**周报**

2018.09.24—2018.09.30

修改Pipeline：

1. 输入一个轮廓Polygon。
2. 任选四点作为依据进行划分，划分后拼接的方式为，以1-3,2-4为两条轴进行顺序摆放，加入旋转和翻转等操作。如图1,2。
3. 对划分情况进行筛选，需排除掉的情况有：1234四个点相距过近，以及拼接后出现模板相交的现象。提取所有围成的形状。
4. 将获取的形状轮廓，与dataset中的图案进行对比，选出合适的候选图案，对候选图案进行变形，最后评估变形结果是否满足要求。
5. 将密铺结果映射到三维模型上。

这周的工作主要有：

1. 完成中文论文初稿定稿，但还缺少不少实验结果图。
2. 完成英文论文的翻译。
3. 实现形状匹配阶段的代码。
4. 参与实验室的各种活动

下周工作：

1. 对中文论文文字叙述部分，如介绍，相关工作和问题描述进行修改。介绍部分加上对比图，加上贡献点。相关工作分条加小标题，简化。问题描述整理清晰，加图，详细解释采用四边形密铺的理由。算法描述删去不重要的段落，着重描述形状匹配和形状变形。
2. 在数据集中加入一批新的候选图，新增可带孔洞的艺术图案，在提取轮廓时仍只保留外轮廓。
3. 实现形状变形的算法。
4. 在拼铺方式中加入旋转和翻折，尽快实现相关代码。

在翻译的过程中，就感觉自己很多地方有点想当然，导致叙述的不清晰。改了之后问题还是很大，尤其是写文章着重点把握不好，重点地方轻描淡写，不重要的地方又臭又长，还需要多下功夫。没有图还是很难讲清楚一件事，需要多下功夫在作图上。

2018.09.17—2018.09.23

Pipeline：

1. 输入一个二值图，求得其轮廓。
2. 从候选点的点集（约为10到20个点）中选择合适的四点作为依据进行划分。该候选点集由轮廓上的高曲率点以及靠近轮廓的骨架端点的中点组成，划分后拼接的方式为，以1-3,2-4为两条轴顺序摆放。如图1,2。
3. 对所有可能的划分情况进行筛选，需排除掉的情况有：1234四个点相距过近，以及拼接后出现交叉重叠的现象，如图3。
4. 得到合适的拼接方案后，获取其围成的形状轮廓，并与dataset中的图案进行对比，**计算误差最小的图案及误差最小时的角度。**
5. **对图案进行轮廓的变形。**

这周的工作主要有：

1.完成第4步中匹配误差的度量函数。因为使用Hu矩计算的是图案的整体性质，虽然能对图案的相似性做初步的判断，但是无法获得更详细的信息，如两个图案处于什么角度什么位置时最为相似，即匹配误差值最小。基于以上考虑，对之前的匹配算法做了修改，同时计算了曲率误差及对齐后的坐标误差，使其能算出两个图案旋转多少度时匹配误差最小。

2.在之前完成的中文论文基础上修改完成了新的中文论文，因为从上一次的工作到这一次的，中间改动的地方还是比较多的。截止到24号，结果分析还没有写，中文论文总字数6400，英文翻译完成了三分之一，约2000词。

下周工作：

1. 完成英文论文的翻译。之前找了同学，但是看过论文之后说很多地方看的不是很懂，我比较担心同学翻译出现偏差，所以最后商量的是我翻译完之后帮忙修改润色，所以我计划先完成翻译再实现全部代码。
2. 完成项目代码。

工作感想：

1. 在暑假前确定了新的算法思路，当时以为整体改动不会很多，暑假就过得放松，出去旅游了10来天，回来后重新梳理算法思路，发现需要改动的地方还是有点多。赶24号ddl的时间有点紧张，心里也有了准备，但日常工作中还是会隔三差五想偷个懒，没有特别严格的要求自己，监督自己，现在因为这些原因又一次错过了ddl，自己心里非常自责，也比较惶恐，同时也清楚还没有达到标准研三学生的水平，所以还是希望能再提高一下自控能力，对自己再严格点，尽快结束当前项目。
2. 另外自己这次也是第一次写英文论文，还是要比写英语作文要吃力不少，因为首要目标是将事情讲清楚，所以有时候不能太自由发挥，写的过于随意，而且自己对翻译的工作也有些低估了，一字一词扣下来其实也是比较累的，总体来说还是能力有欠缺，也希望自己能引以为戒，多努力练习提升能力。

