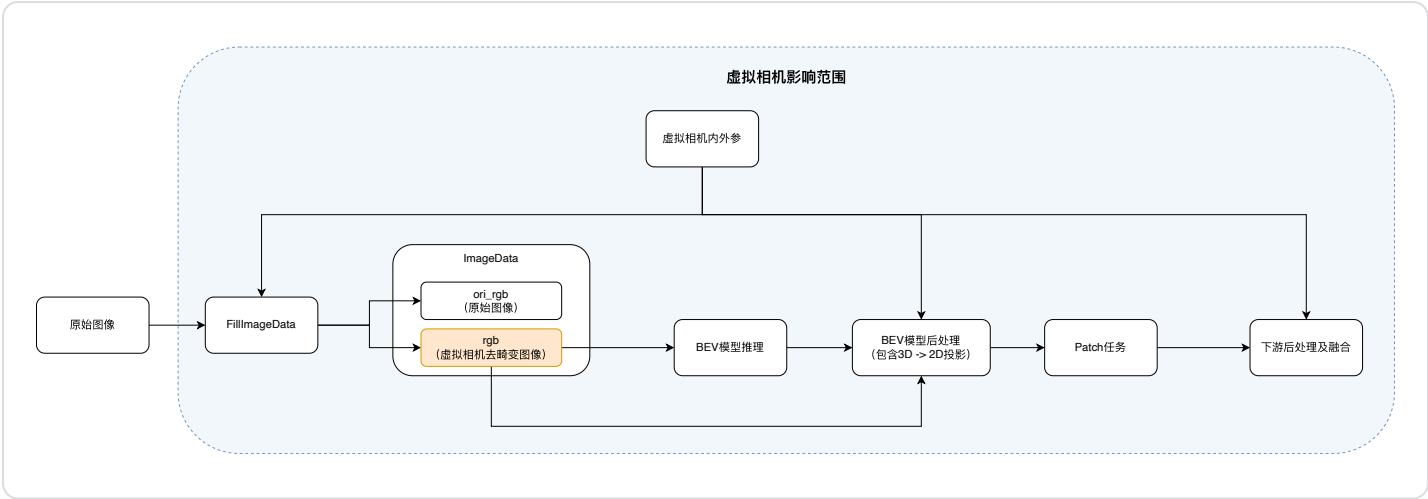


# 障碍物虚拟相机重构

## 2023-04-21 会议纪要

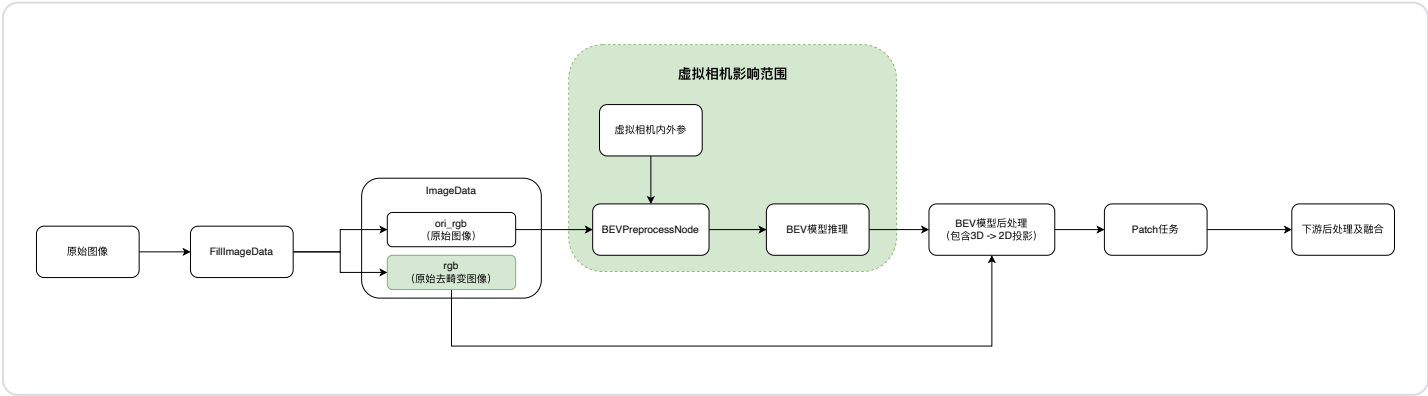
结论：采用原始方案和重构方案结合的折中方案，具体介绍如下：

