1 Introduction

本项目主要的研究内容是基于室内场景的Multiview下的语义分割及场景理解。主要是针对实际采集的场景序列图像的优化分割，因为在单视角下得到的分割结果由于室内场景的条件限制如光照等，会对一些物体的分割结果产生影响。所以此处考虑采用multiple view多视角下得到的图片，利用帧间信息的互补性提高分割的准确率。

在我们的项目中，主要是通过实际采集室内场景的序列图片，然后通过语义分割网络得到每一帧的分割结果，之后再通过采集系统输出的外参矩阵及深度图得到2D到3D的映射结果，主要是为了能在3D空间建立多帧之间的联系，之后再通过投影操作优化2D上的分割结果。

2 Related Work

2.1 Single Frame Segmentation

正因为深度学习的发展，才推动了图像语义分割的结果。这里主要侧重于深度学习方法下的语义分割。

Silberman [错误!未找到引用源。] 等人首次将图像深度信息融入至室内场景领域的语义分割中，分类器使用带反馈的前向神经网络，从场景中获取非监督的图像并学习特征描述子，卷积网络由几个部分组成，每个部分均包含三个层次。第一层是滤波器组层，其将输入与滤波器组进行卷积；第二层是非线性层，其将数据映射至一个不同的空间；最后一层是特征池层，其用于获取最终的输出特性。将上述包含三个层次的部分进行串联，最终的输出特征描述子可用于训练分类器。每个提供给卷积网络的尺度信息，都将用于构建对应于不同尺度的特征描述子。然后分类器可以为每个像素进行语义标签推断并将推断结果融合至超像素中，获取非监督的图像语义标注结果。因为所使用的方法的局限性，从分割直观结果上来看并不是太理想。

14 年，Yann LeCun 等人用多尺度卷积网络直接从图像和深度信息学习特征，主要为解决室内场景与 RGB-D 输入的多类分割。使用图像和深度图对室内场景进行语义分割，结构如图，主要利用深度图和 RGB 图像先用不同尺度的拉普拉斯算子滤波，在不同尺度上进行卷积特征提取，然后将不同尺度上的特征图融合在一起，于此同时 RGB 图像上利用超像素分割（把相近的像素分在一个不规则的小 patch 里），分类器利用学到的特征和预分割结果，对超像素的每个 patch 进行分类。并通过利用场景的视频序列中的时间一致性来进一步改进标记。提出一种从视频中产生时间一致的超像素的方法。主要是用于视频分割领域。

一方面，超像素分割并不稳定，小块物体或者单个物体中存在杂色时，分类存在很多错误；另一方面，弱边界区域，本身对超像素分割也很难处理，所以这种方法的效果并不是太出色。

Ren [错误!未找到引用源。] 等提出了一种性能较好的室内场景语义分割方法。主要原理是充分利用超像素区域化结构来构建树状模型，并将构建的模型与 MRF 随机场的概率图模型相结合。在此基础上进一步训练 SVM 分类器，为 RGB 图像中的超像素提供类别语义标签。虽然该方法在一定程度上可以提高语义分割的准确率，但

是整体方案过于复杂，不利于实时性操作。

Nico[错误!未找到引用源。]等人提出了一种为CNN提供深度信息的新方法，用一个简化的HOD定向深度直方图描述RGBD的深度通道。采用四阶卷积神经网络，建立在已有框架下，采用整流非线性方法、预处理以及提供深度信息等，因为使用来自多个尺度的输入，因此卷积更少更快。

Wang[错误!未找到引用源。]等利用反卷积网络结构，预测像素级别的语义标签，提出了一种桥接CNN和反卷积网络的转换网络，在特征变换网络中，通过发现两种模态之间的共同特征来联系两个模态，以及通过分析两种结构发现模态特性，并且将两种模态的特性相融合，目标提高分割准确性。

15年，Jonathan Long等的核心观点是建立“全卷积”网络，输入任意尺寸，经过有效的推理和学习产生相应尺寸的输出。他们定义并指定全卷积网络的空间，解释了它们在空间范围内预测每个像素所属的类别中的应用并结合联系了之前的模型。FCN的出现，算是真正开启了像素级别的语义分割网络。修改了当前的主要分类网络(AlexNet，the VGG net and GoogLeNet)到全卷积网络，并通过精调将训练好的标记迁移到分割任务中。然后定义了一个跨层架构，将来自深、粗层的语义信息和来自浅、细层的表征信息相结合，来产生准确和精细的分割。

如下图，FCN沿用了AlexNet卷积网络的结构，但是在全连接层的地方，为了能够生成图像的像素级预测，就扩大了卷积阶段和全连接层的plane size，最后全链接层的特征向量变成了特征图，由于最终的feature map比原图小，所以还需要进行上采样。

下面这张图更加清晰地说明的网络结构的变化，上面的就是AlexNet网络，下面是改变后的，首先为了使学习到的feature map更大，作者把训练输入的图像从224x224 扩大为500x500，全连接层输出的feature map 的大小为10x10。

作者比较了AlexNet， VGG16， GoogLeNet的性能，在精度上来讲，VGG16最好，但是由于参数规模比较大，所以也比较耗时。而且不同深度level的卷积学习到的特征抽象程度是不同的，浅层的学习到的都是局部特征，而随着卷积层深度增加，感受野也随之增大，学习到的特征更加抽象，作者测试了使用不同卷积阶段的feature map 生成分割结果，并进行对比，发现Pool3后得到的分割结果最佳，太过浅层学习的feature 对局部变化比较敏感，也就是抗噪性能不

佳，而更高层的学到的 feature map 对于局部变化不敏感，但很严重的缺点就是，梯度消失或者说边界模糊，也就是只能得到一个笼统的预测，但是很难获得准确的分割边界。

关于 FCN-8s 的网络结构如下，方法就是图中画的那样，全连接输出的预测 feature map 上采样 2 倍后，跟 pool4 后的预测 feature map 叠加在一起，再上采样 2 倍，跟 pool3 后的预测 feature map 叠加在一起，最后整体上采样 8 倍，得到最终的预测结果。结果如下：

由于 FCN 的出现，之后有很多的改进和升级版本，如疏松的卷积核方法，可以达到在不增加计算量的情况下增加感受域，弥补不进行池化处理后的精度问题。以及以 CRF 为代表的后期优化处理，CRF 将图像中每个像素点所属的类别都看作一个变量，然后考虑任意两个变量之间的关系，建立一个完全图。因为 FCN 是像素到像素的影射，所以最终输出的图片上每一个像素都是标注了分类的，将这些分类简单地看成是不同的变量，每个像素都和其他像素之间建立一种连接，连接就是相互间的关系。这种 FCN+CRF 又叫做 DeepLab，主要原理是通过卷积，全连接后输出一个粗糙的，比原始图像小得多的预测图，然后把预测图上采样，然后使用条件随机场迭代优化。

16 年，Lin 等人利用图像的上下文信息提升 DCNN 的图像语义分割能力，探索了图像中“区域-区域”和“区域-背景”的上下文信息。对于“区域-区域”的上下文信息，构建了基于 DCNNs 和 CRFs 的深度模型用以学习不同图像区域块之间的语义关联。提出的高效分段训练方法结合深层结构模型，以避免在反向传播过程中重复的 CRF 推断。对于“区域-背景”的上下文信息，采用一种多尺度图像输入和滑动金字塔池化的方式获取。

基于 FCN 架构的方法，基本特点都是使用卷积学习网络全连接层的粗糙的 feature map，上采样生成与原图一样大小的预测结果，这样的做法，主要缺点有两个：第一，因为扩大了卷积层和全连接层的 plane size，使得网络参数规模巨大，VGG16 框架下，参数规模为 134M，这就导致计算量的增大；第二，受限于显卡内存和计算量的限制，在最大化扩大全连接层输出的 plane size 后，得到的 feature map 比原图小的太多，FCN 里输入图像是 500x500 的分辨率，而全连接输出的 feature map 大小只有 10x10，用这样粗糙的 feature map 上采样，很

难得到很精细化的分割结果。

因为 FCN 的一些缺点，之后就有一些探索，首先就是 Encoder-Decoder 系列的，这种方法主要是围绕着直接把全连接或者某一阶段的卷积 feature map 上采样可能过于简单的问题，设法换一种上采样方法。