2018.09.10—2018.09.16

Pipeline：

1. 输入一个二值图，求得其轮廓。
2. 从候选点的点集（约为10到20个点）中选择合适的四点作为依据进行划分。该候选点集由轮廓上的高曲率点以及靠近轮廓的骨架端点的中点组成，划分后拼接的方式为，以1-3,2-4为两条轴顺序摆放。如图1,2。
3. 对所有可能的划分情况进行筛选，需排除掉的情况有：1234四个点相距过近，以及拼接后出现交叉重叠的现象，如图3。
4. **得到合适的拼接方案后，获取其围成的形状轮廓，并与dataset中的图案进行对比，计算误差最小的图案及误差最小时的角度。**
5. **对图案进行轮廓的变形。**

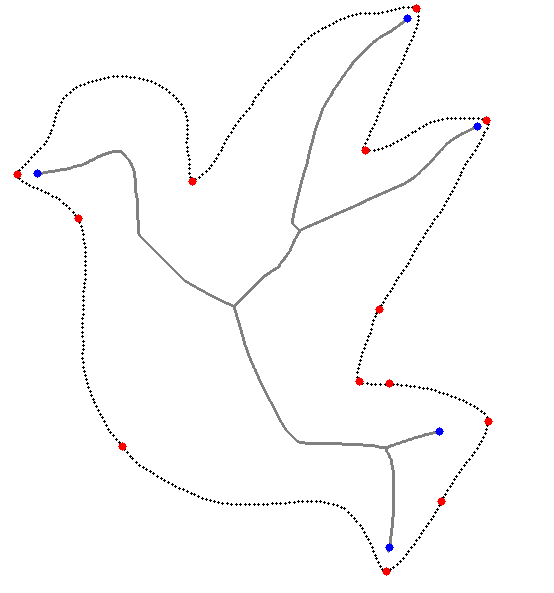
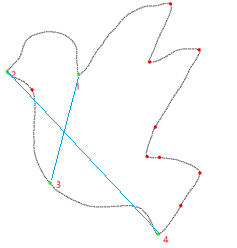
 