### 【原始方案】



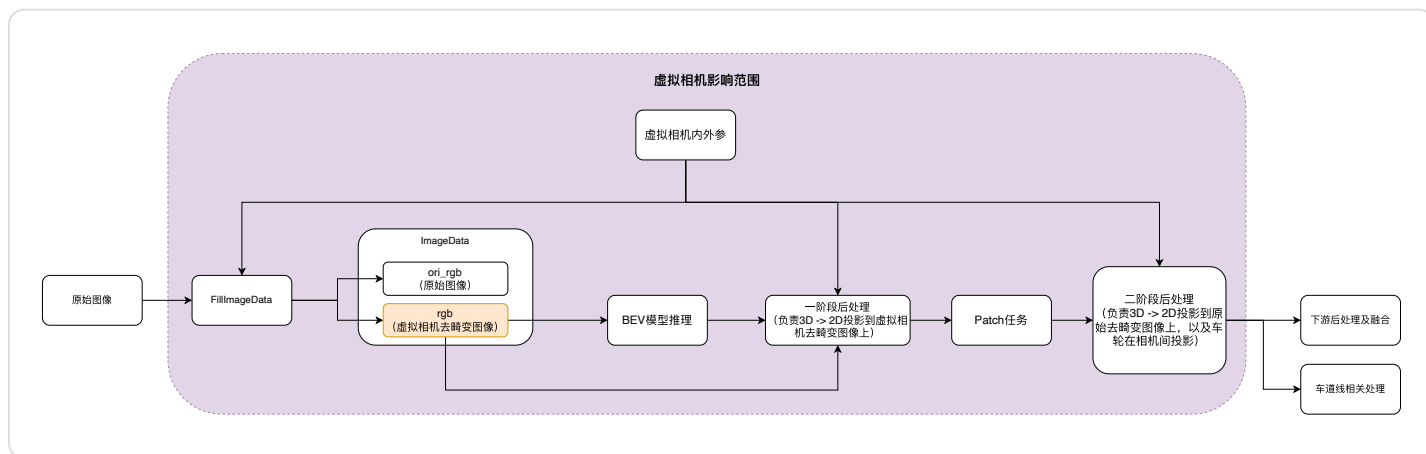
- **好处**：patch任务在虚拟相机去畸变图像上做能利用好虚拟相机调整好的FOV，保证patch任务的结果的正确性和合理性，下游暂时不用做额外的适配；
- **风险**：理解和维护成本高；