主要有 DeconvNet、DecoupledNet、CEDN、segNet 等。

DeconvNet 主要考虑到卷积阶段是一层一层进行下来，那么将整个卷积过程镜像过来，就得到卷积阶段的整个逆过程，为了减少一些参数规模，作者将全连接层缩减成一层，而且 plane size 也缩小，之前提到的 FCN 为了扩大全连接 feature map 的 plane size 把输入图像的大小扩大，全连接输出也扩大，因此，DeconvNet 的做法似乎没有从本质上解决之前 FCN 的问题。不足就是去卷积阶段其实利用的 feature map 仅仅是卷积阶段最后的 feature map，没有从本质上解决之前 FCN 的问题。但是从结果上来看精度上有一定程度的提升。

DecoupledNet 则是在去卷积逆过程的时候，把卷积阶段和全连接都镜像过来，主要问题是原本 FCN 的结构网络参数就很大，这样参数规模相当于扩大了 2 倍，而且有和第一种方法一样的不足。

CEDN 相比前面的两个网络，做了很多简化，首先全连接层去除掉，卷积之后直接就开始去卷积，而去卷积过程也并非是卷积阶段的完全镜像，也做了缩减。但这种方法主要考虑的不是室内场景的语义分割。

从前面的方法来看，Encoder-Decoder 的演变趋势是，Encoder 后的全连接或者卷积层逐渐简化到消失，SegNet 就是这样一个简化版的网络，所有 Encoder 后紧跟着就开始 Decoder。SegNet 在上采样上进行了新的尝试，之前的 FCN 网络上采样也提到过，使用深层的卷积 feature map 上采样后，跟卷积阶段对应大小的 feature map 叠加在一起，上采样操作是学习得到的，segNet 使用一种叫 Max-Pooling 索引的方法，在 encoder 阶段，每个 pooling 层都输出两个 pooling 的 feature map，一个输出到后面的卷积层，一个在用在 decoder 阶段的上采样，上采样的过程不是学习得到的。

SegNet 算是 Encoder-Decoder 方法种的最精简的方式，没有全连接层，上采样层参数也不通过学习得到，主要是为了减小网络参数规模，减小计算量。

PSPNet 能够聚合不同区域的上下文信息，从而提高获取全局信息的能力。在

一般 CNN 中感受野可以粗略的认为是使用上下文信息的大小，论文指出在许多网络中没有充分的获取全局信息，所以效果不好。要解决这一问题，常用的方法是：1. 用全局平均池化处理。但这在某些数据集上，可能会失去空间关系并导致模糊。2. 由金字塔池化产生不同层次的特征最后被平滑的连接成一个 FC 层做分类。这样可以去除 CNN 固定大小的图像分类约束，减少不同区域之间的信息损失。论文提出了一个具有层次全局优先级，包含不同子区域之间的不同尺度的信息，称之为 pyramid pooling module。

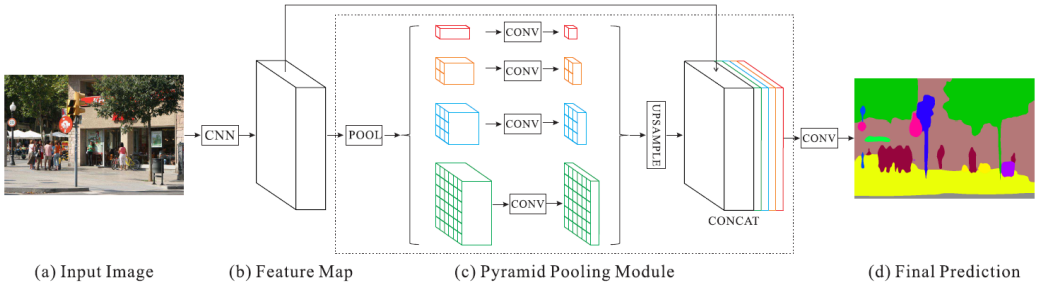


Figure 3. Overview of our proposed PSPNet. Given an input image (a), we first use CNN to get the feature map of the last convolutional layer (b), then a pyramid parsing module is applied to harvest different sub-region representations, followed by upsampling and concatenation layers to form the final feature representation, which carries both local and global context information in (c). Finally, the representation is fed into a convolution layer to get the final per-pixel prediction (d).

该模块融合了 4 种不同金字塔尺度的特征，第一行红色是最粗糙的特征－全局池化生成单个 bin 输出，后面三行是不同尺度的池化特征。为了保证全局特征的权重，如果金字塔共有 N 个级别，则在每个级别后使用 1×11×1 的卷积将对于级别通道降为原本的 1/N。再通过双线性插值获得未池化前的大小，最终 concat 到一起。在 ResNet101 的基础上做了改进，除了使用后面的 softmax 分类做 loss，额外的在第四阶段添加了一个辅助的 loss，两个 loss 一起传播，使用不同的权重，共同优化参数。后续的实验证明这样做有利于快速收敛。

论文在结构上提供了一个 pyramid pooling module，在不同层次上融合 feature,达到语义和细节的融合。 模型的性能表现很大，但感觉主要归功于一个良好的特征提取层。

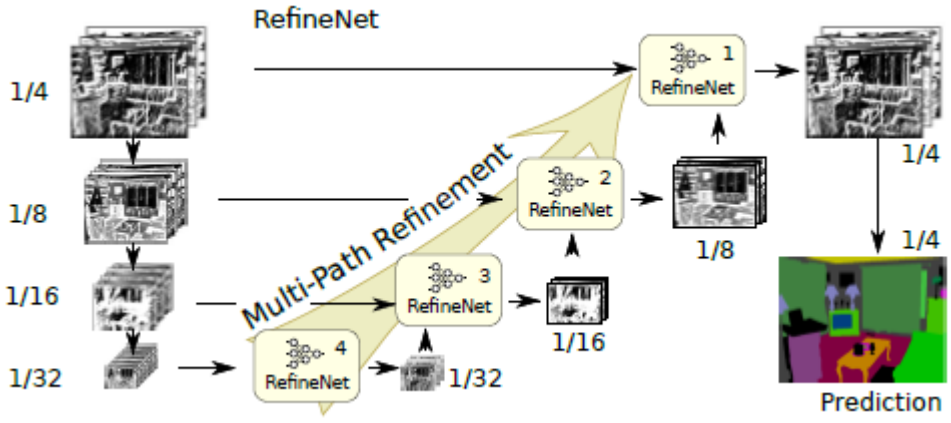
最近的 CNN 网络，如 VGG，ResNet 等在识别中变现很好，但这些方法存在明显的限制当解决密集预测任务如语义分割时，下采样操作使得最终的图像信息损失很大。

一种方式是利用反卷积上采样信息，但这种方式无法恢复底层损失的信息。

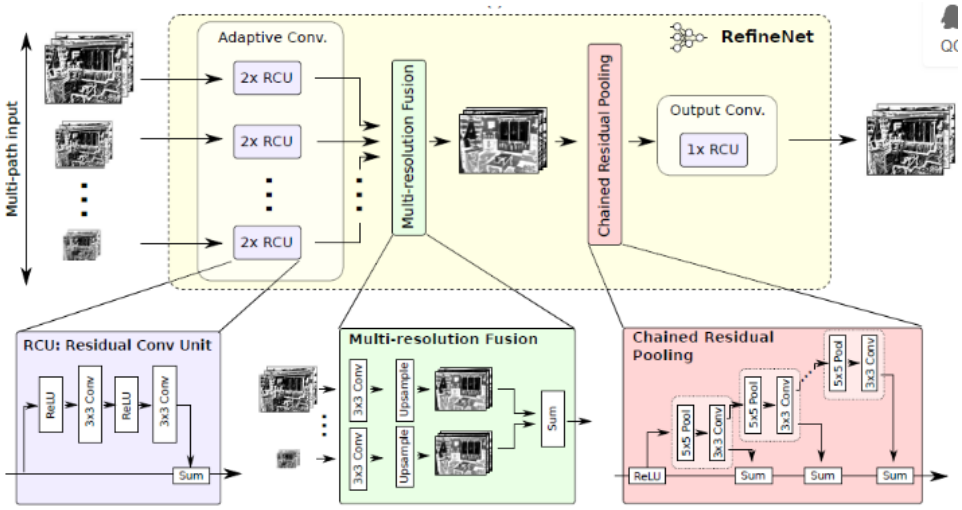
因此它不能生成准确的高层预测结果。底层信息是准确预测边界和细节的基础。DeepLab 提出 dilated 卷积来在不下采样的基础上扩大感受野。但这种策略有两个限制。第一，需要计算大量的卷积特征图在高纬上而引起（computational expensive）+（huge GPU memory resources）。这限制了计算高层特征和输出尺度只能为输入的 1/8。第二，dilate 卷积引起粗糙下采样特征，这潜在导致重要细节的损失。