图1 图2

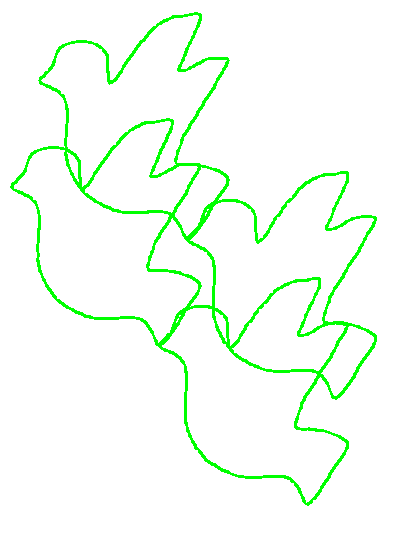
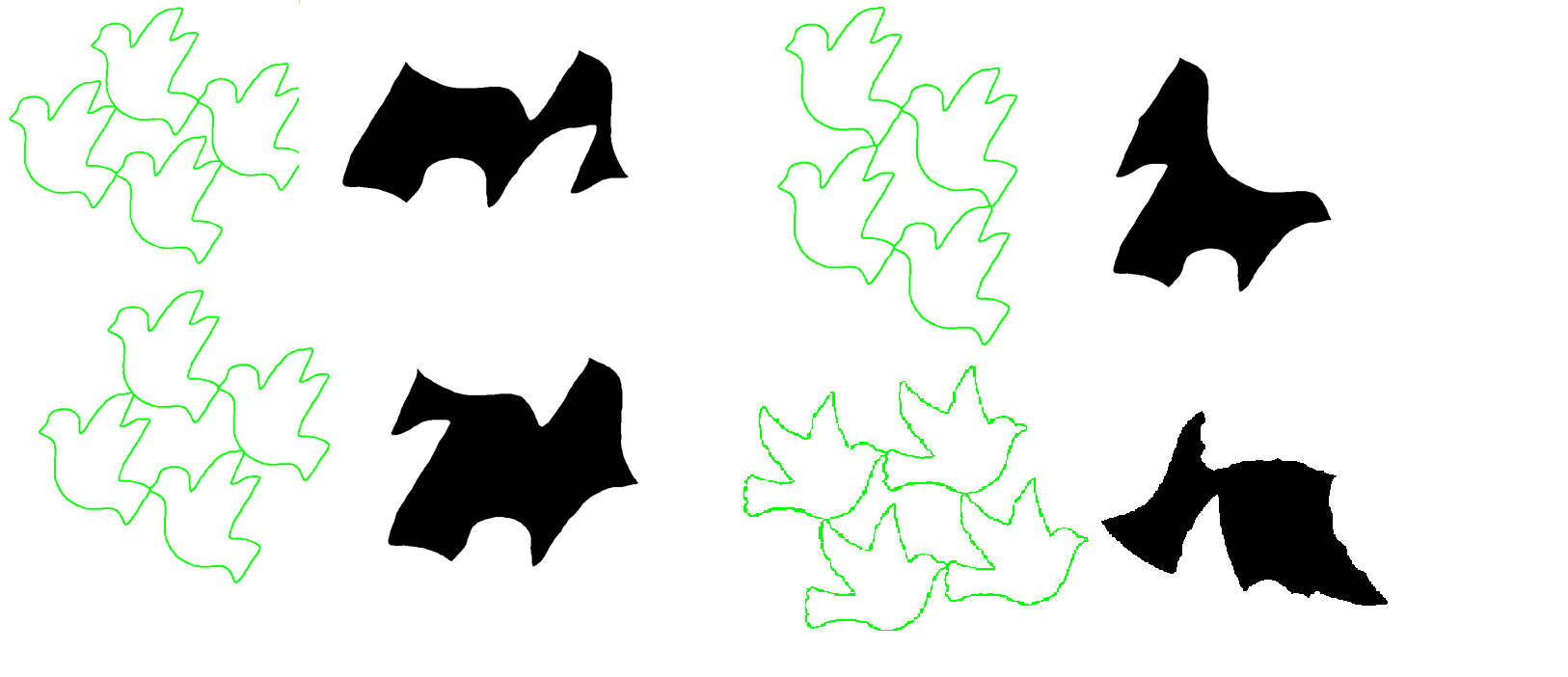


图3

这周的工作主要有：

一． 获取输入图形围成的形状轮廓，如图4，并与dataset中的图案进行对比，计算最相似的图案。德聪发来了他的代码，但不是原来提到的ICP的方法，是他最新实现的利于turning function （如图5）来进行轮廓的比较。但是他的代码有bug，复杂的形状直接会出错，而且他计算distance的部分代码实在没看明白思路。

图4

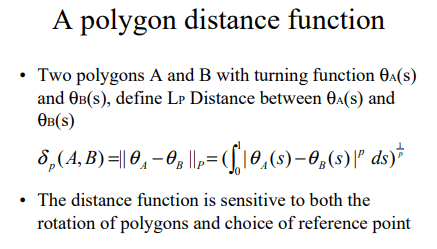
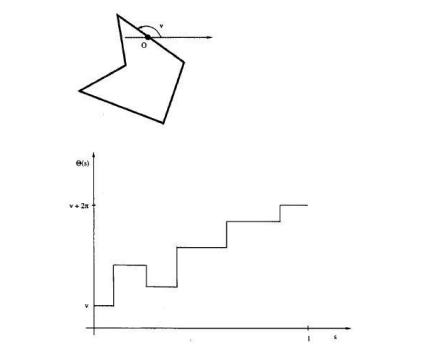
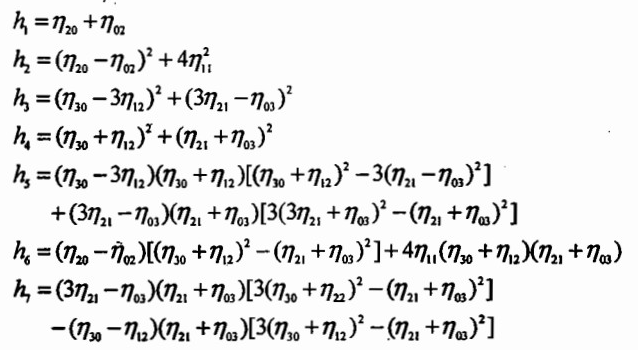


