

汉振3D相机HTTP接口文档

目的

本文档为汉振3D相机，HTTP接口对接方式的描述。

使用HTTP接口，相比传统SDK开发，可以有以下好处：

- 允许对接方使用任何操作系统、开发语言、开发环境和系统架构
- 较为灵活和简单的对接方式，减少了很多繁琐的流程和依赖
- HTTP请求相关的库，已经在几乎所有语言或常用开发库中内置，无需集成汉振的SDK即可使用
- 允许多个主机，多个客户端共用一台相机
- 允许在局域网甚至广域网连接使用
- SDK和业务程序彻底剥离，不需要集成任何动态库，不用担心版本匹配，SDK崩溃，log混杂等问题

使用对象：

汉振相机二次开发用户

接口列表：

- 连通测试 - GET ping
- 获取相机信息 - GET cameraInfo
- 采集2D - POST capture2D
- 采集3D - POST capture3D
- 采集AMP图和3D - POST captureAmpAnd3D
- 采集2D和3D - POST capture2DAnd3D
- 采集深度图 - POST captureDepth
- 采集深度图和3D - POST captureDepthAnd3D
- 采集彩色2D - POST captureRGB2D
- 采集彩色2D和3D - POST captureRGB2DAnd3D

Demo

基本上所有开发框架都有HTTP请求相关库，开发者可以查阅相关文档直接对接HTTP进行开发。

同时为了方便对接，我们准备了3个常用环境下HTTP的使用Demo，可以参考使用。

- C++和Qt

可以参考Demo中的 [HCCameraQtHttpDemo](#) 进行二次开发

此Demo可以运行在标准Qt环境中，方便嵌入到客户自有Qt程序中。

Demo同时支持Qt5和Qt4，需要注意的是Qt4中没有json相关库，因此若使用Qt4，需要带上 [qjson4](#)到你的工程中，Qt5则不需要

- C++和Win32

可以参考Demo中的 [HCCameraWinHttpDemo](#) 进行二次开发

此Demo使用C++配合Win32，可以运行在Windows系统中

- C++和curl

可以参考Demo中的 [HCCameraCurlHttpDemo](#) 进行二次开发

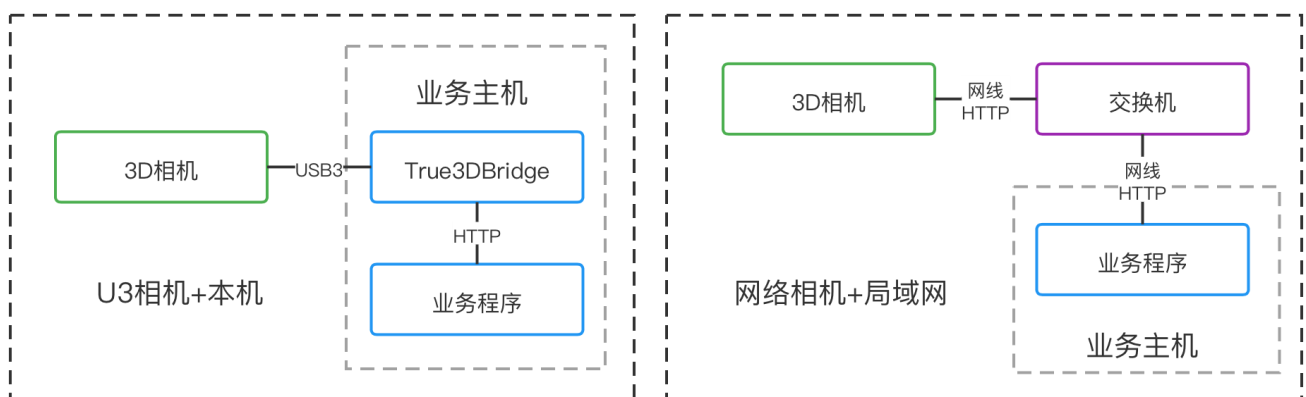
此Demo使用C++配合curl，可以运行在标准C++和curl环境中，Windows下可以用MinGW配合curl，Linux和macOS下自带curl