另外的方法融合中层特征和高层特征。这种方法基于中层特征保持了空间信息。这种方法尽管补充了特征如边界、角落等，但缺乏强大的空间信息。

RefineNet 由许多特殊设计的组件组成，这些组件可以 refine 粗糙的高层语义信息通过融合底层的视觉特征。尤其是，RefineNet 使用长距离和短距离的残差链接使用 identity 映射来优先训练整个端对端系统。提出的网络模型可以分为两段对应于 U-Net 中向下（特征逐步降采样同时提取语义特征）和向上（逐步上采样特征恢复细节信息）两段通路。其中向下的通路以 ResNet 为基础。向上的通路使用了新提出的 RefineNet 作为基础，并作为本通路特征与 ResNet 中低层特征的融合器。



其中左边的四组特征是从 ResNet 的四个对应的 block 取出的。此框架与 U-Net 没有太大区别。不过如作者所说，RefineNet 是一个灵活的模块，其输入的尺度个数可以变化，因此整个网络的拓扑结构可以有很多改变。



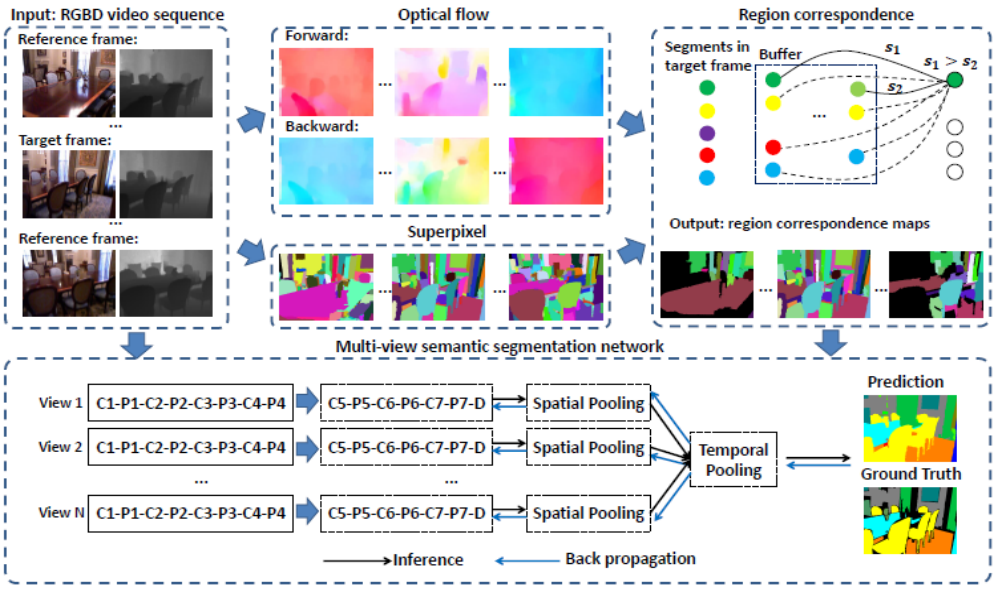
不同尺度（也可能只有一个输入尺度）的特征输入首先经过两个 Residual 模块的处理；之后是不同尺寸的特征进行融合。当然如果只有一个输入尺度，该模块则可以省去。所有特征上采样至最大的输入尺寸，然后进行加和。上采样之前的卷积模块是为了调整不同特征的数值尺度；最后是一个链式的 pooling 模块。其设计本意是使用侧支上一系列的 pooling 来获取背景信息（通常尺寸较大）。直连通路上的 ReLU 可以在不显著影响梯度流通的情况下提高后续 pooling 的性能，同时不让网络的训练对学习率很敏感。最后再经过一个 Residual 模块即得 RefineNet 的输出。RefineNet 的一个特点是使用了较多的 residual connection。这样的好处不仅在于在 RefineNet 内部形成了 short-range 的连接，对训练有益。此外还与 ResNet 形成了 long-range 的连接，让梯度能够有效传送到整个网络中。作者认为这一点对于网络是很有好处的。

2.2 Multiview Segmentation

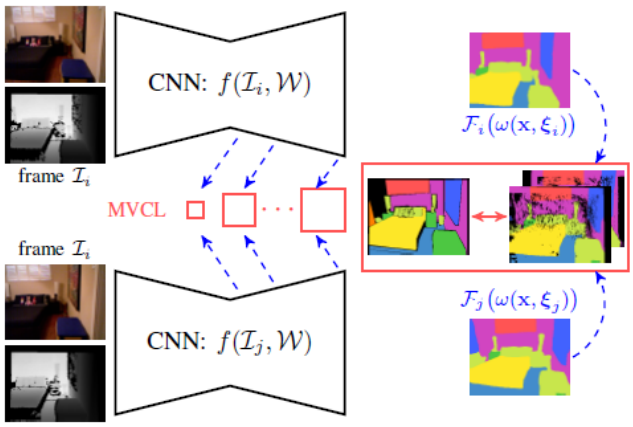
使用多视角下的分割主要是为了有效的利用帧间的互补关系来提高分割效果。现有研究中，Dense 3d semantic 那篇主要是通过贝叶斯融合的方式将结果 map 到 3D 模型中。

He 等人提出了一种数据驱动池化的网络结构用来最终的融合多帧的分割结果，并且将边缘检测的方法用到了语义分割上，使用光流法和超像素分割的方法建立帧与帧之间的联系，并且建立了一个端到端的网络结构来得到帧间的一致性。主要是为了解决之前方法中边缘分割效果不理想的问题。这里随机选择多帧图像中的一帧作为目标帧其余为参考帧，利用光流法和超像素分割的方法建立区域一

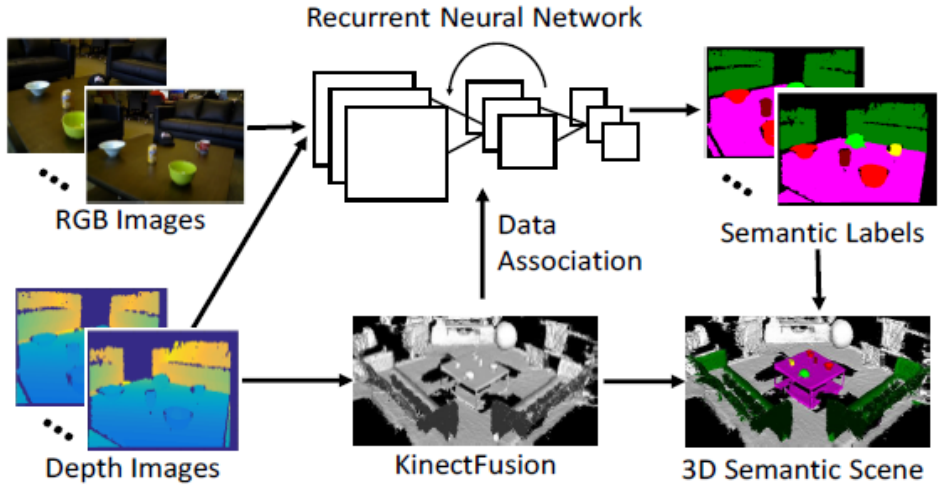
致性，然后设计了可以得到多帧图像对应关系的端到端训练网络，网络是基于FCN的改进，主要是通过数据驱动池化层来融合多视角下图像的信息。如图所示，输入是RGBD序列，首先对每一帧计算其超像素，然后利用光流法建立区域间的对应，通过文中提出的数据驱动池化将信息在网络中融合，建立区域间的对应主要是为了将参考帧信息融合到目标帧中。从网络结构可以看出数据驱动池化有两个pooling，一个是spatial pooling主要是建立feature map和超像素map的对应融合， 另一个是temporal pooling，主要是将multiple frame的信息融合到一帧，最后得到结果。



同样是17年一篇使用multiview的方式的paper。它的网络是基于FuseNet，将RGB与深度depth信息融合，并将网络结构做了相应的改进，增加了多尺度的loss minimization，并且利用RGBD SLAM的方法获得相机的移动轨迹，在训练阶段将经过网络得到的结果wrap到带标签的groundtruth图片中，在测试阶段将从multiple views得到的结果融合至keyframes中。如左图所示，主要的创新点是增加了一个MVCL层（multi-view consistency layers），它可以基于获得的SLAM轨迹估计，获得帧间对应关系，将网络预测结果或者feature map wrap至一个reference view。



