

## Mat\_on\_Pangolin

感觉 pangolin 的资料真是少之又少，而自己要用的 ORB\_SLAM2 的显示界面又是用到它，所以还是花了些时间读了读他的代码，以下是自己写的将 **opencv** 的 **cv::Mat** 类型图像显示在 **Pangolin** 界面上程序笔记 ([https://github.com/wangzhaodong123/Mat\\_on\\_Pangolin](https://github.com/wangzhaodong123/Mat_on_Pangolin))，如有错误，欢迎指正 ([yuanliudongdong@163.com](mailto:yuanliudongdong@163.com))。

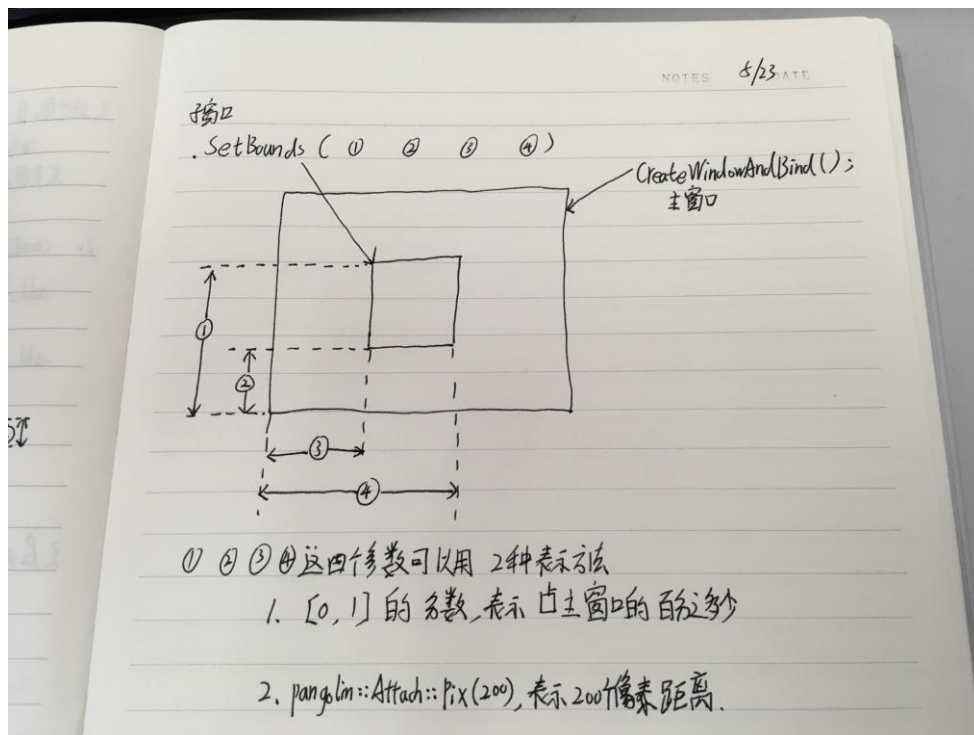
参考博客：Pangolin 安装及其使用

地址：(<https://blog.csdn.net/c602273091/article/details/65441315>)

将 **opencv** 的 **cv::Mat** 类型图像显示在 **Pangolin** 界面上：（主要函数代码）

- //图像显示“窗口”，位置与按钮栏紧邻，大小 640\*480  
pangolin::View& d\_image = pangolin::Display("image")  
.SetBounds(1,0,1,0)  
.SetLock(pangolin::LockRight, pangolin::LockBottom);

**SetBounds()**函数是设置窗口位置的函数，下图是对其参数的简单描述：



- //pangolin 的输入图像数组  
`unsigned char* imageArray = new unsigned char[3*width*height];`  
`pangolin::GltTexture imageTexture(width,height);`  
具体的，**pangolin** 需要一个类似于一维数组的容器来保存图像数据，因为我们要加载的图像是 3 通道的，所以共  $3*width*height$  个元素；
- `void setImageData(unsigned char * imageArray, cv::Mat image){};`  
定义的将 **cv::Mat** 数据存放到 **imageArray** 地址下的函数，其实就是遍历图像的所有元素，并按顺序存放到指定地址下；
- `setImageData(imageArray,image_initial);`

```
imageTexture.Upload(imageArray, GL_RGB, GL_UNSIGNED_BYTE);
```

```
d_image.Activate();
```

```
//设置图片显示的色调好像是
```

```
glColor3f(1.0,1.0,1.0);
```

```
//图像翻转显示函数
```

```
imageTexture.RenderToViewportFlipY();
```

这些都是 **pangolin** 的基本显示函数，需要放在一个 **while** 下，不断循环刷新显示；