- C#和halcon

可以参考Demo中的 [HCCameraCSharpHttpDemo](#) 进行二次开发

软硬件架构

基于USB的相机和基于网络的相机都可以使用HTTP接口，分别对应以下2种组合：



- 业务程序为客户自有程序，例如检测程序，抓取程序等。
- USB相机需要常开Bridge程序在后台，网络相机则不用。
- 交换机作为可选项，可以使用也可以不使用。

常见问题

- **True3DBridge**程序基于3D相机C++ SDK开发，本身只做了在C++ SDK上的HTTP接口封装，并作为单独进程运行。
- 延迟问题：一般来说在本机（127.0.0.1）环境，使用230W像素相机时。相比直接使用C++ SDK对接，HTTP封装会给每次采集增加大约50~150ms的延迟，千兆网环境会增加大约50~300ms的延迟，2.5G网环境会增加大约50~200ms延迟，具体取决于传输速度、数据大小和是否开启点云压缩（压缩默认关闭）。
- 接口速率：汉振所有网络相机，均默认配置2.5G有线网卡。同时向下兼容千兆、百兆网络环境。追求极致性能的话，客户主机也可以配置2.5G网卡以达到最快传输速率。
- 拍摄顺序：**True3DBridge**程序开启后，多个客户端可以同时访问，相机资源非独占。但是在具体拍摄层面，顺序根据是HTTP请求，先到先拍摄。若在拍摄过程中新收到的HTTP请求，会等待之前拍摄完毕再处理。
- 连接方式：推荐在相机和主机之间增加一个交换机，方便扩展和调试。若直连也是可以的，在性能上几乎无区别。
- 防止程序卡死：在Windows操作系统中，请不要用鼠标点击CMD窗口内部。点击CMD窗口内部会导致控制台程序运行冻结，表现为有一个光标闪烁，必须键盘输入回车才能继续运行。关闭控制台的**快速编辑模式**，也可以防止此问题。

使用流程（网络相机）

- 配置客户主机网卡IP信息：
 - IP: 10.20.1.2 ~ 10.20.2.254

在这个IP段内任意选择一个IP即可

基本上所有操作系统都可以对单网卡设置多IP，也就是说在已经设置了其他固定IP的情况下，还可以再针对相机额外设置IP，而且互不影响。

 - 掩码: 255.255.224.0
 - 网关: 不用设置，留空
 - DNS: 不用设置，留空
- 参考相机基本信息贴纸，查看相机IP信息。
- 参考《Product Manual-True Series 3D Camera》手册，在 **True3DViewer** 程序中跑通完整流程，可以正常采集点云。
- 根据需求、工件特性、使用环境，在 **True3DViewer** 程序中调整好采集参数，可以采集合适的点云数据。
- 在 **True3DViewer** 程序右侧的参数配置面板，点击 **导出** 按钮，导出采集参数到配置文件中。
- 保存好导出的配置文件，后面使用HTTP接口时会用到这个配置文件。

使用流程（U3相机）

- 参考《Product Manual-True Series 3D Camera》手册，在 **True3DViewer** 程序中跑通完整流程，可以正常采集点云。
- 根据需求、工件特性、使用环境，在 **True3DViewer** 程序中调整好采集参数，可以采集合适的点云数据。
- 在 **True3DViewer** 程序右侧的参数配置面板，点击 **导出** 按钮，导出采集参数到配置文件中。
- 保存好导出的配置文件，后面使用HTTP接口时会用到这个配置文件。
- 关闭 **True3DViewer** 程序，开启 **True3DBridge** 程序

注意，因为相机通讯是独占的，因此同一时刻只能开启一个程序。若同一时刻打开两个程序，第二个程序将无法正常和相机通讯，并提示报错。

- 查看控制台提示
 - **Camera name: TrueD2305M-d0610001**
表示正常识别到相机，准备初始化
 - **True3DBridge initialize succeed**
表示代理程序初始化完成
- 若没有其他报错，表示 **True3DBridge** 程序已经正常初始化，可以使用了。