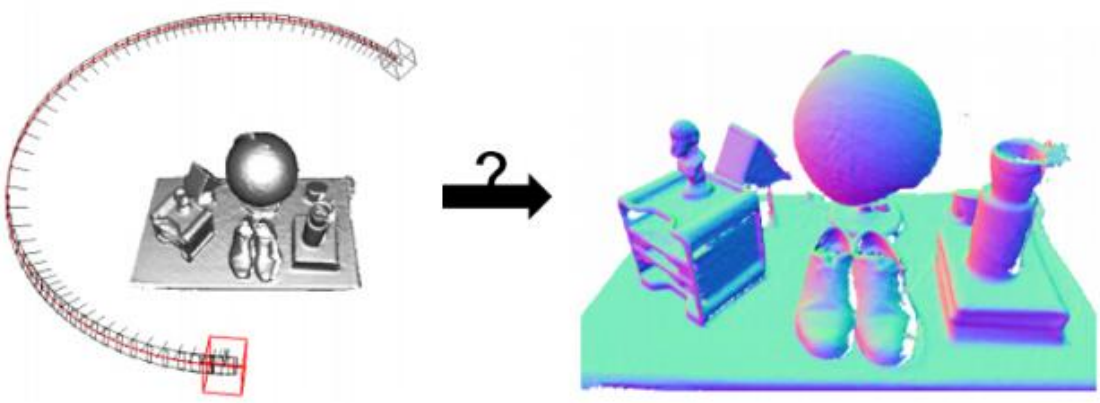
DARNN 利用了 KinectFusion 结合 RNN 建立 multiview 多帧图片之间的联系。作者在文中设计了一种 DA-RNN 的网络结构，主要是在 RNN 结构中增加了一个循环单元 DA-RU，每个单元对应输入图像的一个像素。输入 RGB 和 depth 进入网络结构后，使用 KinectFusion 建立 RGBD 各个帧间的 data association，然后经过网络得到的分割结果生成 3D 的语义分割场景。网络结构可以更好的看出多帧信息融合。如图红色部分代表的是循环层，主要设计的是将先前帧图像与当前帧图像信息结合起来如箭头所示。并且将 DA-RNN 与 KinectFusion 结合用于 3D 场景的分割，得到相邻两帧图像的相机 pose，然后计算两帧的 data association（将一帧图像投影到 3D 点云，然后利用估计出来相机位姿，将此 3D 点云投到另一帧图像）。



2.3 Data Scanning

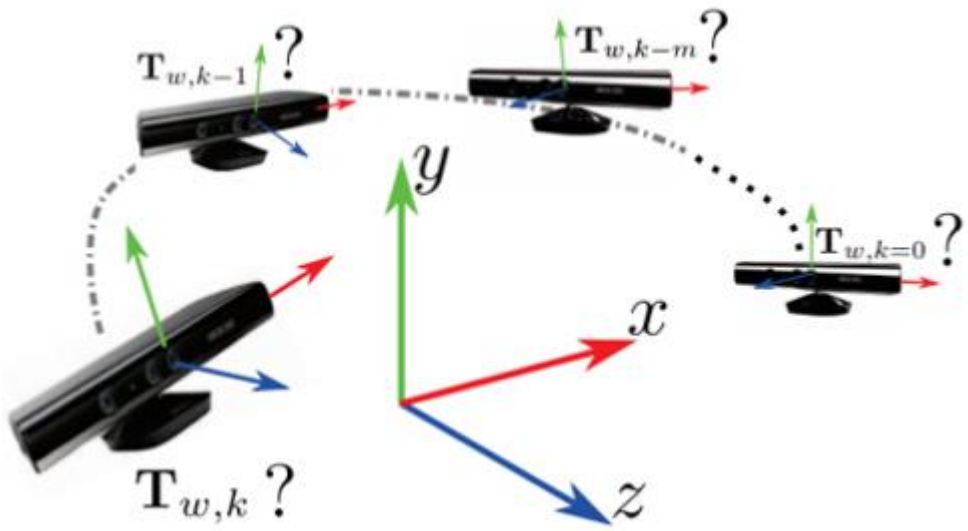
因为采用的是实测场景的序列图片作为输入，并且后期会用到相关的 RGBD 重建后的结果，因此此处介绍 RGBD 实时重建的相关研究。

如果输入的 RGBD 数据只有一帧，那么只需要把这一帧对应的点云模型作为重建的模型输出即可。不过通常的深度相机的帧率普遍较高，所带来的数据量是非常庞大的。以微软的 Kinect v1 为例，其 FPS=30，即 1 秒钟扫描 30 帧，也就是 1 秒钟便可得到 30 张 RGB 图像和 30 张深度图像。每一帧图像的分辨率通常是 640x480，那么在短短的 1 秒钟，深度相机得到的点云的点的个数是 640x480x30=9216000。如何在重建过程中处理如此庞大的数据就成了主要问题。另外深度相机所得到的深度数据是存在误差的，即使相机位置固定，现实场景中的点在不同帧中的深度值也会有区别 。也即是说，对于每一个现实中的点，在扫描过程中会得到众多“测量值”位置。那么，如何估计点的最终位置？这个问题可以被称为“从大数据中建立模型”问题。



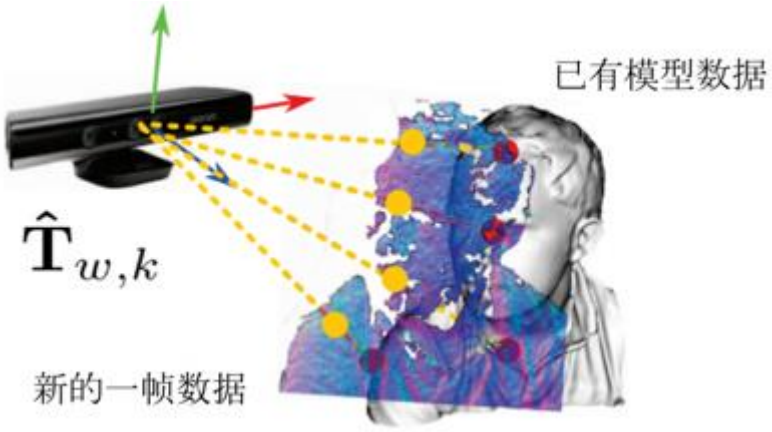
多角度 RGBD 数据的重建模型

除了上述问题外，重建过程中还有一个关键性问题——相机位置的估计。因为每一帧深度图像对应的点云模型是在相机的局部三维坐标系中。因此，不同的相机位置（即不同帧）便对应着不同的局部三维坐标系。然而重建后的模型需要坐落在一个坐标系，即世界坐标系或全局坐标系中。于是，我们需要找到每一帧的相机局部坐标系同世界坐标系的位置关系，也就是确定每一帧中相机在世界坐标系中的位置。



估计不同帧中的相机位置

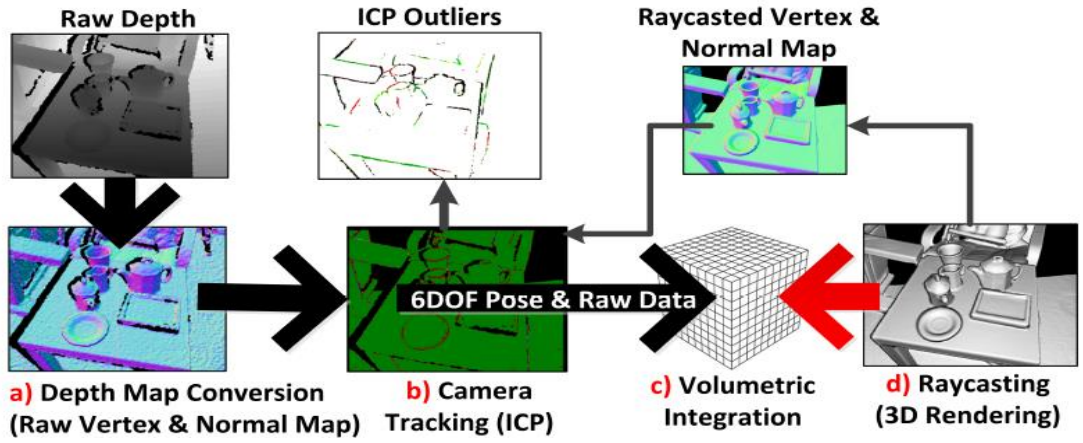
给定每一帧输入的 RGBD 数据，我们需要估计相机在世界坐标系中的位置。通常我们会把第一帧的相机位置作为世界坐标系的原点，于是我们需要估计的便是相机在此后每一帧相对于第一帧的位置的转移矩阵。使用数学语言描述是：在给定了第 k-1 帧重建的模型以及转移矩阵 Tw,k-1，还有第 k 帧的输入 RGBD 数据，估计出第 k 帧的转移矩阵 Tw,k。这里的 w 下标指代世界坐标系 world，k 是帧的编号，k>1。



