

航拍拼接模块接口说明

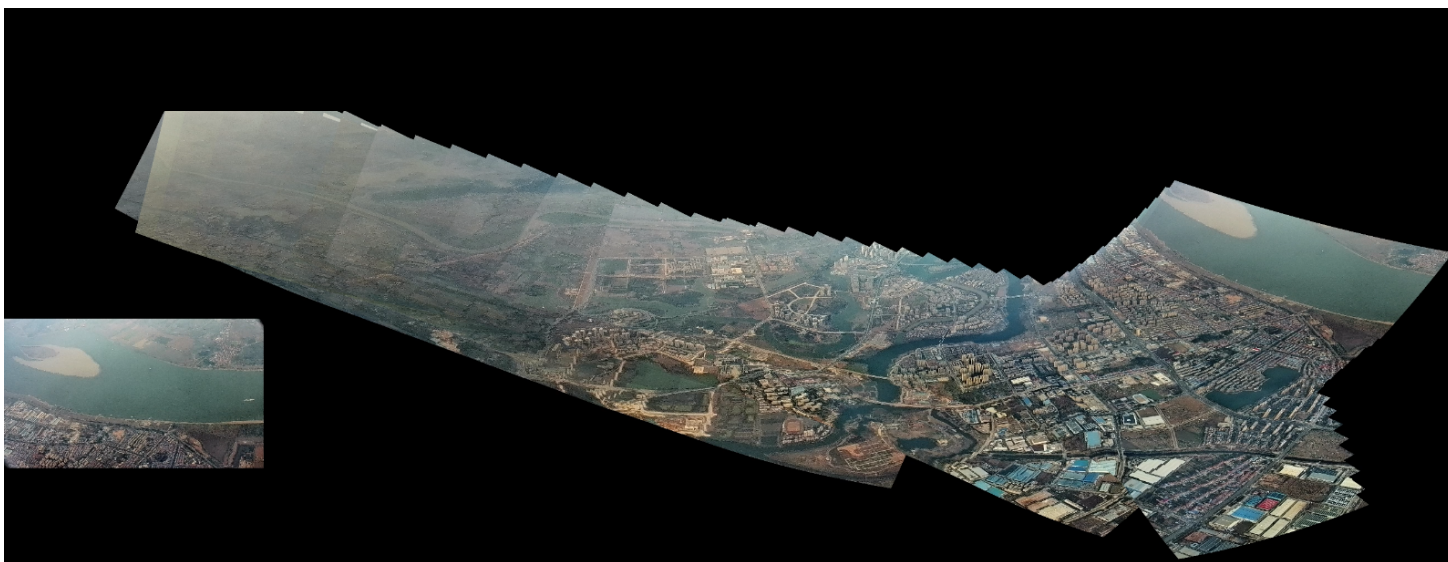
本模块用于吊舱的扫描全景图拼接，包含前视、下视以及特征拼接三个模式。前视与下视投影面和效果均不同。二者接口大部分相同。

特征拼接是完全基于特征的图像算法，本身存在较大的累计误差，仅用于测试和对比出图，暂时没有应用价值。

下视拼接是最广泛使用的拼接方式，可以随着载体飞行进行连续长时拼接，并生成含有地理信息



前视拼接的效果接近360°全景。



1. 依赖

本模块依赖ceres优化库(包含gflag、glog)以及cuda支持, opencv 455支持。
除cuda库外, 其余库均以vc14动态库文件形式提供。

使用cuda将极大加速前视极坐标拼接模块。

2. 下视接口

```
static API_UnderStitch* Create(std::string cachedir = "./cache");
```

模块创建接口, 其中默认参数为使用cache的路径, cache将缓存需要处理的帧原始数据。

```
static void Destroy(API_UnderStitch* obj);
```

模块析构接口, 使用完毕后释放内存。

```
virtual UPanInfo Init(FrameInfo info) = 0;
```

初始化接口, 请使用拼接时刻的内外参信息进行初始化。初始化完成后返回全景图的参数

```
virtual void SetOutput(std::string name, std::string outdir) = 0;
```