- 简单测试

打开浏览器，url栏输入 **http://127.0.0.1:13890/cameraInfo**，即可获得相机基本信息

若用户有curl环境，可以使用以下命令进行capture测试，这里假设之前保存的配置文件名字为 **config.json**

```
curl http://127.0.0.1:13890/capture -X POST -d @config.json --output received
```

- 手动指定相机和监听端口

若需要在1台电脑同时连接多台相机，需要手动指定HTTP监听端口和相机

True3DBridge 程序默认监听13890端口，会选择相机列表里的第一个可用相机用于初始化

通过命令行参数可以手动指定监听端口和相机，语法如下：

```
True3DBridge.exe port=18888 camera=TrueD2305M_d0610001
```

接口定义

- 方式：HTTP
- Method：GET 或 POST
- 数据类型：JSON
- 编码：UTF-8
- 通用返回示例：

参数名称	类型	描述
isSucceed	bool	接口调用是否成功
message	string	信息，接口调用成功时一般为空
data	object 或 array	若接口有返回信息，则会包装在data内返回

注：二进制数据返回的接口根据接口具体格式定义，不为JSON

连通测试

- 请求地址

<http://127.0.0.1:13890/ping>

注：127.0.0.1:13890 地址和端口可以根据实际需求更换

- 请求方法

GET

- 请求参数

参数名称	类型	必填项	示例	描述
------	----	-----	----	----

暂无

- data 返回参数

参数名称	类型	示例	描述
serverTime	number	1600000000000	服务端时间信息，毫秒级时间戳

获取相机信息

- 请求地址

<http://127.0.0.1:13890/cameraInfo>

注：127.0.0.1:13890 地址和端口可以根据实际需求更换

- 请求方法

GET

- 请求参数

参数名称	类型	必填项	示例	描述
暂无				

- data 返回参数

参数名称	类型	示例	描述
available	bool	true	相机是否可用
serialNumber	string	"TrueD2305M-d0600000"	产品序列号
configureFilePath	string	"C:/xxxx.xml"	相机配置文件存储路径
imageSize -> width	number	1920	相机图像水平像素数量，也为有序点云水平数量
imageSize -> height	number	1200	相机图像垂直像素数量，也为有序点云垂直数量

采集

调用相机采集功能，等待采集完成后再返回数据。

请求参数可以在 [True3DViewer](#) 程序中直接导出，也可以通过二次开发自定义组合，格式为标准的JSON

- 请求地址
 - 采集2D: <http://127.0.0.1:13890/capture2D>
 - 采集3D: <http://127.0.0.1:13890/capture3D>
 - 采集AMP图和3D: <http://127.0.0.1:13890/captureAmpAnd3D>
 - 采集2D和3D: <http://127.0.0.1:13890/capture2DAnd3D>
 - 采集深度图: <http://127.0.0.1:13890/captureDepth>
 - 采集深度图和3D: <http://127.0.0.1:13890/captureDepthAnd3D>
 - 采集彩色2D: <http://127.0.0.1:13890/captureRGB2D>
 - 采集彩色2D和3D: <http://127.0.0.1:13890/captureRGB2DAnd3D>

注: [127.0.0.1:13890](#) 地址和端口可以根据实际需求更换

- 请求方法

POST

- 请求参数

参数名称	类型	必填项	示例	描述
hdr2DSetting	array	是	[object,...]	2D采集参数
hdr3DSetting	array	是	[object,...]	3D采集参数
pointCloudRotationSetting	object	否	{...}	点云旋转后处理参数
pointCloudCentroidClipSetting	object	否	{...}	质心包围后处理参数
pointCloudRoiClipSetting	object	否	{...}	ROI过滤后处理参数
pointCloudFilterSetting	object	否	{...}	滤波后处理参数
pointCloudFixSetting	object	否	{...}	补洞后处理参数
pointCloudAxleRotationSetting	object	否	{...}	倾斜矫正后处理参数