估计新帧的转移矩阵

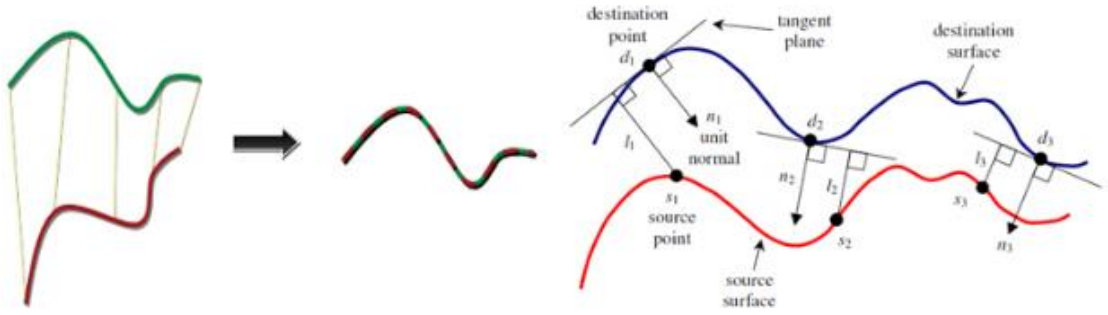
KinectFusion 是在这一领域标志性的成果，其开辟了 RGBD 实时三维重建的先河。其重建的主要流程如下图所示，首先(a)用设备读入的深度图像并转换为三维点云，根据相邻像素点计算每一点的法向量；（b）camera tracking 部分主要是用计算得到的带有法向量的点云和通过光线投影算法根据上一帧位姿从模型投影出来的点云，利用 frame-to-model 的 ICP 算法配准并计算相机位姿；

（c）根据计算得到的位姿，将当前帧的点云融合到 TSDF 网格模型中去， 其中 TSDF 模型将整个待重建的三维空间划分成网格，每个网格中存储了相应的数值；（d）根据当前帧相机位姿利用光线投影算法从模型投影得到当前帧视角下的点云，并且计算其法向量，用来对下一帧的输入图像配准。如此是个循环的过程，通过移动相机获取场景不同视角下的点云，重建完整的场景表面。



KinectFusion 主要流程图

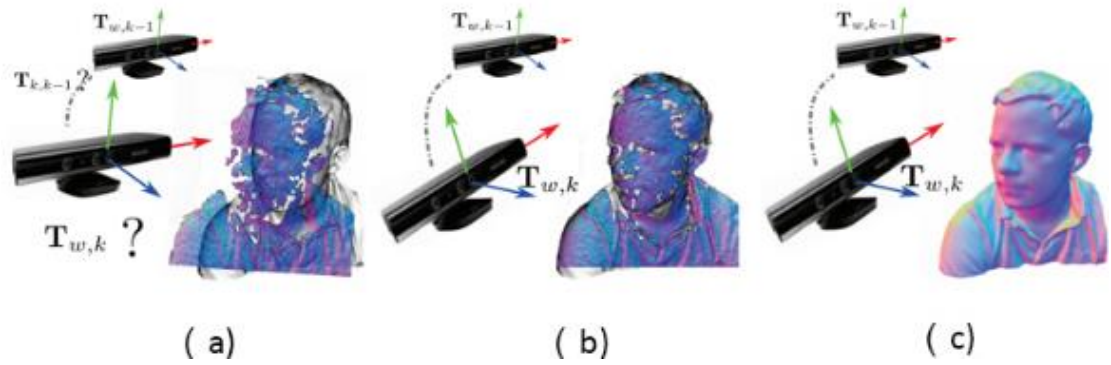
针对上面提到的问题，KinectFusion 使用了迭代最近点 ICP 的方法来解决。对于给定的输入原始数据（source）和目标数据（target），以及两者的数据点之间的对应关系（correspondence），ICP 计算得到原始数据和目标数据之间的转移矩阵，该矩阵使得所有的目标数据点到其对应的原始数据点所在的切平面的距离之和最小。图中的 si 和 di 是原始数据点和对应的目标数据点，ni 是 si 所在的切平面的法向量



ICP 实现效果示意图及相关参数含义

为了给 ICP 算法找到合适的对应点，KinectFusion 方法简单的将目标数据点——第 k 帧的数据点（图 5 中的黄色点）——通过转移矩阵 Tw,k-1 投影到原始数据点——第 k-1 帧的点（图 5 中的红色点），然后将两者作为对应相互对应的点。依照这种对应关系的 ICP 算法的最大优点是速度快，并且在扫描帧率较大，

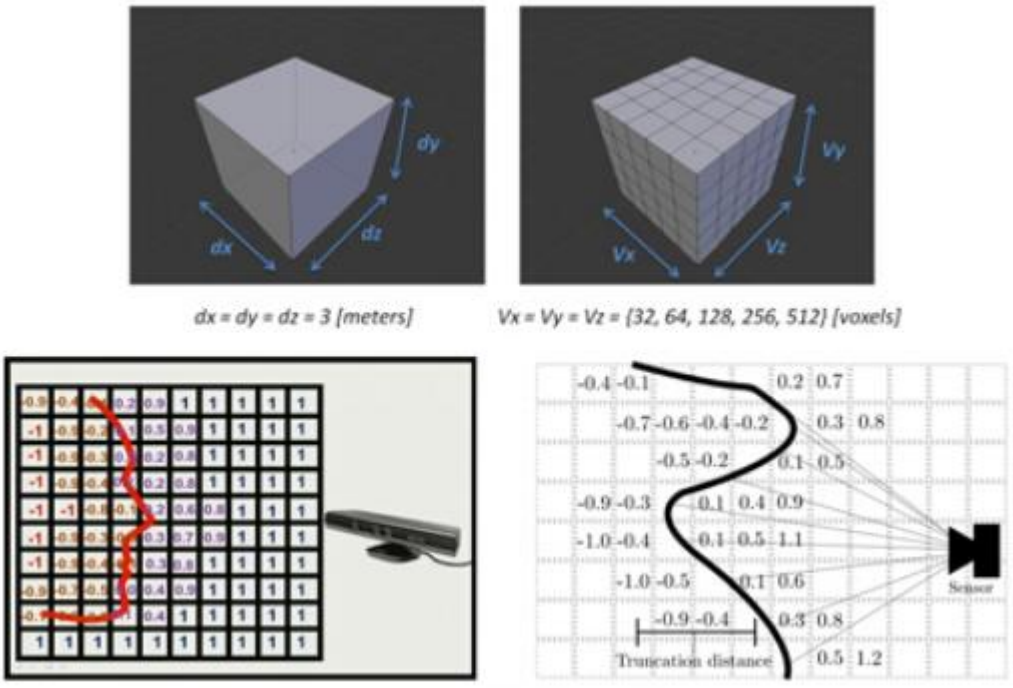
相邻两帧差别很小的情况下的精度很高。在估计了第 k 帧的转移矩阵后，将其作用到第 k 帧的在相机的局部坐标系的数据中，便可得到在全局坐标系中的数据。图 7 展示了典型的从输入数据（a），到估计相机位置并作用到数据上（b），然后到最终的优化之后的重建模型（c）的流程。



新的一帧数据的处理流程

在估计了相机位置后，我们需要把新一帧第 k 帧的数据同已有的第 k-1 帧的模型数据结合起来，以输出优化后的模型。即对于每个现实场景中的点，如何从该点的众多“测量值”位置中估计出最终位置的问题。

KinectFusion 在世界坐标系中定义了一个立方体，并把该立方体按照一定的分辨率切割成小立方体（voxel）。如图中定义了一个 3x3x3 米的立方体，并把立方体分为不同分辨率的小立方体网格。也就是说，这个大立方体限制了经过扫描重建的模型的体积。然后，KinectFusion 使用了一种称为“截断符号距离函数”（简称 TSDF）的方法来更新每个小网格中的一个数值，该数值代表了该网格到模型表面的最近距离，也称为 TSDF 值。对于每个网格，在每一帧都会更新并记录 TSDF 的值，然后再通过 TSDF 值还原出重建模型。例如通过图中两幅图中的网格的 TSDF 数值分布，我们可以很快还原出模型表面的形状和位置。这种方法通常被称为基于体数据的方法（Volumetric-based method）。该方法的核心思想是，通过不断更新并“融合”（fusion）TSDF 这种类型的测量值，我们能够越来越接近所需要的真实值。

