

## 工作记录

昨天将 2 种生产的方式都调整完毕，今天进行 **depthlayer** 的设置，然后打印深度信息图片，扩大模型中车库的数量，加速模型打印

任务：

1. **depthlayer** 构建，深度信息图片打印
2. 模型车库扩建
3. 加速模型打印

具体实现：

### 1. **depthlayer** 构建，深度信息图片打印

- i. 使用代码进行设置：先在模型中添加 **depth** 层，并按照原来模型进行设置，然后使用代码进行设置

- a) 使用代码将 **render** 打印图片分层，得到 4 层

- i. 摄像头阳光层
- ii. 车辆层
- iii. 背景层
- iv. 地面层

打印可以分离，但是没有深度信息

- b) 错误为缺少 **nodes**，解决方法，使用代码创建 **nodes**,不行,
- c) 使用各类 **Nodes** 接口均无效
- d) 将各类接口使用在原来的模型上，可以行

注：说明原来模型上有 **node**，才可以构建深度信息图

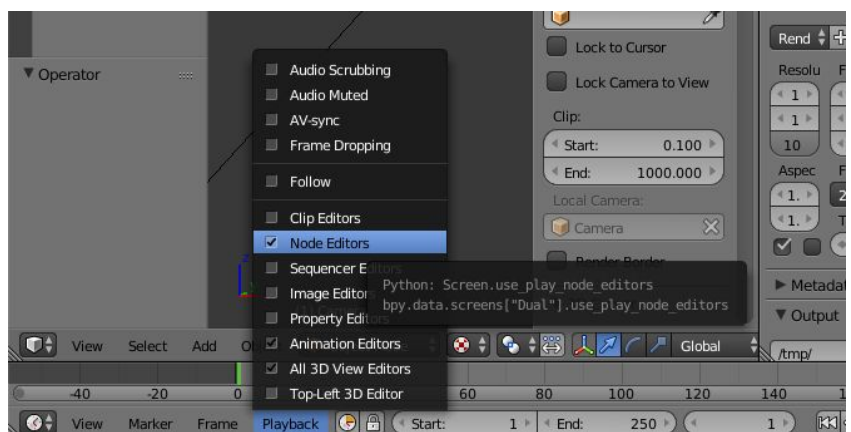
- ii. 使用模型进行 **nodes** 的添加

#### 1. 分析 **node** 的结构

- a) **Node** 的作用
- b) **Node** 的使用
- c) 分析原来模型中宝马车无法打印深度信息图的原因：

- i. 开启 **nodes** 编辑器

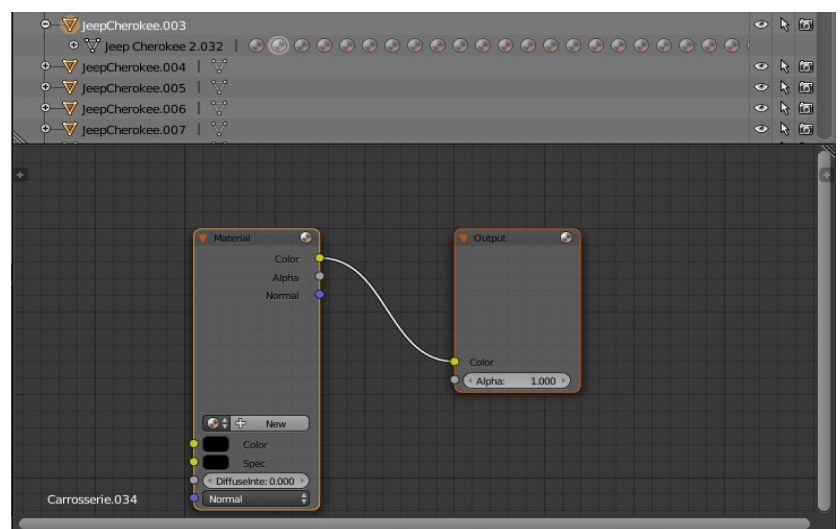
1. 选择并勾选：



## 2. 开启 nodes 视图

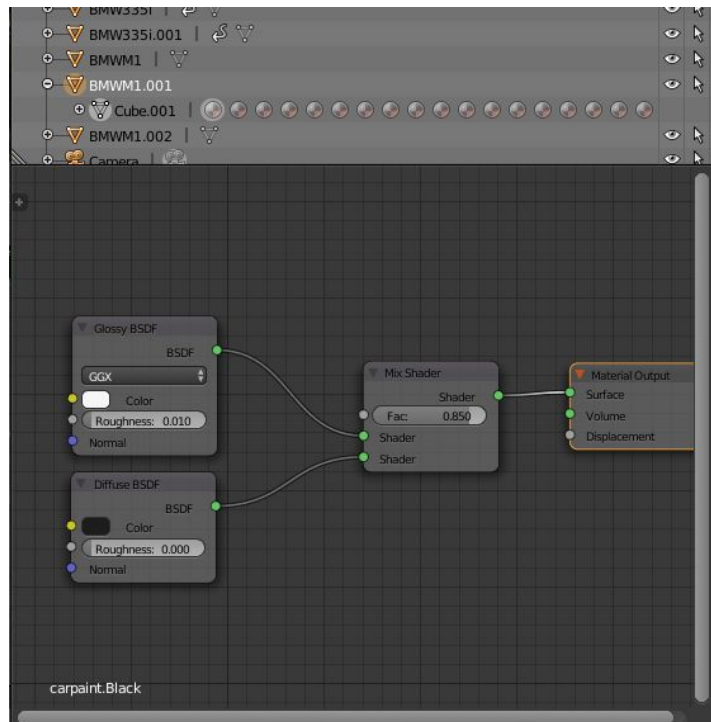


注：开启后，选择车模型，如下如图所示：



注：图中可以看出，jeep 中的 nodes 是 color 连接着输出，对应的是 BMW,没有 nodes

相对应的是新模型的 nodes:

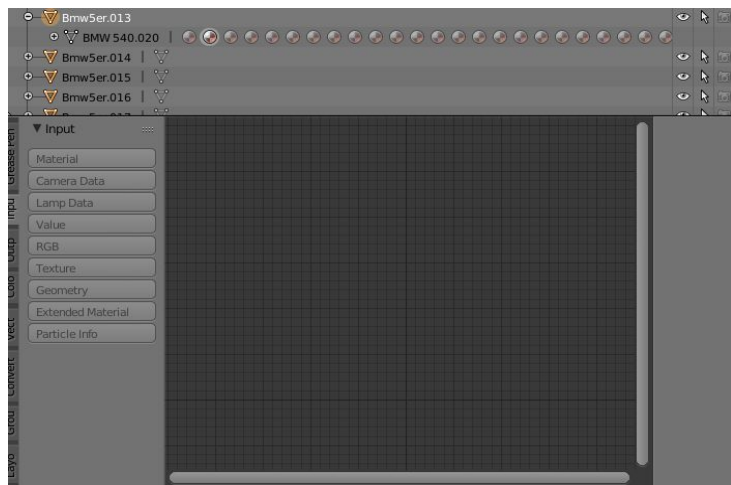


注：新模型中也有 Nodes，相对原来的模型.nodes 加复杂，

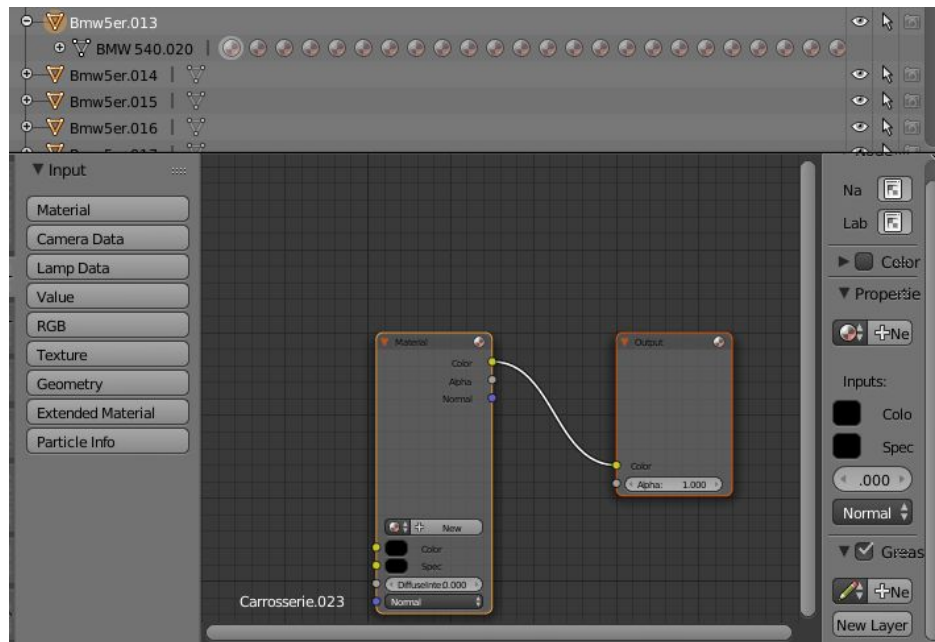
## ii. 分析 nodes

1. 原来模型中 nodes 对应的 color 是不是就是深度信息打印的关键，需要进行实验分析
  - a) 在原来模型中，为少数 BMW 添加相应的 Nodes
  - b) 打印并查看
  - c) 分析：
    - i. 原来模型 BMW 添加相应 nodes 后正确打印深度信息
    - ii. 原来模型 BMW 添加相应 nodes 后无法打印深度信息

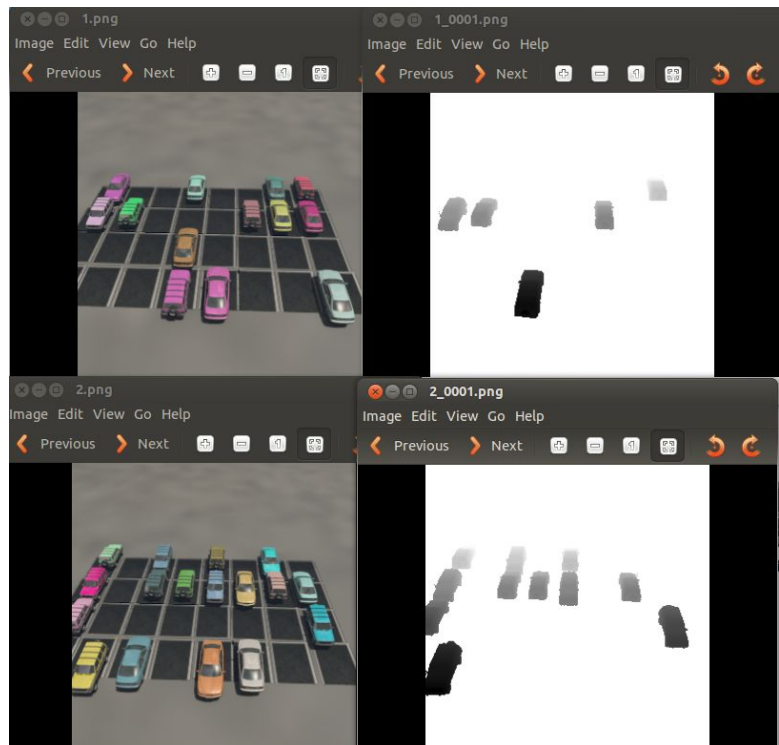
注：实验最后发现，BWM 也有 nodes，只是有些不在第一个 material 中，选中即可，如图所示：



选中第一个 material 时:



模型有 nodes,说明 material 的 nodes 不是深度打印的关键,部分打印图如下所示(BWM 无法打印深度信息图的原因暂时无法找到):