图5

二．最后我选择opencv里的MatchShapes函数来进行轮廓的比对。MatchShapes函数利用了图像的Hu矩。Hu矩是归一化中心距的线性组合.之所以这样做是为了能够获取代表图像某个特性的矩函数,这些矩函数对于某些变化如缩放,旋转和镜像映射(除了h1)具有不变性.Hu矩是从中心矩中计算得到,其计算公式如下所示。由Hu矩组成的特征量对于物体的形状描述得比较好，速度也很快，缺点是对纹理特征较为复杂的图像识别率偏低，因此也能比较好的满足需求。



三．论文的修改完成了一半，但是因为卡在了第4步最后，所以现在还是没有足够的结果。已经修改的论文整理一下会交给周宇学姐帮忙翻译处理。

下周工作计划：

1.完成剩下的工作。

2. 完成中文论文的修改，并尽快完成英文翻译及初稿撰写。

**周报**

2018.09.03—2018.09.09

Pipeline：

1. 输入一个二值图，求得其轮廓。
2. 从候选点的点集（约为10到20个点）中选择合适的四点作为依据进行划分。该候选点集由轮廓上的高曲率点以及靠近轮廓的骨架端点的中点组成，划分后拼接的方式为，以1-3,2-4为两条轴顺序摆放。如图1,2。
3. 对所有可能的划分情况进行筛选，需排除掉的情况有：1234四个点相距过近，以及拼接后出现交叉重叠的现象，如图3。
4. 得到合适的拼接方案后，获取其围成的形状轮廓，并与dataset中的图案进行对比，计算误差最小的图案及误差最小时的角度。
5. 对图案进行轮廓的变形。

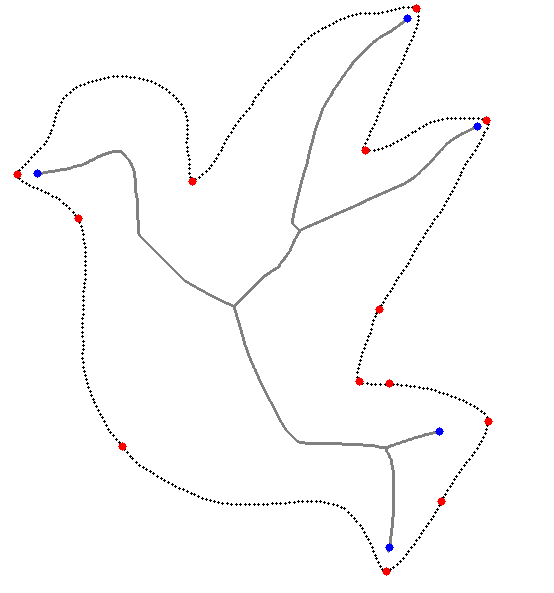
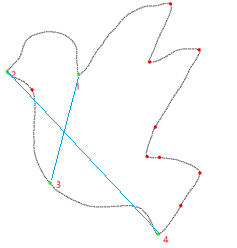
 

图1 图2

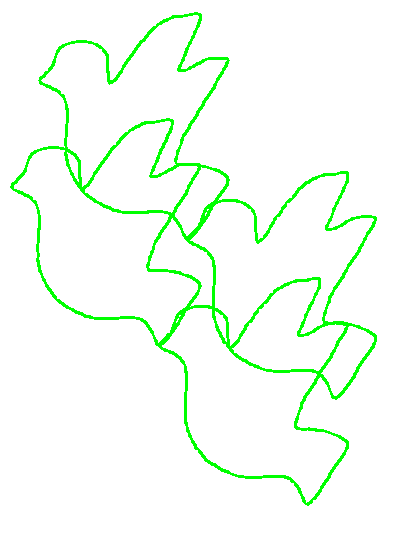


图3

这周的工作主要有：

一． 确定候选点，求得划分方案。在确定候选点的时候，我采用了骨架加轮廓上高曲率点的标准，如图，蓝色点是提取骨架的结果，红色点是根据轮廓曲率得到的高曲率的点。并在最终的点集中加入了骨架端点的中点，以获得更多的划分情况。