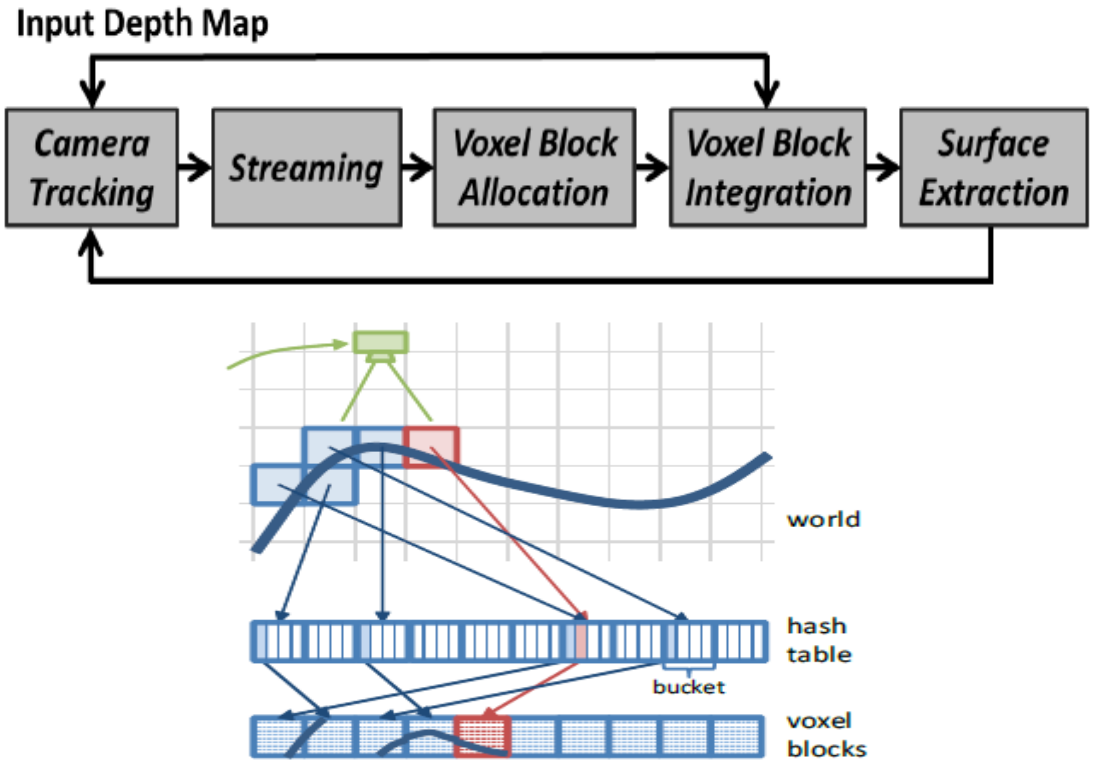


KinectFusion 立方体网格形式及 TSDF

KinectFusion 中 TSDF 的更新方法核心思想就是简单的对所有的测量值加权平均的过程。这种更新方式效率高，对于保证实时三维重建非常有必要。基于体数据的方法简单直观，而且容易使用并行计算实现，因此可以极大的增加扫描和重建效率。另外， 使用计算机图形学中的网格生成相关方法，我们可以很容易从这种体数据的结构中生成三角网格模型，这对于进一步的研究和渲染非常重要。不过，这种方法也有很大缺点。例如，KinectFusion 这种基于体数据的方法提前已经限定了扫描空间（例如上图的 3x3x3 米），超过这个空间的显示场景的物体将无法重建，这是因为定义立方体和网格需要的内存空间非常大 。这就意味着，KinectFusion 无法用来扫描大范围空间。另外，立方体中的所有的网格中的 TSDF 都需要记录，即便这个网格在现实场景中根本没有点，这就造成了极大的内存空间的浪费，并限制了扫描范围。针对这些问题，KinectFusion 之后的科研工作者们也提出了一些改进方法。

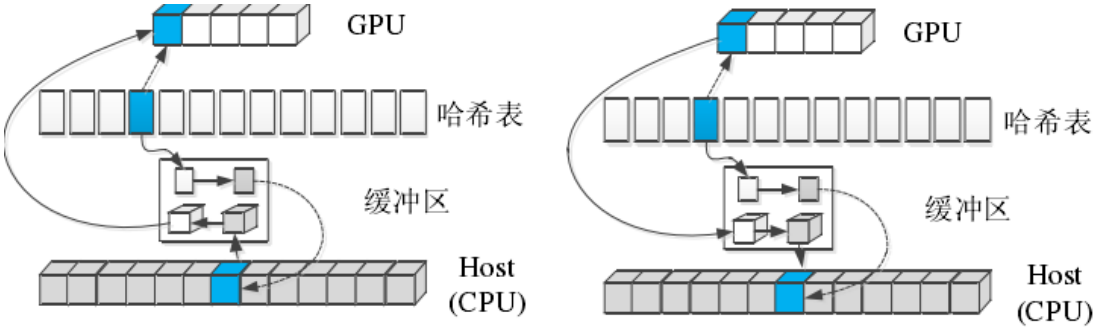
前面介绍的 Kinect Fusion 算法中使用的数据结构是固定大小的密度体积，这对于场景的 3D 重建有一定的限制。Voxelhashing 是 KinectFusion 的改进，主要思想是使用一个空间哈希方案来进行可伸缩的体素重建，基于一个简单的内存和速度高效的哈希表空间散列技术，主要是为了压缩空间，另外数据流可以高效的进出 Hash 表，允许传感器运动期间的进一步可伸缩性的扩展。像前面一样

也是利用 TSDF 模型对重建进行建模，只是在建模的时候，不是对整个空间都划分等大小的网格，只是在场景表面的周围划分网格。数据结构如下：用无穷多个的网格将空间划分成 voxel blocks，每个 voxel block 是等大小的，每个 voxel block 包含 512 个 voxels，每个 voxel 存储了 TSDF 值。其流程图如左上图所示，输入一个新的深度图，从融合开始。首先，分配新的 voxel blocks，并将块的描述插入到哈希表中；其次，扫描每一个分配的 voxel 来更新每一个 voxel 包含的 SDF，颜色和权重；另外，回收那些距离等值面太远和没有包含权重的 voxel blocks。这包括释放分配的内存和移动哈希表中的 voxel blocks 项。随着时间的推移，这些步骤的执行能够确保该数据结构仍然保持稀疏。整合后，从当前估计的相机位姿中投射隐式表面来提取等值面，包括相关的颜色。这个提取的深度和颜色缓冲区用来对相机位姿估计：考虑下一帧输入的深度图，通过点到面 ICP 算法进行投影来执行估计新的 6 自由度的相机位姿。



VoxelHashing 流程图及结构

并且根据哈希表能够快速查找、删除等特点，将重建过程中活动部分存储在显卡内存中，其余部分存储在主机中，当有需要的时候再交换到显卡内存中，这样就可以较为有效的解决显卡内存不足的问题。

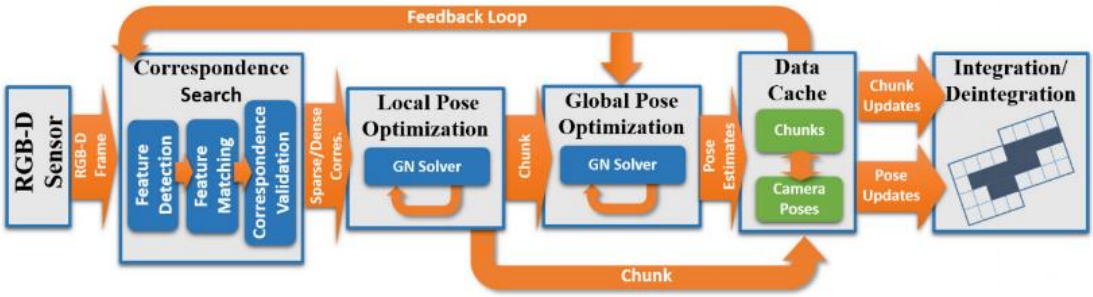


GPU 及 CPU 双向交流

但是对于 VoxelHashing 算法由于没有加入闭环检测及回环优化，在实时重建中，如果扫描返回到之前重建过的区域，同一个场景会被重建两次，而位姿轨迹通常是带有误差的，同一个场景两次重建的结果会错开，所以在扫描过程中容易造成累积误差，这样算法在回扫场景时的表现不够好，这对于实际场景的数据采集及重建过程带来了很大的影响，因此又有了相应的改进算法。