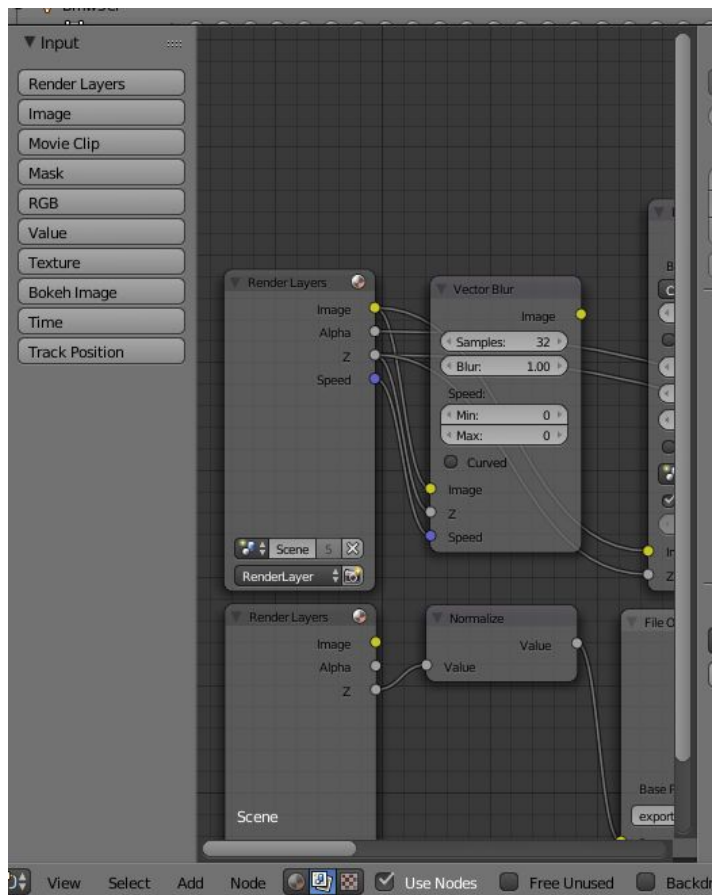


注: BWM 无法正确打印

2.material 的 nodes 不是打印的关键,分析 layer 的 nodes

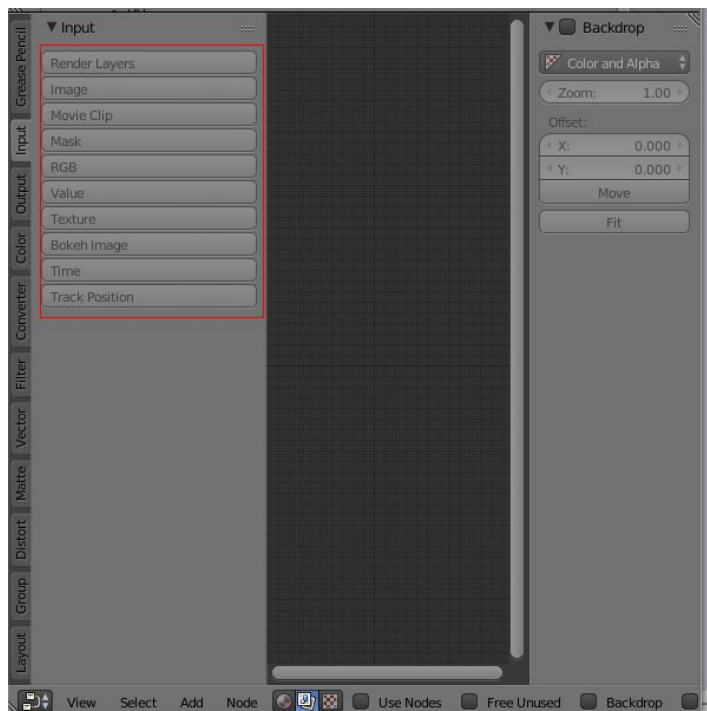
a) 原来的模型有 layer nodes,新的模型没有 layer nodes,尝试进行添加,观察最终结果

## i. 原来模型的 layer nodes



注：应该是Z决定了最终的深度图像的生成

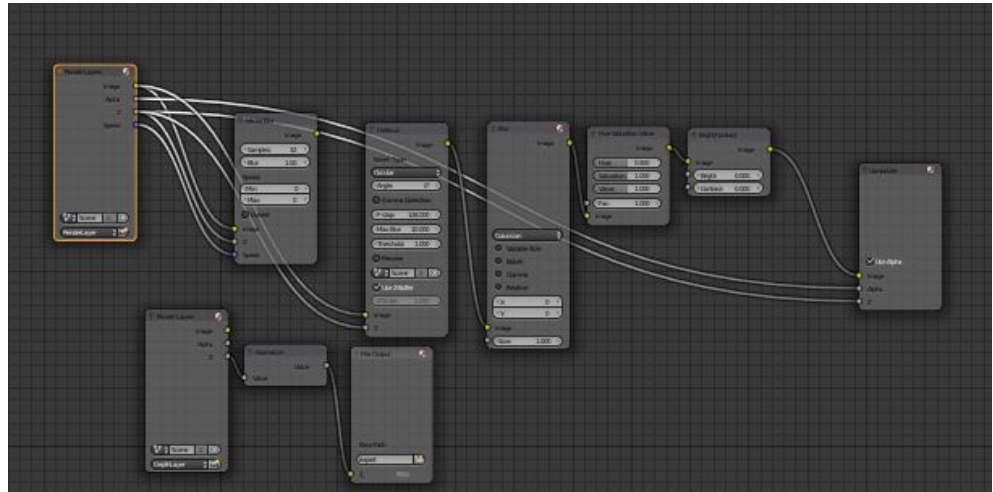
## 2. 新模型的 layer nodes:



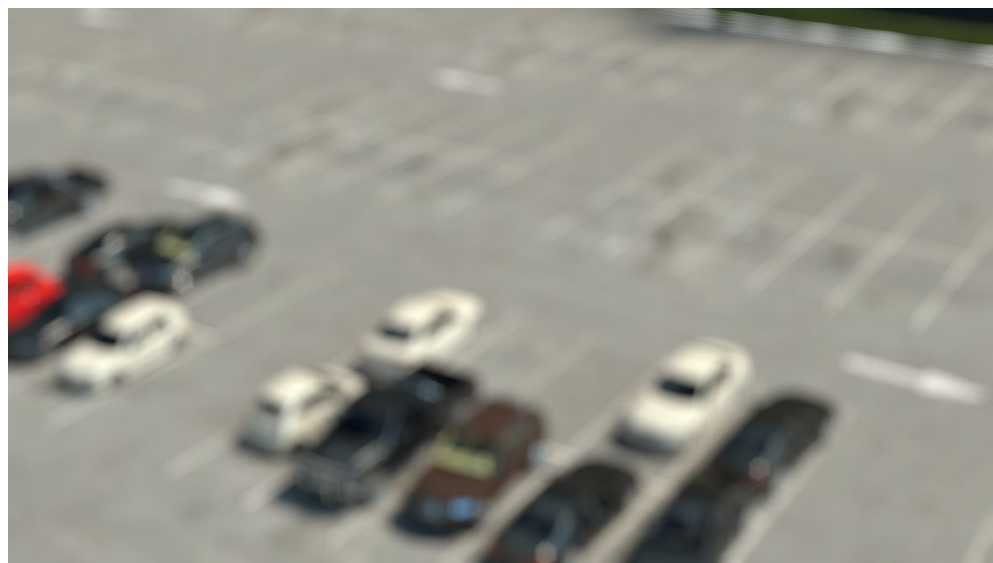
注：由图可知新模型没有 layer nodes，也不能直接直接点击制作

查找资料无效后选择使用代码进行解决，接口提示中缺少了一个参数，没有做好，查看 <http://blender.stackexchange.com/> 也没有找到答案，多次对比后，发现底部的 **USE NODES** 没有勾选，细节决定成败，不仔细看浪费时间了

按照原来模型进行设计后：得到如图所示的 nodes 图：



打印后的 render layer 如下图所示:



注：原因是因为使用了 **blur** 层，删除即可，再次运行，跑了 7 分钟后图还没有打印，问题应该在添加了多个打印层的问题上，将 **render layer** 的 **nodes** 删除即可



最终打印效果：



b) 其他方法

暂时未做

代码：

```
nodes = bpy.context.scene.node_tree.nodes
links = bpy.context.scene.node_tree.links
nodes['File Output'].base_path = output_dir
```

```
filepath = output_dir + '/images/' + str(i + 1) + '.png'
bpy.context.scene.render.filepath = filepath

nodes['File Output'].file_slots[0].path = '/images2/' + str(i + 1) + '.png'
print('Rendering car no. ' + str(i + 1))

bpy.ops.render.render(write_still = True)
annot_file.flush()
```

事物性的：

1.购买内存条时又出现问题，发票改开为 2 张，但是本次疏忽，没有注意，以为卖家和上次一样，但是卖家开为一张，最终价钱超过报销的金额，虽然是货到付款，最终没有签收，但是这种事提醒自己时刻注意，任何事都得认真，得负责，做事慢点可以，但是要仔细，不要留下出错的隐患

2.给老台式机安装 win7，为了后面使用，保留了 x p，双系统方式安装，win7 xp 共存，测试 nodemcu 过程中，依旧出现闪退问题，查看笔记本配置，发现应该是 .Net 版本问题，升级到 .Net4.5 即可，最终解决，李振威师兄之前遇到的闪退应该也是这个，只是之前笔记本损坏，没有帮师兄找出问题所在

