

初步思路：3D mesh→ mesh decomposition为大量super-patch→ 两两merge形成hierarchical segmentation

1 super-patch

对应2D image的superpixel，3D mesh decomposition的算法有很多，但是能够得到源代码的工作几乎没有。不过，受到近年来比较受欢迎的SLIC superpixel算法[1]的启发，根据[5]的工作，我们提出一个利用K-means的super-patch算法。对于任意一个triangular mesh model，标准化后，我们将其看作一个graphical model, 用 $G(V,E)$ ：每个节点 V_i 代表一个face，相互连接的节点表示相互相邻的面，每一条Edge上定义 $Distance(V_i, V_j)$ 为两个face的“距离”：

$$Distance(V_i, V_j) = a \cdot (1 - \cos^2(\alpha)) + b \cdot Phy_Dist(V_i, V_j)$$

其中 α 是两个面之间的dihedral angle， $Phy_Dist(V_i, V_j)$ 是两个面的重心到相邻edge的中点的距离之和。权重 a, b 保证了这个距离在 $[0,1]$ 之间。具体的选取由下段描述的training决定。

1.1 Training

受Berkeley的segmentation dataset[3]的影响，2009年Princeton发布了3D segmentation的benchmark[2]，400个model中每个模型都由若干位志愿者做出了分割，做为ground truth。任取200个模型作为training set，剩下的其中100个模型作为test set，另外100个模型作为validation set，把training set中的每一对相邻的face提取出来，计算 $((1 - \cos^2(\alpha)), Phy_Dist(V_i, V_j))$ ，定义 $Distance_{groundtruth}$ 为 V_i, V_j 同属不同segment中的概率（在每个模型中13个ground truth中label不同的概率）。接下来就可以通过一个简单的logistic regression 来将 a, b 确定。

1.2 K-means clustering

当distance被定义好后，对于任意模型，每一对相邻的 (V_i, V_j) 之间的distance就可以算出来了。再定义任意两个不相邻的face 之间的距离为

$$Distance(V_i, V_j) = \min_{V_3 \neq V_1, V_2} (Distance(V_1, V_3) + Distance(V_3, V_2))$$

在计算的时候可以运用寻找最短路径的Shortest Path Faster Algorithm(SPFA)算法，接下来就可以开始做clustering了。由于我们定义的距离函数很简单，而且对初始的over-segmentation的精度没有特别严格的要求（仅仅是想让每个3d 模型中的patch个数相同），所以我们取k为一个比较大的值。（k=2000）

- 1: Initialize Cluster centers C_k by randomly choosing k faces
- 2: set label $l(i) = -1$ for each face i
- 3: set distance $d(i) = \infty$ for each face i
- 4: set residual error $E = \infty$
- 5: **while** E won't change **do**
- 6: **for** Each cluster center C_k **do**
- 7: **for** each face i **do**
- 8: compute $D = Distance(C_k, i)$
- 9: **if** $D < d(i)$ **then**
- 10: set $d(i) = D$
- 11: set $l(i) = k$
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: Compute new cluster centers(move to the closest face centers)
- 16: Compute residual error E (distance between previous centers and recomputed centers)
- 17: **end while**

super patch的代码（包括training）在3d.cascaded_seg/super_patch中。

2 Super-Patch Merging

2.1 Boundary Recall Measurement

受[1]中二维图片measurement的启发，我们定义三维mesh的boundary recall如下：

$$BR_G(S) = \frac{\sum_{p \in \partial G} \mathbb{1}(\min_{q \in \partial S} \text{distance}(p, q) < \epsilon)}{\partial G} \quad (1)$$

其中G为ground truth segmentation, S 为machine segmentation, ∂ 表示分割的边缘(boundary)。对于3D mesh，定义两两vertex之间的edge作为boundary，这样一个segmentation的边缘就由很多线段组成。将每条线段类比为2D中的pixel，2D中两个pixel之间的距离可以直接计算，而计算两个线段的距离的时候，我们利用[6]中的方法（helper/DistBetween2LineSegment_mex.mex）

这个measurement数值越大说明真实的边缘检测出来的比例越多。

★我写的计算这个measurement的代码复杂度是 $O(\partial G * \partial S)$ ，因此计算速度极慢，不知道能不能写一个更快的算法，因为速度原因在cascade training的时候也收到了很大的影响。