BundleFusion 是在 VoxelHashing 基础上的进一步改进，其算法的主要思路是首先利用 RGBD 相机获取场景数据流，并将当前数据帧和先前数据帧使用 sift 进行匹配，为了减少匹配时间带来的匹配误差，建立严格的筛选机制，如果帧间的匹配计算误差较大，则两帧之间所有的匹配都会被移除。在稠密局部配准上，将所有数据帧划分为各个等大小数据块，先在块内做位姿优化，由于块之间的交叠作用，可以将块内的位姿变换至同一个坐标系，块内的每两帧之间都做特征配准，同时将空间位置和描述子相似的特征进行融合，特征位置用所有特征位置的最小二乘拟合获得。在全局范围内，则建立块与块间的位姿优化，此处位姿也是通过块间交叠帧的计算获得。位姿优化包括稀疏特征优化项和稠密优化项，稀疏特征优化项是特征 3D 点的欧氏距离，而未采用以往方式中的 3D 到 2D 的重投影误差。稠密的优化项是匹配点的点到平面距离几何误差和匹配点的像素误差。优化时由于特征点位置是固定不变的，在做稠密匹配优化时，会被稀疏特征优化重置，因此只在最后扫描结束后的全局优化时，才包含稠密特征优化项。因为优化变量包括所有帧的位姿，变量个数较大，此处采用 Gauss-Newton 法极小化非线性目标函数。之后再将每个数据块的位姿映射到全局体素模型中，这里我们使用高效的哈希表形式存储的体素模型。映射过程中挑选出位姿改变量最大的数据块，按照将帧融合至体素模型时的位姿，做下采样融合，即将之前融合至体素模型中的数据从模型中减掉，再按照优化之后的位姿，将数据块内的数据重新融合

至体素模型中，模型融合的基础是在 VoxelHashing 框架基础上进行的。



BundleFusion 流程图

使用 BundleFusion 算法可以有效的对场景的回扫进行优化，结果有明显改善，但是其算法本身也不够完美。因为特征坐标在优化时是保持不变的，如果 SIFT 特征偏移几个像素，或者深度图像中特征位置处噪声比较大，这种误差会造成最小二乘求解的 SIFT 特征点的位置不对，从而优化的 pose 有误差。理想情况下，系统应该做 BA 优化，把特征的坐标也放在优化中，但是这样计算量会很大。实测 BundleFusion 在重复结构纹理的时候，效果不是特别好；建立的三维结构可能因为上述原因，特征的位置不对，使得重建的三维结构也可能会有偏差。

Paper reading:

3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation

Angela Dai[1] Matthias Nießner[2]

[1]Stanford University [2]Technical University of Munich

将 2D RGB 投影到 volume 中然后将 3D 几何信息结合，再输入到 3D 网络结构中。

要解决什么问题？现有方案有什么不足？如何解决？

The goal of our method is to predict a 3D semantic segmentation based on the input of commodity RGB-D scans.

In theory, one could simply add an additional color channel to the voxel grid in order to incorporate RGB information; however, the limited voxel resolution prevents encoding feature-rich image data.

We propose a novel network architecture that takes as input the 3D scene representation as well as the input of nearby views in order to predict a dense semantic label set on the voxel grid. Instead of mapping color data directly on the voxel grid, the core idea is to first extract 2D feature maps from 2D images using the full-resolution RGB input. These features are then downsampled through convolutions in the 2D domain, and the resulting 2D feature map is subsequently backprojected into 3D space. In 3D, we leverage a 3D convolutional network architecture to learn from both the backprojected 2D features as well as 3D geometric features

主要贡献是什么？

Our main contribution is the formulation of a joint, end-to-end convolutional neural network which learns to infer 3D semantics from both 3D geometry and 2D RGB input.

方法是什么？网络结构？

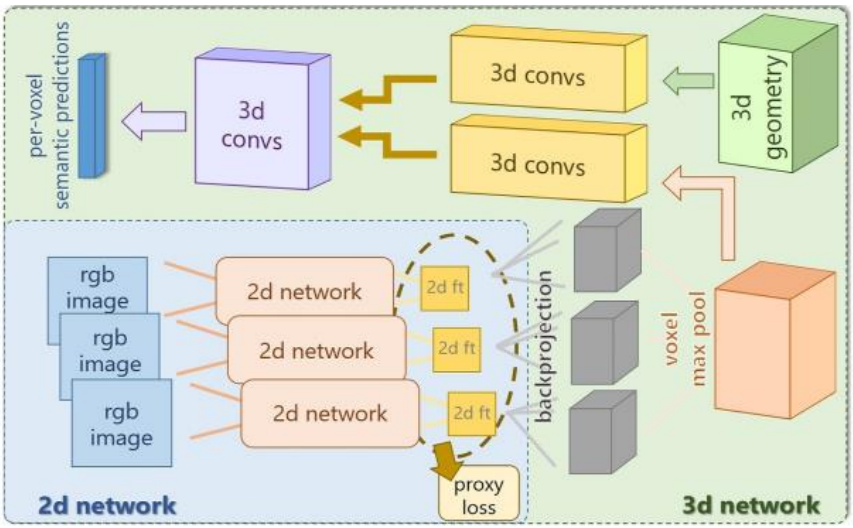


Fig. 2. Network overview: our architecture is composed of a 2D and a 3D part. The 2D side takes as input several aligned RGB images from which features are learned with a proxy loss. These are mapped to 3D space using a differentiable backprojection layer. Features from multiple views are max-pooled on a per-voxel basis and fed into a stream of 3D convolutions. At the same time, we input the 3D geometry into another 3D convolution stream. Then, both 3D streams are joined and the 3D per-voxel labels are predicted. The whole network is trained in an end-to-end fashion.

3D input： a volumetric grid representing the geometry of a 3D scan

2D input：the associated RGB images.

In order to combine the 2D and 3D features, we introduce a differentiable backprojection layer that maps 2D features onto the 3D grid. These projected features are then merged with the 3D geometric information through a 3D convolutional part of the network. In addition to the projection, we add a voxel pooling layer that enables handling a variable number of RGB views associated with a 3D chunk; the pooling is performed on a per-voxel basis. In order to run 3D semantic segmentation for entire scans, this network is run for each *xy*-location of a scene, taking as input the corresponding local chunks.

2D part：

The aim of the 2D part of the network is to extract features from each of the input RGB images. The final goal of the 2D network is to obtain the features in the last layer before the proxy loss per-pixel classification scores; these features maps are then backprojected into 3D to join with the 3D network, using a differentiable backprojection

layer.

数据集是什么？

The ScanNet dataset introduced a 3D semantic segmentation task on approx. 1.5k RGB-D scans and reconstructions obtained with a Structure Sensor. It provides ground truth annotations for training, validation, and testing directly on the 3D reconstructions; it also includes approx. 2.5 mio RGB-D frames whose 2D annotations are derived using rendered 3D-to-2D projections

效果如何？

	wall	floor	cab	bed	chair	sofa	table	door	wind	bkshf	pic	cntr	desk	curt	fridg	show	toil	sink	bath	other	avg
ScanNet [1]	70.1	90.3	49.8	62.4	69.3	75.7	**68.4**	48.9	20.1	64.6	3.4	32.1	36.8	7.0	66.4	46.8	69.9	39.4	74.3	19.5	50.8
ScanComplete [12]	87.2	96.9	44.5	65.7	75.1	72.1	63.8	13.6	16.9	70.5	10.4	31.4	40.9	49.8	38.7	46.8	72.2	47.4	85.1	26.9	52.8
PointNet++ [24]	**89.5**	**97.8**	39.8	69.7	**86.0**	68.3	59.6	27.5	23.7	84.3	0.0	**37.6**	66.7	48.7	54.7	**85.0**	84.8	62.8	86.1	30.7	60.2
3DMV (ours)	73.9	95.6	**69.9**	**80.7**	85.9	**75.8**	67.8	**86.6**	**61.2**	**88.1**	**55.8**	31.9	**73.2**	**82.4**	**74.8**	82.6	**88.3**	**72.8**	**94.7**	**58.5**	**75.0**

Table 2. Comparison of our final trained model (5 views, end-to-end) against other state-of-the-art methods on the ScanNet dataset [1]. We can see that our approach makes significant improvements, 22.2% over existing volumetric and approx. 14.8% over state-of-the-art PointNet++ architectures.