### 【重构方案】



- **好处**：没什么理解和维护成本，结构上也更简单和清晰；
- **风险**：patch任务可能会因为FOV的问题不能保证正确性和合理性；

### 【折中方案】

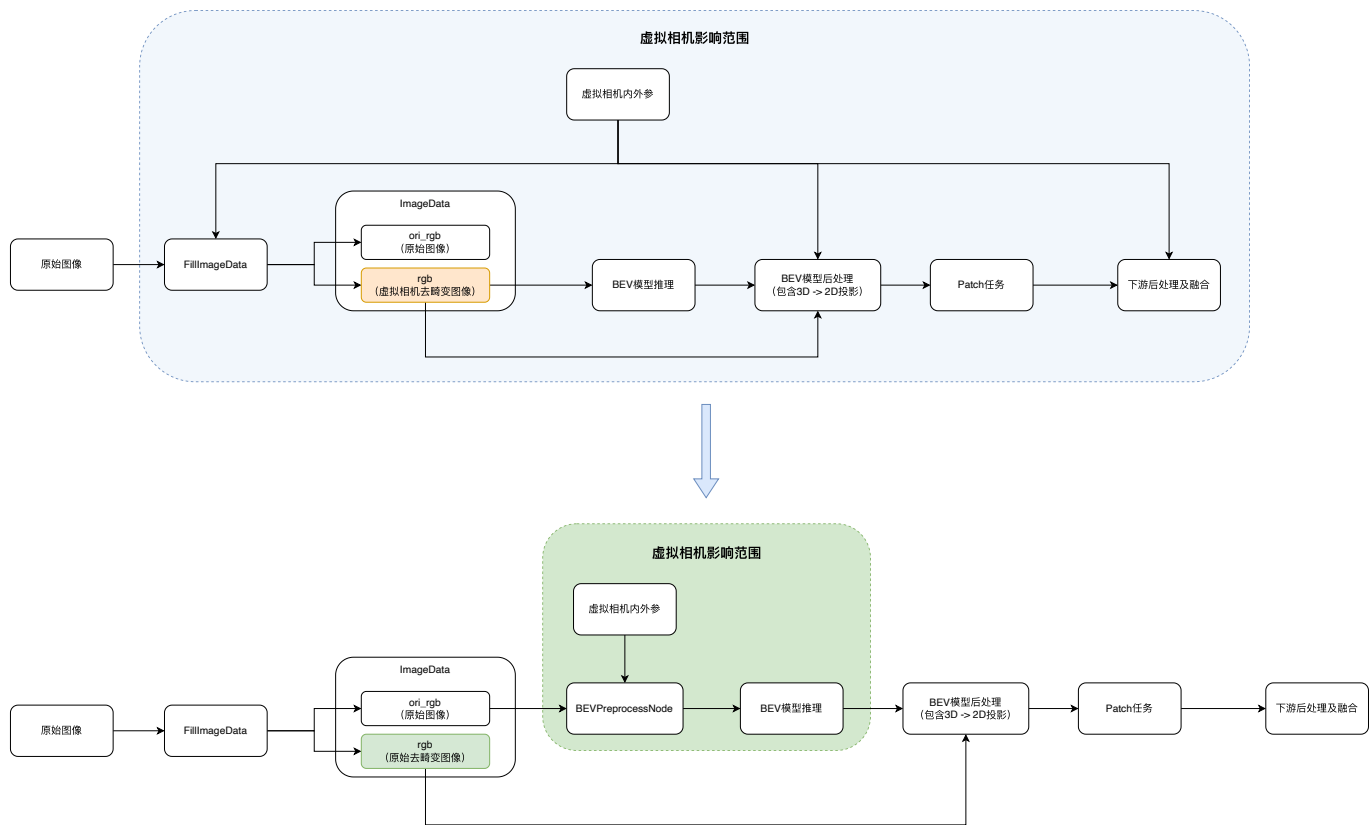


- 还是把虚拟相机的影响范围缩小，不过不会缩小到只影响BEV的推理部分，patch的任务还是在**虚拟相机去畸变的图像**上做，属性相关的信息直接绑定到3D障碍物中，只不过在传给下游之前由专门的模块来生成下游使用的2D信息，包括将**3D障碍物投影成原始去畸变图像上的2D框**以及**车轮从虚拟相机图像投影到原始去畸变图像**，所以下游也不需要考虑障碍物虚拟相机；
- **好处**：理解和维护成本低，且在虚拟相机的FOV下能保证patch任务的正确性和合理性；
- **风险**：由于虚拟相机去畸变图像和原始去畸变图像的FOV不同，车轮从虚拟相机投影到原始去畸变图像**可能会超出图像范围**而被丢弃；

## 背景

- 感知接收到图像之后会保存至CameraFrame中的ImageData中，ImageData中包含ori\_rgb和rgb两个Image8U，都是用来保存图像的；
- 之前虚拟相机放在InferenceEngineComponent中的FillImageData中做的，即之前去畸变的位置，然后将虚拟相机去畸变的图像放在CameraFrame中的ImageData的rgb中，而ori\_rgb保存着原始图像；
- 之前的这种方案将虚拟相机的部分从模型端扩散到了上游即InferenceEngineComponent中，而且修改了原始去畸变的代码，在Component的配置文件中多加了use\_virtual\_camera这个参数，因为其他的Component比如前视车道线以及道路结构都是不用BEV障碍物的这套虚拟相机的；
- 之前的方案不得以将BEV的3D框投影到了虚拟相机去畸变的图像中，然后下游不得不考虑虚拟相机的内外参来配合使用BEV的2D投影框，这也是不合理的；
- 所以这次修改的目的是将BEV障碍物的虚拟相机的“作用域”限制在BEV模型推理的部分，上游以及下游都不需要考虑虚拟相机相关的东西，即对上下游来说是个“黑盒子”；