2.2 other measurements

其他的measurement包括rand index(有代码), Cut Discrepancy, Hamming Distance等等。其中rand index是很多3D segmentation paper中都使用的度量。这个度量的计算写出的代码运行可以很快(randindex.cpp)。

2.3 Cascaded algorithm for super-patch agglomeration

这部分的思路主要借鉴[4]。将每个super-patch对应于文中的super-pixel。对于一个3D Model I_i ，我们将其分为 k_i 个super-patch $R_i = \{R_{i,1}, \dots, R_{i,k_i}\}$ ，再令 $N(R_i)$ 为所有的相邻区域。每一个区域对 $(p, q) \in N(R_i)$ 都由一个表示特征的向量 $\phi_{p,q}(I_i, R_i)$ 组成。具体使用的特征在section 2.3.1中会介绍。

我们同时也有ground truth的数据，它们将作为训练样本。这些真实的分割数据是以带标签的3D模型表示的，由于我们不假定任何类别信息，所以我们仅仅需要得到的信息是：任意一对像素是否属于同一区域。

利用模型 I 的真实分割的数据，我们可以给 $N(R_i)$ 中的区域对加上标签：任给一个真实分割图 G_i ，我们将 $R_{i,p}$ 标定为 G_i 中与它重合最大的区域的标签。接下来，如果 $R_{i,p}$ 与 $R_{i,q}$ 的标签是相同的，则定义 $y_{pq}^i = 1$ ，否则 $y_{pq}^i = 0$ 。如果对于一张图有很多的真实分割，则定义 y_{pq}^i 为针对所有真实分割定义标签的平均。这样 y_{pq}^i 的值就在0和1之间，度量了人们认为 $R_{i,p}$ 和 $R_{i,q}$ 相同的概率，同时也反映了区域之间边缘的强度。

将训练样本 I_1, \dots, I_N 中区域对的特征和他们的标签放在一起，我们得到 $\{< \phi_i, y_i >\}$ ，这样就是一个预测问题：给定一个区域 I 和初始分割 R ，估计相邻区域聚合的条件后验概率为 $Pg(p, q; I, R) \triangleq P(y_{pq} = 1 | \phi_{pq}(I, R))$ 。如果 $Pg(p, q; I, R) > \frac{1}{2}$ ，则我们必须聚合 R_p 和 R_q ，并且将它们之间的边缘去掉，否则我们将保留边缘。

2.3.1 features between neighboring superpixels

这里所说的features都是定义在相邻两个patch $R_{i,q}$ 上的。

- Bounding Box

先计算每个patch i 的bounding box b_i ，在 $R_{i,q}$ 中，定义 $b_i \cap b_q$ 为两个box的相交部分 c_{iq} ，取 $\max(\frac{\text{volume}(b_i)}{\text{volume}(c_{iq})}, \frac{\text{volume}(b_q)}{\text{volume}(c_{iq})})$

- Average Norm Vector

附录

整个project均在F盘上进行操作。

A Mesh Segmentation Data

- Princeton's segmentation dataset 存放在F:/MeshsegBenchmark-1.0 中
- data/{train, test, val}.txt 分别为training, test, validation set
- data/off 中存放了所有3维模型，共380个，每10个为一个类别（人，椅子，杯子，等等。。）
- data/seg 中存放了各个算法的分割结果，其中Benchmark为ground truth, super_patch 中存放了super-patch 算法的结果，每个模型2000个分割。

这些是所有需要用到的东西。。

B codes

代码保存在F:/github/3d_cascaded_seg 其中的readme文件介绍了各个目录的组织情况，全部代码在这里托管https://github.com/luvegood/3d_cascaded_seg 请clone这里的代码，台式机里的代码不一定是最新的。

References

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *École Polytechnique Fédérale de Laussanne (EPFL), Tech. Rep*, 149300, 2010.
- [2] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

- [3] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [4] Zhile Ren and Gregory Shakhnarovich. Image segmentation by cascaded region agglomeration. June 2013.
- [5] Shymon Shlafman, Ayellet Tal, and Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer Graphics Forum*, volume 21, pages 219–228. Wiley Online Library, 2003.
- [6] Dan Sunday. Distance between 3d lines and segments. http://geomalgorithms.com/a07-_distance.html.