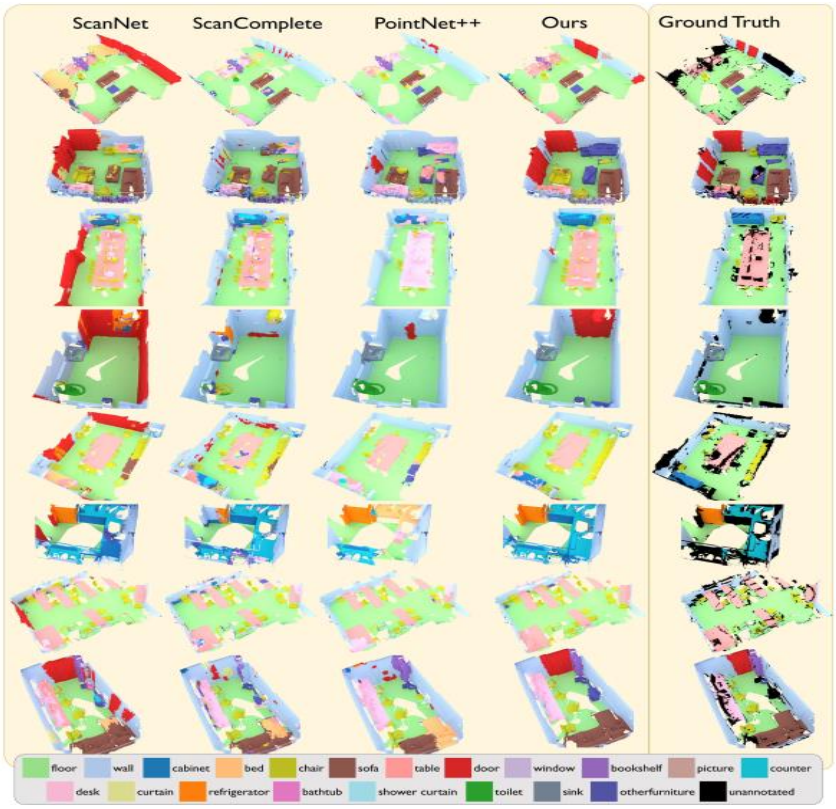


Fig. 3. Qualitative semantic segmentation results on the ScanNet [1] test set. We compare with the 3D-based approaches of ScanNet [1], ScanComplete [12], PointNet++ [24]. Note that the ground truth scenes contain some unannotated regions, denoted in black. Our joint 3D-multi-view approach achieves more accurate semantic predictions.

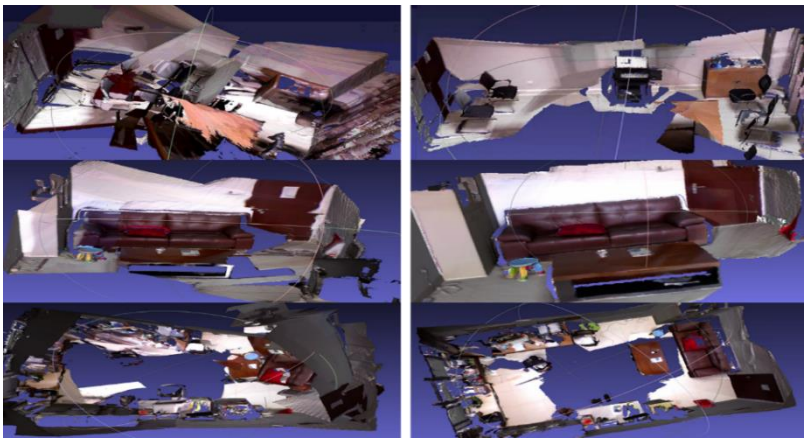
3 Experiments

实验目的：

对于多视角图片进行一致性的语义分割。

实验准备：

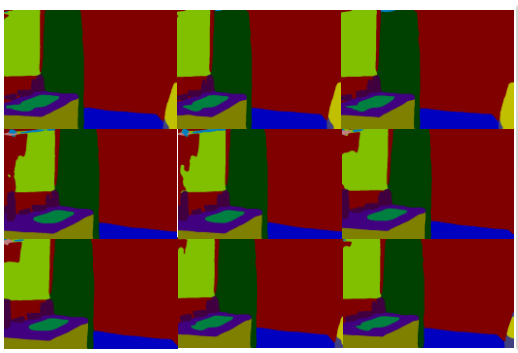
首先采用 Bundlefusion 采集实际场景的数据，其中包括采集的 RGB、depth、每一帧所对应的外参变换矩阵、生成的最终 mesh 结果。



实验步骤：

1)对于采集到的 2D 图片进行语义分割

采集得到一系列的序列图片后，将这些图片作为输入，使用的网络结构是上文提到的 RefineNet，得到相应的分割结果。并且在标准数据集上测试相应的分割结果。



nyu 数据集结果

实测场景数据结果

实验结果观察及分析

1)单帧的语义分割结果具有不错的效果

2)多帧的分割结果存在明显的跳变。

可能原因：

这是由于语义分割网络对单帧图片进行了处理，导致没有利用多帧图片时间和空间上的相关性联系。而单帧图片的语义分割结果受光照，视角的影响较为明显。

3)对于 3D 空间上的固定点在 2D 图像上的投影，在多帧上大部分具有一致的结果。

实验结果总结及拟解决方案：

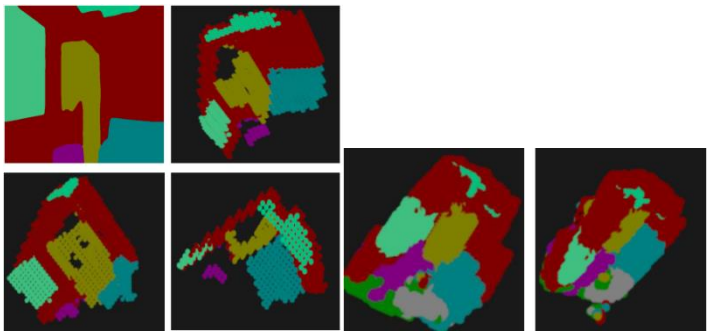
1)怎样解决多帧图片分割的跳变问题?

利用多视角图片的空间相关性，将多视角的结果投影到 3D 空间，做 voting，利用多视角图像在 3D 空间中的相关性，得到一个一致的分割结果。

2)基于外参矩阵和深度图将 2D 分割结果投影到 3D 空间。

得到相应的序列分割结果后，利用之前采集得到的 depth 图片以及相应的每一帧的外参矩阵，将 2D 上的分割结果以投票的方式投到立方体 voxel 中。考虑利用 3D 上的一致性来建立 2D 上的序列图像的一致性对应。

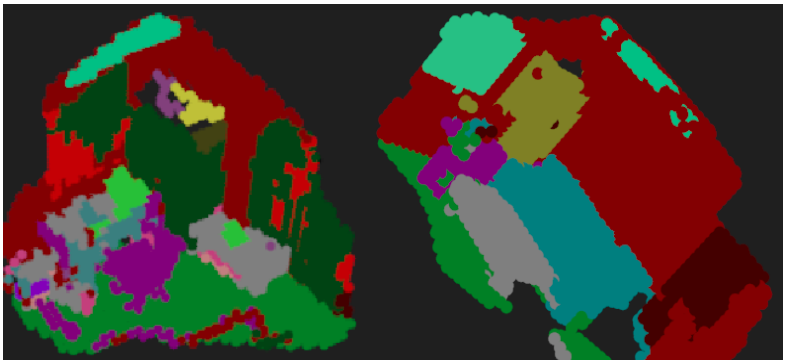
之前由于外参不对应的错误的投影结果：



单帧结果　　多帧结果

错误原因：程序中外参矩阵输出问题。

校正外参，外参对应后的 3D 结果，并在 Scannet 数据集上测试了相应的映射结果：



数据集结果　　实测数据结果

3)将 3D 点云的结果反投影回 2D 平面。

结果对比图：

实验结果观察和分析：

1） 2D 到 3D 的映射从结果上看能够较好的复原场景的 3D 分割结构。

2）3D 点云反投影回 2D 平面的结果存在严重的马赛克。

拟解决方案：1. 由于改变 voxel 分辨率会对结果造成影响，50*50*50 效果比 100*100*100 的结果差，因此考虑增大分辨率，但是问题是增大后会出现内存不够的问题，设法解决内存问题。2. 对得到的 2D 结果进行 refine，具体方案这块还有些模糊，有待讨论。