二．修改了中文论文中问题描述及定义这一部分。一开始设想的是，首先完成中文论文，然后使用google进行整体的翻译，并在此基础上进行修改。

下周工作计划：

1. 与德聪讨论，完成对比筛选以及求误差最小角度的工作。

2. 完成中文论文的修改，与周宇学姐一起完成英文翻译的修改及初稿的撰写。

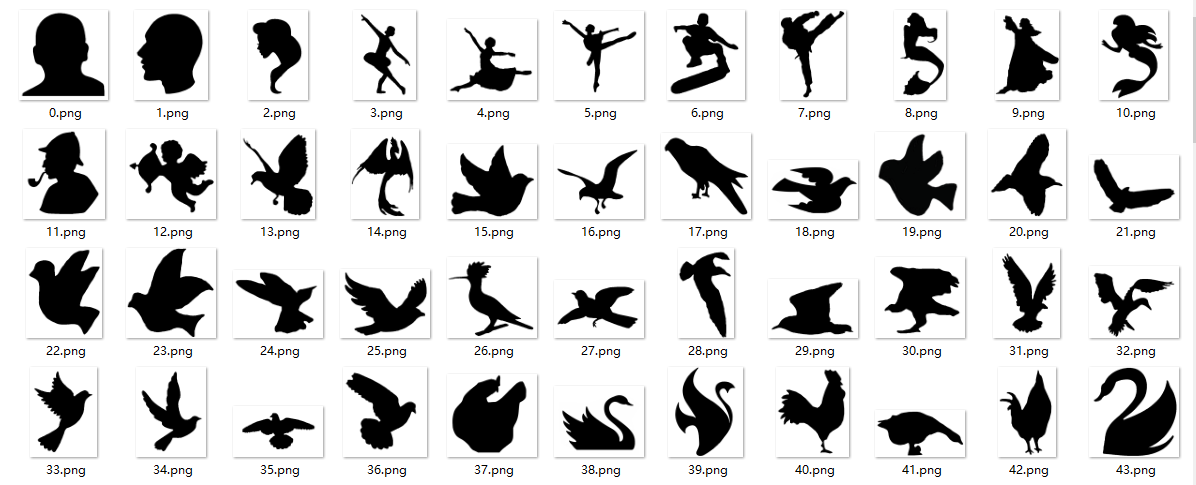