## 设计思路



- 还原InferenceEngineComponent中的去畸变，删除use\_virtual\_camera的参数，即在InferenceEngineComponent中还是做以前的去畸变，将原图保存至ImageData的ori\_rgb中，将原始去畸变图像保存至rgb中；
- 在BEVPreprocessNode中读取ImageData中的ori\_rgb，配合data\_provider中新加的FillVirtualImageData来获取虚拟相机去畸变的图像，并将该图像保存至BEVPreprocessNode的成员变量中，然后再对该图像进行crop + resize，将结果保存至output的blob中，供下游的BEV模型使用，这样的话虚拟相机去畸变的图像只存在于BEVPreprocessNode中；
- 下游拿到BEV的3D检测结果后利用原始内外参将3D框投影到原始去畸变图像（即ImageData中的rgb）中，然后利用投影的2D去原始去畸变图像中扣取patch图像做后面的patch任务（如车轮车灯等），同时将投影2D框填充至CameraFrame中传到下游使用；

## Commit

评审: [baidu-adu-lab-andes-610 \[general\]\[perception\]\[perf\]](#) keep BEV virtual camera inside of BEV obstacle

# 验证方式

- 保存InferenceEngineComponent中填充后的ImageData中的ori\_rgb和rgb，理论上ori\_rgb应该是原始图像，rgb为原始去畸变图像；
- 保存BEVPreprocessNode中做完虚拟相机去畸变后的图像，理论上和原始去畸变图像是有差异的；
- Dueye可视化的时候不使用虚拟相机的图像，即不给定--enable\_virtual\_camera的参数，理论上andes\_infer中的2D投影框和图像是贴合的，因为这里直接取的线上的投影结果，而线上是投影到原始去畸变图像上的；