设置输出路径, 下视模块按照通用谷歌瓦片生成标准256*256瓦片, 提前设置好输出文件夹。

```
virtual void SetConfig(UPanConfig config) = 0;
```

设置运行参数。

```
virtual BYTE8 Run(GD_VIDEO_FRAME_S img, FrameInfo para) = 0;
```

运行拼接流程，在该过程中程序自动将当前图像重新投影到全景图上，可以通过访问全景图内存进行实时显示。

```
virtual GD_VIDEO_FRAME_S ExportPanAddr() = 0;
```

获取全景图接口，其中GD_VIDEO_FRAME_S是高德研发中心通用帧接口，内部维护了图像类型和指针等信息。具体参见GD_VIDEO_FRAME_S接口说明

```
virtual SINT32 OptAndOutCurrPan() = 0;
```

全景投影完毕后，可以调用该接口进行优化，主要是精细对齐帧以及拼接缝消除等，暂未实现。

前视接口

```
static API_FrontStitch* Create(std::string cachedir = "./cache");
```

模块创建接口，其中默认参数为使用cache路径，前视模块不使用cache。

```
static void Destroy(API_FrontStitch* obj);
```

模块析构接口，使用完毕后释放内存。

```
virtual FPanInfo Init(FrameInfo info, ScanRange stAzRange, ScanRange stPtRange) :
```

初始化接口，使用拼接时刻的内外参信息进行初始化，

```
virtual BYTE8 Run(GD_VIDEO_FRAME_S img, FrameInfo para) = 0;
```

运行拼接流程，在该过程中程序自动将当前图像重新投影到全景图上，可以通过访问全景图内存进行实时显示。

```
virtual GD_VIDEO_FRAME_S ExportPanAddr() = 0;
```

获取全景图接口，其中GD_VIDEO_FRAME_S是高德研发中心通用帧接口，内部维护了图像类型和指针等信息。具体参见GD_VIDEO_FRAME_S接口说明

```
virtual SINT32 OptAndOutCurrPan() = 0;
```

全景投影完毕后，可以调用该接口进行优化，主要是精细对齐帧以及拼接缝消除等，暂未实现。

附 下视拼接的标准地理产品

下视拼接过程目前是功能最为完善的，主要分为粗拼接+优化两个过程。

粗拼接 是指利用外部参数信息直接利用无源定位技术对帧进行地理纠正，纠正后的数据天然具备地理属性，只需要简单的仿射变换即可显示为空间一致的底图。

优化 是指利用帧间同名点匹配关系建立误差方程，迭代优化映射关系的过程。

粗拼接是整个工作流的核心，形成任意场景均有高可用性的拼接实现。而优化过程是可以关闭的，用于对拼接质量要求较高的场景或者外参误差较大的设备。

如API说明所示，SDK提供了全景图的显示接口。**但我们不建议使用该全景图**，主要原因是全景显示策略与逻辑是为了便于展示算法效果和内部切瓦片使用的。对图层叠加以及漫游、回溯等高级功能难以支持。

相反的，在下视算法执行期间，还生成了两种标准地理数据，具有通用的行业接口。能够轻松支持各类地理web、桌面端应用。

1. 单帧纠正产品

帧输入后，算法利用外参信息对图像进行地理纠正，简单说就是正北+下视校正。且给出帧的kml信息（谷歌地球标准数据层）。

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<GroundOverlay>
  <name>DJI6830</name>
  <Icon>
    <href>DJI6830.png</href>
  </Icon>
  <LatLonBox>
    <north>30.267059</north>
    <south>30.264926</south>
    <east>114.446180</east>
    <west>114.443523</west>
  </LatLonBox>
</GroundOverlay>
</Document>
</kml>
```

2. 谷歌瓦片

视频输入过程中，算法根据外参变化情况自行判断执行优化时机，优化完成后会将拼接好的信息切成标准谷歌瓦片。

谷歌瓦片按照{Z}-{X}-{Y}.png形式进行命名，能够被gis前端正确显示。下图是cesiumjs的可视化效果。**因此整个执行流程可以概括为实时显示基于kml的帧，并周期性的或者触发式的读取瓦片覆盖显示。**

