论文中出现了一些算法。我对出现过的部分算法进行了一定的查阅和了解。可能对项目本身没有太大作用，只是加深理解。

以下是一些摘要。

1. CNN

卷积神经网络（Convolutional Neural Network, CNN）是一种 前馈神经网络，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。它包括卷积层(alternating convolutional layer)和池层(pooling layer)。

卷积操作

一个图像矩阵经过一个卷积核的卷积操作后，得到了另一个矩阵，这个矩阵叫做特征映射（feature map）。每一个卷积核都可以提取特定的特征，不同的卷积核提取不同的特征，举个例子，现在我们输入一张人脸的图像，使用某一卷积核提取到眼睛的特征，用另一个卷积核提取嘴巴的特征等等。而特征映射就是某张图像经过卷积运算得到的特征值矩阵。

讲到这里，可能大家还不清楚卷积核和特征映射到底是个什么东西，有什么用？没关系，毕竟理解了CNN 的卷积层如何运算，并不能自动给我们关于 CNN 卷积层原理的洞见。为了帮助指导你理解卷积神经网络的特征提取，我们将采用一个非常简化的例子。

特征提取

特征提取指的是使用计算机提取图像信息，决定每个图像的点是否属于一个图像特征。特征提取的结果是把图像上的点分为不同的子集，这些子集往往属于孤立的点、连续的曲线或者连续的区域。

至今为止特征没有万能和精确的定义。特征的精确定义往往由问题或者应用类型决定。

卷积核在图像上不断滑动运算，就是卷积层所要做的事情。同时，在内积结果上取每一局部块的最大值就是最大池化层的操作。CNN 用卷积层和池化层实现了图片特征提取方法。

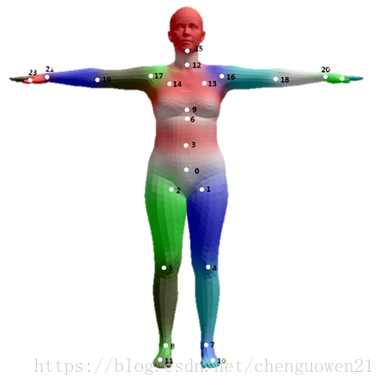
反向传播算法

反向传播，就是对比预测值和真实值，继而返回去修改网络参数的过程，一开始我们随机初始化卷积核的参数，然后以误差为指导通过反向传播算法，自适应地调整卷积核的值，从而最小化模型预测值和真实值之间的误差。

1. SMPL

  SMPL模型是一种参数化人体模型，是马普所提出的一种人体建模方法，该方法可以进行任意的人体建模和动画驱动，采用了不同姿势的真实人体网格。这种方法与传统的LBS的最大的不同在于其提出的人体姿态影像体表形貌的方法，这种方法可以模拟人的肌肉在肢体运动过程中的凸起和凹陷。因此可以避免人体在运动过程中的表面失真，可以精准的刻画人的肌肉拉伸以及收缩运动的形貌。

该方法中β和θ是其中的输入参数，其中β代表是个人体高矮胖瘦、头身比等比例的10个参数，θ是代表人体整体运动位姿和24个关节相对角度的75个参数。



SMPL 模型关节点

而因为SMPL是裸的人体模型，但该实验的目标是包含衣服头发等的非裸露模型，所以增加了补偿项。

关于SMPL可参考 http://files.is.tue.mpg.de/black/papers/SMPL2015.pdf

1. KinectFusion

kinectfusion是微软研究院利用kinect进行三维重建的项目。

KinectFusion重建的基本流程是：首先，处理采集到的原始深度图，获取点云voxel的坐标以及法向量坐标（Depth Map Conversion）；接着，根据当前帧的点云和上一帧预测出的点云计算当前相机的位置姿态（Camera Tracking）；然后，根据相机位置姿态更新TSDF值，融合点云（Volumetric Integration）；最后根据TSDF值估计出表面（Raycasting ）。

pre-work

在进行下面的处理处理之前，kinectFusion实际还对原始的深度信息进行了一定的降噪平滑，采用双边滤波（Bilateral filtering），在保留边缘的基础上进行平滑。双边滤波在空间域加权平均的基础上再对值域加权平均，即像素灰度值越靠近中心像素的灰度值，权重越高。在边界附近，灰度值差异很大，所以虽然边界两边的像素在空间域靠在一起，但是由于灰度值差别非常大，对于互相的权重很低，所以可以保持清晰的边界。