# 验证结果

- 在离线x86的机器以及出车测试（2023-03-12下午 ARCF009） 均验证通过

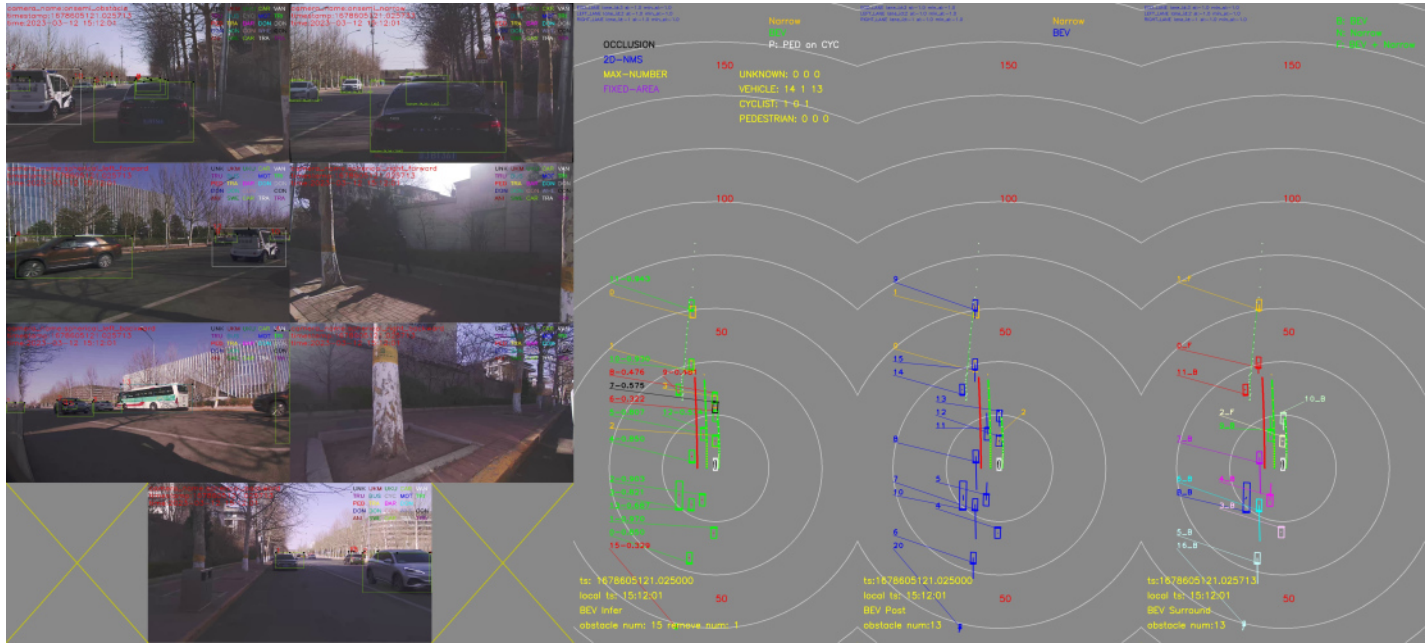
【离线x86】



原始图像（ImageData中的ori\_rgb）

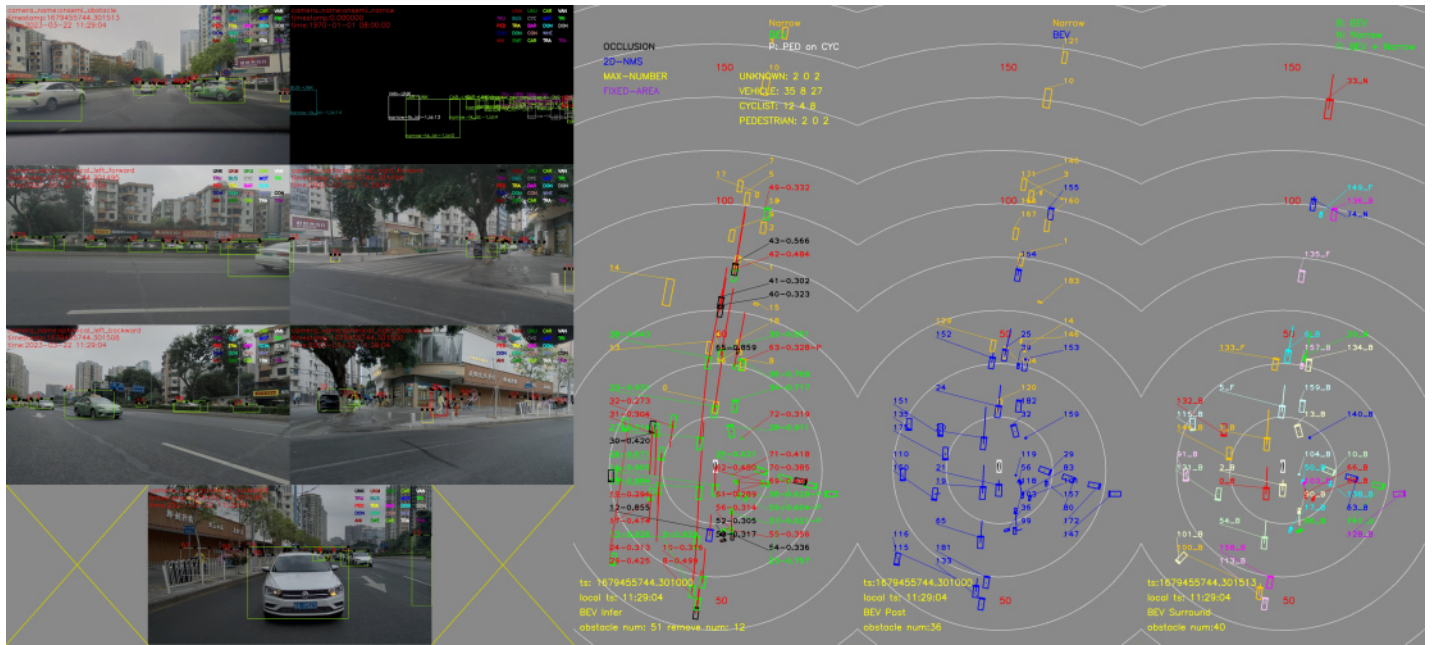
原始去畸变图像（ImageData中的rgb）

虚拟相机去畸变图像（BEVPreprocessNode中的成员变量）



离线用ARCF009落盘数据跑的结果





JIDU2095 0419离线x86

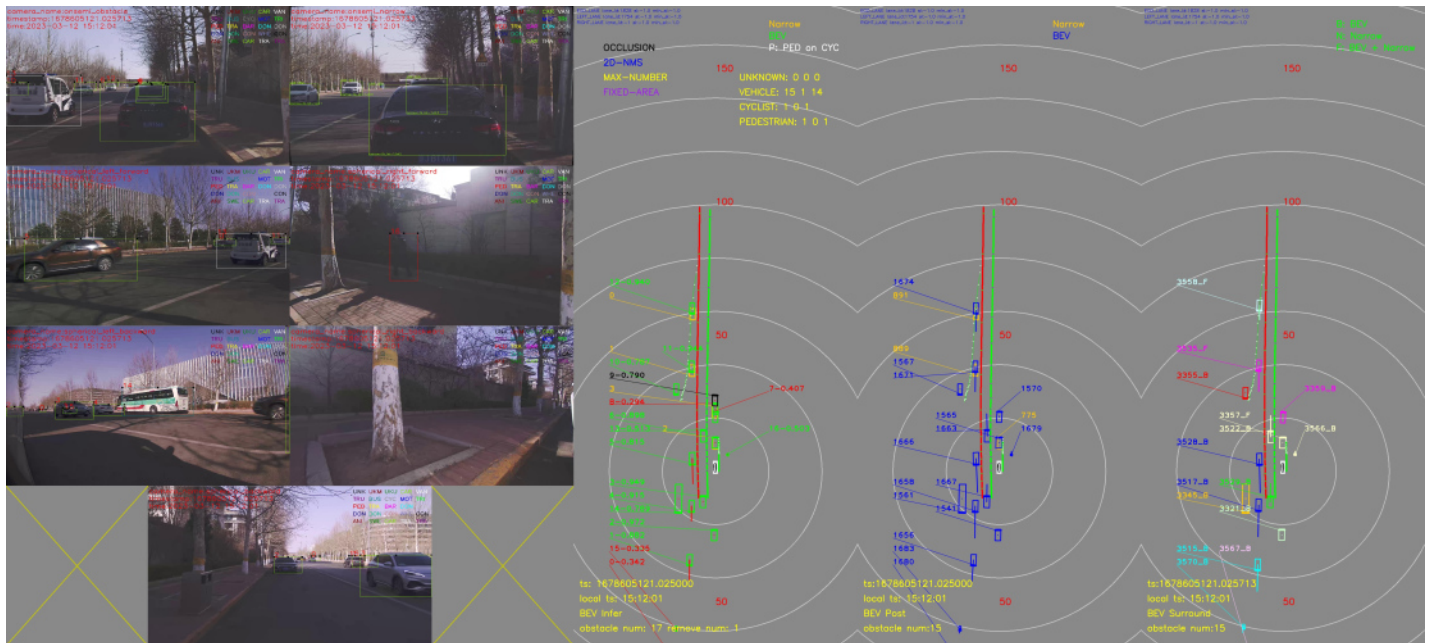
## 【ARCF009】



原始图像 (ImageData中的  
ori\_rgb)

原始去畸变图像 (ImageData中的  
rgb)

虚拟相机去畸变图像  
(BEVPreprocessNode中的成员变量)



ARCF009落盘数据的可视化

## (补充) 可能的风险

## 【背景】

- 修改后BEV模型推理是在虚拟相机去畸变的图像上做的，而3D到2D投影是在原始去畸变图像上做的，由于VP车虚拟相机的侧前旋转比较多（主要是为了和前向有更多的overlap，之前环视也是这么处理的），而原始去畸变没有做旋转，所以侧前可能会有盲区以同一个障碍物在前向和侧前都严重截断的情况，可能会影响patch任务的效果，需要评估一下。

## 【验证方式】

- 可以可视化同一个VP车的record，分别用虚拟相机去畸变图像和原始去畸变图像进行可视化，定性比较一下overlap的差异。

## 【验证结果】

- 从下图可以看到，虽然原始去畸变的侧前没有做旋转，然后由于前向没有改内参，前向的视野更宽，弥补了和侧前的overlap，由于虚拟相机修改了内参，导致前向的视野变窄，同一时刻，反而是原始去畸变前向看到的更多，对前向的patch任务更友好。
- 还有侧前和侧后的overlap看起来比虚拟相机更大，没有相关的风险。



原始去畸变图像

虚拟相机去畸变图像





原始去畸变图像



虚拟相机去畸变图像

相关文档：[ANP CI拦截申请](#)