基于小块的对基于图像的材质贴图优化

**1 介绍**

为现实物体建模在各种应用场景下都是一个很重要的问题。集合重建已经是一个研究的重点，很多有力的算法被提出。依靠深度摄像头的大众化，普通用户现在可以使用如Kinect之类的设备构造几何模型。

然而，重建全面的现实物体同样需要高精度的表面纹理。基于图像的材质贴图是一种常见的手段，它使用一些视角的照片来重建物体的视角无关的材质贴图。然而这是一个有挑战的问题，因为几何模型和照片一般都有着噪声数据，因此，简单的将数据做映射然后叠加会得到重影和混合。

我们发现我们可以通过生成原图像的一个对齐图像来克服几乎所有的不正确贴图。我们的算法基于Zhou和Koltun的一个优化系统去矫正图像的误差和噪声。

受到最近基于小块的方法在图像和视频处理上取得的成功的启发，我们给出了一种目前最好的全局基于小块的优化系统来合成对齐后的图像。我们的能量函数结合了两个主要的对齐后图像的特征，1.包括尽可能多的原图像的信息，2.投影后保持一致。通过优化能量函数，我们逐步最大化对齐后图像和原图像的局部相似度，同时保证了所有对齐后图像做投影质量优秀。

我们的系统从原图像得到信息，通过小块。因此，这个方法很有弹性，能够接纳巨大的误差，甚至是几何模型缺了一部分，可以通过拼接来弥补。而这目前的方法都不能做到。最后，和Zhou和Koltun的方法作对比，我们在和几何模型复杂程度无关的比这两种方法取得了优势。

**2 算法**

大部分基于图像的材质贴图的目标是使用N张图片生成几何模型的视角无关的高质量贴图。这些方法一般给出了物体的大致几何形状和粗略的拍了N张照片的照相机的位置。一旦贴图完成，就可以从任何角度得到效果。

一个简单的方法去生成就是直接根据目前所给出的角度把原图像做映射到几何模型上然后把所有的图片叠加。然而，实际中因为一些误差，这么做会导致结果有重影和混合。

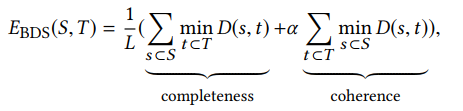
因为几何模型和照相机位置的误差，图片会没有对齐。因此，简单的叠加就会使最终材质有重影。

为了克服没有对齐的问题，我们给出了一种生成对齐后（目标）图像的方法，对于每张原图像。首先，我们通过将原图像内容拼接位移来纠正原图像没有对齐。之后，再做映射的时候由于图像已经对齐，得到的结果就会很好。下一节会详细说明这个做法。

**3.1 基于小块的能量函数**

我们主要的目的是生成高质量的材质贴图，因此目标图像需要满足两个主要特征，1.每个目标图像需要和原图像足够相似，2.最终合成的材质贴图和各张目标图像要足够相似。我们的目标是设计一个全局能量函数满足上述两个特征。

为了满足第一个特征我们要求每张目标图像包含足够的原图像的信息。为了满足，我们使用了Simakov算法中的BDS。这是一个基于小块的能量函数，定义如下：



其中a是一个参数，来控制两边的比例，s和t是原图像和目标图像的小块，D是s和t所有像素值在RGB域上的方差和。L是一个小块的像素数量。

这里，第一块指每块原图像的小块都和目标图像里某小块足够相似。第二块则是反过来。第一块的目的是原图像里面的信息需要尽可能在目标图像里体现，第二块的目的是目标图像的信息都是来自于原图像的，而不是其随意生成的。在我们的实现中，我们设置了a=2，认为第二块比第一块更重要。

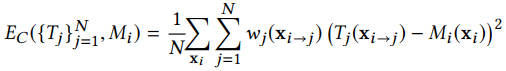
注意到上式定义了一对原图像和目标图像之间的函数，为了将其扩展到所有图像，我们扩展函数为：



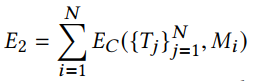
基于小块的方法比起局部变换来说要更加出色，因此能够处理更大的几何模型和照相机位置信息的误差。更加的，当局部变换隐式的导致了视觉形变，它从原图像到目标图像包含了所有信息，这不适合我们。如果几何模型没有包含某些特殊的特征，那些包含了这些特征的区域不应该被包括到贴图上。因此，这些方法得到了重影和混合。Waechter算法选择了某一个面作为结果避免了重影，然而这依然没有办法移除不正确的材质，因为它出现在了所有原图像里。注意到，集合模型有缺失几乎出现在了所有的不正确的几何模型里，这就是为什么目前的方法处理这个问题效果都很差。

尽管原图像和目标图像的相似度很重要，但是这部足够。除了要保证这点以外，我们还要保证目标图像投影生成的贴图和目标图像要足够相似。这个可以通过若干方法实现。

比如说，我们可以通过将现在的目标图像和其他所有和其相交的目标图像作比较。同样，我们也可以将目标图像和最终生成的材质贴图相比较。由于材质贴图是用所有目标图像生成的，所以这两种方法做优化没有明显差别。然而为了做算法上的优化，我们使用后一种策略。则我们的能量等式是：