注：后处理参数详细解释，请参考《Product Manual-True Series 3D Camera》手册中的第4章。

注：一般来说不需要二次开发来修改这些参数，所有参数都可以在 True3DViewer 中配置好，并导出到json配置文件中。

- 请求参数示例

```

{
  "capture2DEnabled": true,
  "capture3DEnabled": true,
  "hdr2DSetting": [
    {
      "brightness": 10,
      "exposureTime": 8000,
      "gain": 0,
      "ledColor": "WHITE",
      "lightSource": "PROJECTOR"
    }
  ],
  "hdr3DSetting": [
    {
      "brightness": 10,
      "exposureTime": 8000,
      "frameRateControl": "WHOLE",
      "gain": 0,
      "grayLevel": 0,
      "ledColor": "WHITE"
    }
  ],
  "pointCloudAxleRotationSetting": {
    "autoUpdateOnScan": true,
    "enabled": false,
    "xAxleRotationValue": 0,
    "yAxleRotationValue": 0
  }
}
    
```

```

    },
    "pointCloudCentroidClipSetting": {
      "enabled": true,
      "value": 6
    },
    "pointCloudFilterSetting": {
      "enabled": true,
      "value": 30,
      "xyGaussianEnabled": true,
      "zGaussianEnabled": true,
      "zGaussianValue": 2,
      "zMedianEnabled": true
    },
    "pointCloudFixSetting": {
      "enabled": true,
      "iterationCount": 0,
      "value": 30
    },
    "pointCloudRoiClipSetting": {
      "enabled": false,
      "maxX": 100,
      "maxY": 100,
      "maxZ": 250,
      "minX": -100,
      "minY": -100,
      "minZ": 50
    },
    "pointCloudRotationSetting": {
      "enabled": true
    }
  }
}

```

- 数据说明

- 点云数据可能会包括nan点，nan点时xyz的3通道均为nan
- 3D点云数据和物理尺寸对应，单位为毫米。即12.3数值表示12.3毫米。
- 2D数据和3D数据已经是行列对齐的，可以根据2D图像做定位，然后到3D做深度信息提取。
- 图像行对齐（step）为宽度相关，不填充空白数据

- 返回参数

默认返回数据（内存块数据），根据2D和3D不同采集模式组合，有多种正常结果和1种错误结果。

可以通过HTTP返回header中的Content-Type判断具体数据类型

判断采集是否成功有多种方法，可以根据自己实际需求选择其一

- 判断请求是否超时，一般来说在局域网或者直连环境，即使多组HDR，5秒内也可以完成采集。若超时则说明相机工作不正常或者参数错误。

- 判断返回数据区大小，这个方法最方便，推荐用这个方法。因为在参数相同并采集成功时，每次HTTP接口返回的数据区（body）大小是完全一致的。
- 判断HTTP返回码，200表示成功。
- 判断Content-Type，若和下方列表中不匹配，则可认定为采集失败。
- 不同请求类型返回值说明
 - 采集2D - capture2D
 - Content-Type: `image/grayscale8`
 - 作用：返回2D采集的数据
 - body数据区大小：`width * height`，即1920 * 1200分辨率的相机一共会返回固定 `2304000` 字节数据
 - 数据排序：二进制数据，均为2D图像数据
 - 数据类型：单通道，单字节，无符号整形（uint8），表示灰度值
 - 采集3D - capture3D
 - Content-Type: `pointcloud/xyz`
 - 作用：返回3D采集的数据
 - body数据区大小：`width * height * 3 * 4`，即1920 * 1200分辨率的相机一共会返回固定 `27648000` 字节数据
 - 数据排序：二进制数据，均为3D点云数据
 - 数据类型：3通道，4字节，单精度浮点数（float），XYZ排序
 - 采集AMP图和3D - captureAmpAnd3D
 - Content-Type: `image/grayscale8+pointcloud/xyz`
 - 作用：返回AMP图 + 3D采集的数据
 - body数据区大小：`width * height + width * height * 3 * 4`，即1920 * 1200分辨率的相机一共会返回固定 `29952000` 字节数据
 - 数据排序：二进制数据，前 `width * height` 字节数据为2D图像数据，剩余部分为3D点云数据
 - 数据类型（2D部分）：单通道，单字节，无符号整形（uint8），表示灰度值
 - 数据类型（3D部分）：3通道，4字节，单精度浮点数（float），XYZ排序