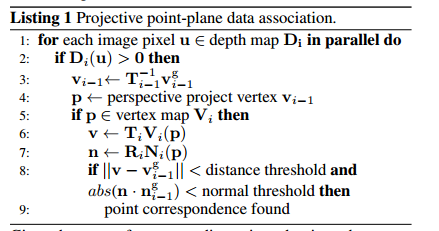
Depth Map Conversion

主要是求出有原先的图像点u=(x,y)，以及深度值D(u)，求得每个点的法向量n(u)。根据相机的内部矩阵，将图像2D坐标转化为相机原点坐标系的3D点。

IMG_256

Camera Tracking

此步骤用ICP（iterative closest point）算法，求解出相机每次的相对位移与转动。相机位置可以用来将相机原点坐标系的结果转化到世界坐标系。



每次迭代的基本思路就是：

1.筛选：点集或曲面的筛选（滤波）

有全选，随意筛选，均匀分布，特征筛选等多种方法，kinectFusion中使用的应该是最朴素的全选（由于使用了GPU，并行化增加的前提下简单的算法有无可比拟的优势）

2.匹配：两个点集之间的点进行配对

注意这并不是要保证匹配的点对是真实的匹配，因为本身ICP是迭代计算出匹配点并将其拟合的收敛过程。我们需要做的是每次选取合适的初始值。

KinectFusion采用的是投影关联(projective data association)。简单来说就是用上一个世界坐标系中的点，先转化成上个相机坐标系点vi−1vi−1（3D），再转化成上一个图像坐标系点P（2D），然后找到本次图片中同样的2D坐标点，用当前的Ti值和Ri值去计算出点v及其法向量n，最后判断v和n与其对应点的相容性（其实是第4步去除的操作）。这样就完成了一次找匹配点的计算。

3.权重：给每个匹配的点对分配权重

4.去除：去除不符合条件的点对

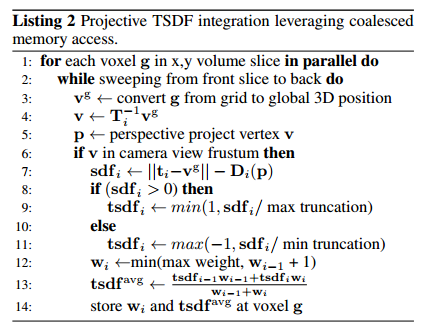
5.误差度量：基于以上点对，给出每个点对的误差计算方法

6.最小化：最小化误差度量

经过上面的一次迭代，我们找到一堆匹配点，并求出其中使得匹配度最优（可以取类似最小二乘的值）的T，然后将本次测量的深度图进行相应的变换来进行下次迭代。

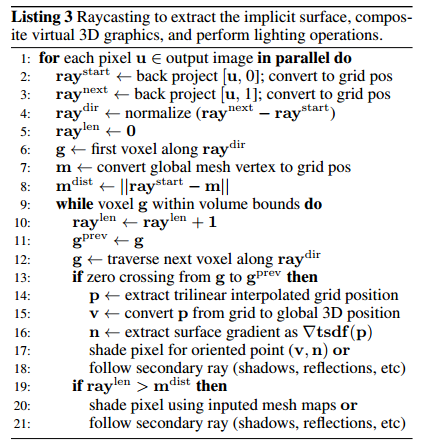
Volumetric Integration

上一环节已经形成了每个相机图像个子的T和R，可以将每一帧采集的数据转化成世界坐标系当中。通过本环节的操作，我们可以形成水密的物体重建，采用TSDF（Truncated Signed Distance Function）的方法。



Raycasting

之前的操作只是形成了IMG_256表示的物体的重建结果，要形成我们能见到的图片，还需要相应的渲染和投影，主要算法如下。



参考资料：

1. <https://zhuanlan.zhihu.com/p/27908027>
2. <https://zhuanlan.zhihu.com/p/50800849>
3. <https://blog.csdn.net/chenguowen21/article/details/82793994>
4. <https://blog.csdn.net/xiaohu50/article/details/51592503>
5. <https://blog.csdn.net/xiaohu50/article/details/51592503>