3. 争取多做几组结果，放到打印机上打印。

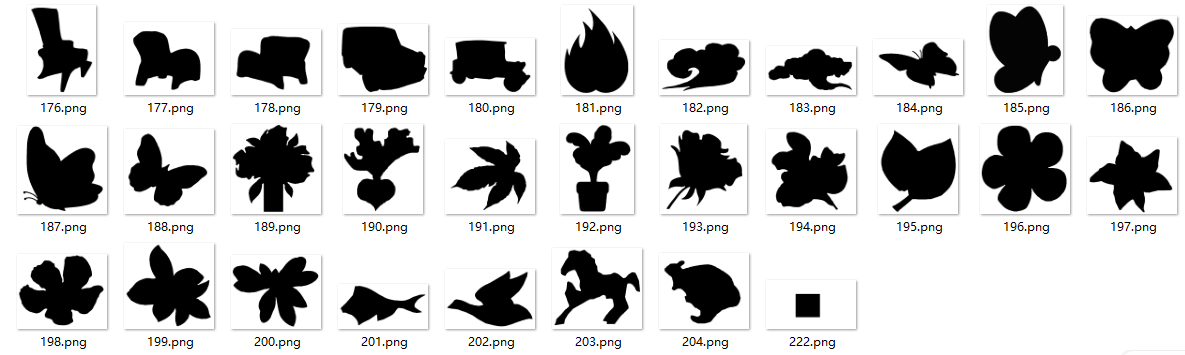
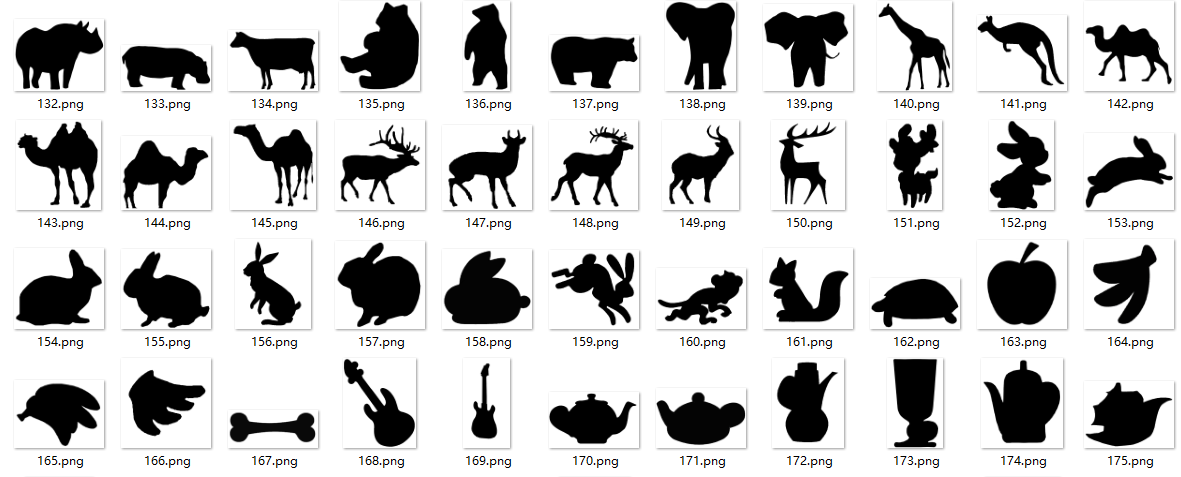
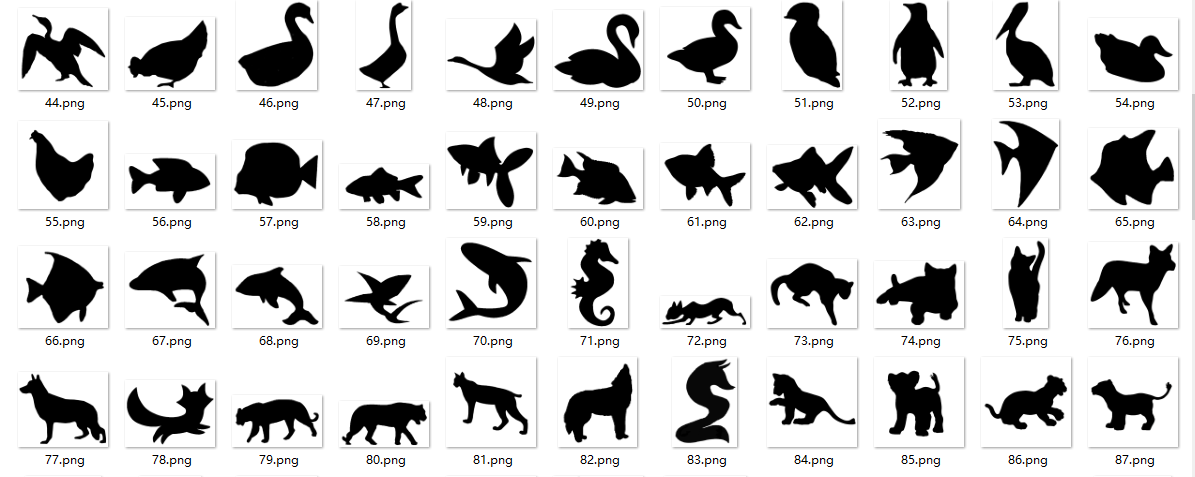
**周报**

