].mtModelImage  =  mtModelImage;
                    m_tagOCR->m_parameters.TagInfoList[nModelOrder].kpTemplate  =  kp;
                    m_tagOCR->m_parameters.TagInfoList[nModelOrder].mtTemplateDescriptor  =  mtDesriptorData;
```

以下是文心给出的大概做法：
在Android上使用OpenCV的GPU加速可以通过利用OpenCV提供的GPU模块（opencv_core、opencv_imgproc、opencv_features2d等的GPU版本）来实现。这些模块利用了底层硬件加速，如CUDA（在支持CUDA的Android设备上）或OpenCL，来显著提高图像处理的速度。不过，需要注意的是，并非所有Android设备都支持GPU加速，且具体的加速效果也会因设备而异。

以下是在Android上使用OpenCV GPU加速的一般步骤：

| 安装OpenCV SDK| ：

首先，从OpenCV官方网站下载适用于Android的OpenCV SDK。
将下载的SDK解压到你的开发环境中，并配置好Android Studio以使用它。

| 配置Gradle| ：

在你的Android项目的build.gradle文件中，添加对OpenCV库的依赖。如果你使用的是本地OpenCV SDK，你可能需要将其添加为本地模块。
确保你的项目配置正确，以便能够编译和链接OpenCV库。

| 初始化OpenCV| ：

在你的应用程序启动时，初始化OpenCV库。这通常是通过调用OpenCVLoader.initDebug()或OpenCVLoader.initAsync()来完成的。

| 使用GPU模块| ：

在你的代码中，导入OpenCV的GPU模块。
例如，使用org.opencv.core.CudaMat代替org.opencv.core.Mat来创建矩阵，并使用GPU加速的函数进行处理。
注意，并非所有OpenCV函数都有GPU加速版本。你需要查阅OpenCV文档来了解哪些函数支持GPU加速。

| 处理图像| ：

将图像加载到CudaMat对象中。
使用GPU加速的函数对图像进行处理。
如果需要将处理后的图像显示或保存到设备上，你可能需要将其转换回常规的Mat对象。

| 优化和测试 ：

在不同的设备和不同的图像上测试你的应用程序，以确保GPU加速带来的性能提升是显著的。
根据需要调整你的代码和算法，以最大化利用GPU加速。

以下是一个简单的示例代码片段，展示了如何在Android上使用OpenCV的GPU加速来处理图像：

java
Copy Code
```
import org.opencv.android.OpenCVLoader;
import org.opencv.core.CvType;
import org.opencv.core.Mat;
import org.opencv.core.CudaMat;
import org.opencv.core.Scalar;
import org.opencv.imgproc.Imgproc;

// 在应用程序启动时初始化OpenCV
if (!OpenCVLoader.initDebug()) {
// 处理初始化失败的情况
}

// 创建一个CudaMat对象（注意：这里假设图像大小是已知的）
CudaMat cudaMat = new CudaMat(imageHeight, imageWidth, CvType.CV_8UC3);

// 将图像数据加载到CudaMat对象中（这里需要你自己将图像数据转换为适合CUDA处理的格式）
// ...

// 使用GPU加速的函数对图像进行处理（例如，将图像转换为灰度图）
Imgproc.cvtColor(cudaMat, cudaMat, Imgproc.COLOR_BGR2GRAY);

// 如果需要将处理后的图像显示或保存，你需要将其转换回常规的Mat对象
Mat mat = new Mat();
cudaMat.copyTo(mat);

// 现在你可以使用mat对象进行后续操作，如显示或保存图像
// ...
```

请注意，上述代码仅是一个示例，并没有展示如何实际加载图像数据到CudaMat对象中或如何处理异步操作。在实际应用中，你可能需要使用OpenCV提供的实用函数来加载图像数据，并处理GPU和CPU之间的数据传输。此外，由于GPU资源的有限性，你可能需要合理地管理GPU内存的使用，以避免内存泄漏或性能下降。

#2 - 2024-11-26 09:10 - 蔡旭庆（Wendell）

- 状态 从 New 变更为 In Progress

#3 - 2024-11-26 09:12 - 蔡旭庆（Wendell）

11/25 8h 特征码提取GPU加速测试

#4 - 2024-11-27 09:31 - 蔡旭庆（Wendell）

11/26 8h
orb特征点寻找代码逻辑整理与c++接收流数据代码异常调整与库文件提供，寻找特征值方法分布耗时，无法GPU加速，findNext目前不支持GPU加速，第一次失败后重试图片未进行二值化处理，处理后效率提示

#5 - 2024-11-28 09:27 - 蔡旭庆（Wendell）

11/27 7.5h
调整代码测试与批量测试工具测试回归，重试使用二值化图片也会出现多次findNext情况，单品规回归，未识别由10减到8，默认寻找参数定位失败3.86%，调整后定位失败2.25%

#6 - 2024-11-29 09:17 - 尹駉（Antoine）

- 计划完成日期 从 2024-11-27 变更为 2024-11-29

#7 - 2024-11-29 09:39 - 蔡旭庆（Wendell）

11/28 7.5h stitch拼接gpu加速测试与详细实现步骤整理

#8 - 2024-12-02 09:31 - 蔡旭庆（Wendell）

11/29 7.5h ORB特征匹配与SURF方法完整对比，原图缩小一半后，从图片到切到匹配区域耗时，调整全局GPU加速开关，速度无变化，
ORB约160ms,SURF约350ms, 各标签特征点阈值不同，慕斯最低约30，ORB原图不缩小时，阈值约100，耗时300ms

#9 - 2024-12-02 09:32 - 蔡旭庆（Wendell）

11/29 7.5h ORB特征匹配与SURF方法完整对比，原图缩小一半后，从图片到切到匹配区域耗时，调整全局GPU加速开关，速度无变化，
ORB约160ms,SURF约350ms, 各标签特征点阈值不同，慕斯最低约30，ORB原图不缩小时，阈值约100，耗时300ms

#10 - 2024-12-03 09:24 - 蔡旭庆（Wendell）

12/02 6.5h
stitch分步执行测试，最终拼接图效果不理想，可使用其他图片单独执行最后拼接步骤，耗时约700ms,应可进一步优化。特征相似图片无法拼接（校正板），进一步拼接应使用自然图片进行stitch拼接

#11 - 2024-12-03 09:27 - 尹駧（Antoine）

- 状态 从 In Progress 变更为 Feedback