其中第一个求和是图像i的所有像素点xi。权重wj代表了这个像素点对于它映射后的点的贡献大小。在实现中我们使用了，因为如果这个点相对集合模型角度越差；距离越远则其信息越不可信。我们扩展这个等式去满足所有的图像，有：



为了满足以上两个属性，我们给出了完整的主要函数，是两个函数加权得到的：



其中λ定义了一致性的权重，我们在实现中设置为了0.1。优化我们给出的基于小块的能量函数给出了目标图像，包括了源图像的几乎所有信息，视觉上连贯，同时做映射也不会出问题。当优化的目标图片生成，它们可以用多种方式被用来生成一个优秀的纹理贴图。比如，首先将所有目标图像映射到几何模型上。然后，几何模型上的点就有来自不同目标图像的颜色。最后的颜色可以使这些颜色的取平均。

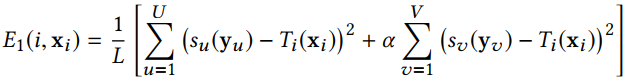
**3.2 最优化函数**

为了有效率的最优化我们的能量函数，我们给出了一种交替的优化策略，可以很好的解决给出目标图像和不同视角的纹理的问题。特殊的，我们最小化我们的能量函数，通过交替优化两组变量的方式。我们初始设置目标图像和最终纹理在某个视角的投影都为源图像。然后我们使用两步对齐和重建方法进行迭代直到结果满足要求。接下来我们来说明这两步方法：

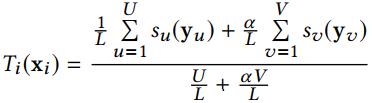
1. **对齐**

在这个阶段，我们固定最终纹理的投影图像M1, …, Mn然后最小化能量函数，找到最优的目标图像。这个阶段可以使用Simakov的查找和投票的迭代来完成。第一步，我们构造了一个小块搜索过程，类似于Simakov，用于找到一个小块的最小距离块。第二步，我们使用的投票过程来生成目标图片最小化了我们的能量函数。注意和原本的投票函数我们有关键的不同，因为我们新加的一致性约束Ec。

第一项：我们考查E1式子，可以得到：

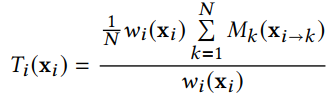


其中指某个特定的相机i和像素Xi。Su和Sv是原图像中覆盖了像素Xi的。Yu和Yv指Su和Sv中的单个像素。而U和V则指小块的数量，那些完整性和一致性的小块。注意到大部分变量是目前像素Xi的一个函数，但是我们忽略相关性，为了计算的简便。为了最小化Ti，我们对上式求导，可以得到：



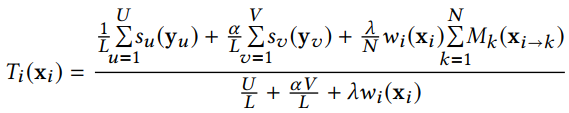
为了之后的方便，这里保留了1/L在分子和分母上。

第二项：第一项是标准投票过程，就如Simakov算法里一样。然后从源图像里提取出了信息。我们最关键的不同在于第二项，保证了一致性，通过保证目标图像和纹理足够相似。我们也可以求导得到第二项：



同样的，为了之后的方便保留了。在这里，每个目标图像被计算，通过其他映射到相同位置的其他视角的点的平均。这强制了对齐后的图像要和最终纹理尽可能相似。

组合：将上两个式子组合可以得到：

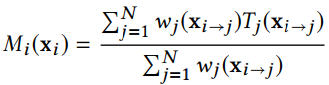


可以看到，最终更新的目标图像是普通投票和目前纹理的综合的平均。这说明了生成的目标图像的一致性有了保证。

能量函数的最小化会经过几次搜索和投票过程的迭代。迭代体现在每次更新过的目标图像会被用于下次迭代的搜索过程。我们通过经验发现只经过一次迭代就已经可以得到很好的效果了。

1. **重建**

在这个阶段，我们固定目标图像来生成最终纹理。因为纹理只在能量函数的第二阶段一致性阶段出现，我们可以简单的得到：



这是我们的材质生成等式，可以看到最优化的材质就是所有有投影上的目标图像的加权平均。如果目标图像没有很好的对齐，比如刚开始的一段时间，那么材质纹理会有重影的混合。接下来的迭代就会尝试降低不对齐的程度，导致材质纹理的效果变好。

我们迭代的重复对齐和重建过程直到满足要求。如果一般的基于小块的方法，我们在多维度进行过程避免陷入局部最小值和加速收敛。注意这个迭代指的是我们的大方法对齐和重建的迭代。在对齐方法中，我们也有搜索和投票的迭代方法。然而就如上面讨论的，我们发现只做一次迭代就已经有足够高的对齐效率了。

当算法收敛，我们的算法给出了目标图像，同时也给出了材质纹理的不同视角的贴图M1, …, Mn，看起来十分相似。因为我们的算法保证了一致性，全局的材质纹理可以通过对所有目标图像做到几何模型的投影并取平均来得到。

14307130167 蒋骐泽