注：此接口不会单独开启2D拍摄，在性能上优于 capture2DAnd3D 接口，AMP图为相位计算得出，画质较差，仅可用于对画质无要求的场景
 - 采集2D和3D - capture2DAnd3D

- Content-Type: `image/grayscale8+pointcloud/xyz`
- 作用：返回2D + 3D采集的数据
- body数据区大小： $width * height + width * height * 3 * 4$ ，即1920 * 1200分辨率的相机一共会返回固定 `29952000` 字节数据
- 数据排序：二进制数据，前 $width * height$ 字节数据为2D图像数据，剩余部分为3D点云数据
- 数据类型（2D部分）：单通道，单字节，无符号整形（uint8），表示灰度值
- 数据类型（3D部分）：3通道，4字节，单精度浮点数（float），XYZ排序

注：此接口会单独开启2D拍摄，性能上慢于 `captureAmpAnd3D` 接口，但是2D图质量较好，可以用于需要对画质有要求的场景

○ 采集深度图 - `captureDepth`

- Content-Type: `image/grayscale8`
- 作用：返回采集的深度数据
- body数据区大小： $width * height$ ，即1920 * 1200分辨率的相机一共会返回固定 `2304000` 字节数据
- 数据排序：二进制数据，均为深度数据
- 数据类型：单通道，单字节，无符号整形（uint8），表示深度值

注：此接口返回的是经过处理的深度数据，只有单字节精度（0到255），和实际物理尺寸不对应，可以用于做图像识别，提取XY坐标。若需要和物理尺寸对应的深度数据请直接使用点云数据。

○ 采集深度图和3D - `captureDepthAnd3D`

- Content-Type: `image/grayscale8+pointcloud/xyz`
- 作用：返回采集的深度数据 + 3D采集的数据
- body数据区大小： $width * height + width * height * 3 * 4$ ，即1920 * 1200分辨率的相机一共会返回固定 `29952000` 字节数据
- 数据排序：二进制数据，前 $width * height$ 字节数据为深度数据，剩余部分为3D点云数据
- 数据类型（深度部分）：单通道，单字节，无符号整形（uint8），表示深度值
- 数据类型（3D部分）：3通道，4字节，单精度浮点数（float），XYZ排序

○ 采集彩色2D - `captureRGB2D`

- Content-Type: `image/rgb888`
- 作用：返回2D采集的数据

- body数据区大小: $width * height * 3$, 即1920 * 1200分辨率的相机一共会返回固定 6912000 字节数据
- 数据排序: 二进制数据, 均为2D图像数据
- 数据类型: 三通道RGB排序, 单字节, 无符号整形 (uint8)
- 采集彩色2D和3D - captureRGB2DAnd3D
 - Content-Type: image/rgb888+pointcloud/xyz
 - 作用: 返回2D + 3D采集的数据
 - body数据区大小: $width * height * 3 + width * height * 3 * 4$, 即1920 * 1200分辨率的相机一共会返回固定 34560000 字节数据
 - 数据排序: 二进制数据, 前 $width * height * 3$ 字节数据为2D图像数据, 剩余部分为3D点云数据
 - 数据类型 (2D部分): 三通道RGB排序, 单字节, 无符号整形 (uint8)
 - 数据类型 (3D部分): 3通道, 4字节, 单精度浮点数 (float), XYZ排序
- 发生错误
 - Content-Type: application/json;charset=UTF-8
 - 作用: 若参数组合有误或相机情况异常, 返回具体错误信息
 - 数据类型: 文本数据, 格式为JSON