2018.08.27—2018.09.02

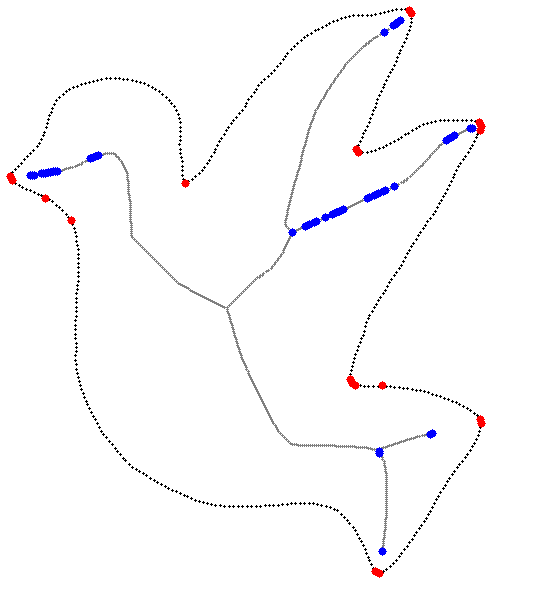
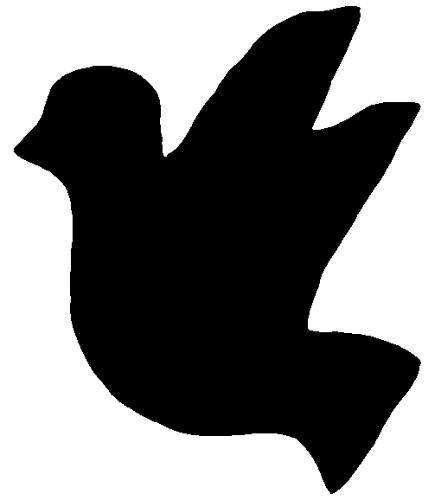
这周的工作主要有两项：

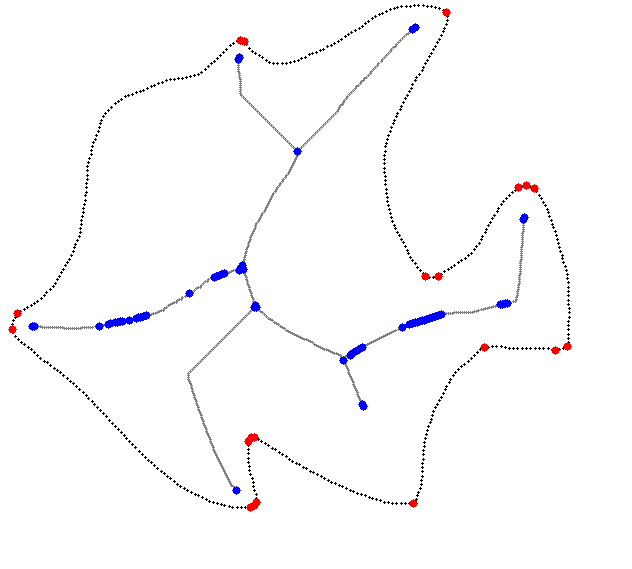
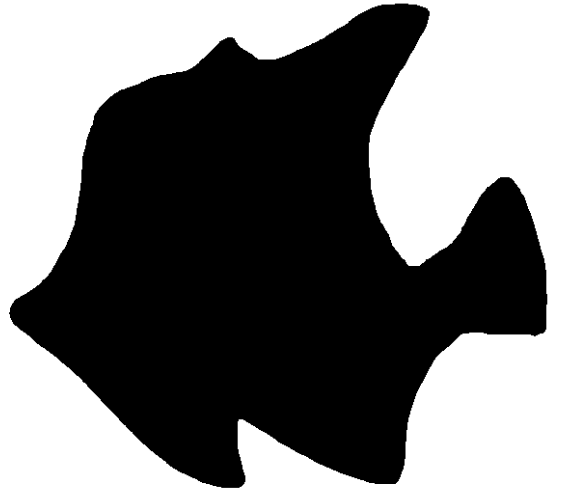
一．继续扩充数据集，去除了一些过于简单的图形，加入了一些更为复杂的图案。





二．继续实现划分算法。在最初设计划分算法的时候，还同时考虑了两个对应片段的凹凸性，即一个图案其中一边为凸时，其对应的另一图案上的另一边应为凹，这样的考虑带来了不少复杂度和难度。在确定先做一个图案的轮廓划分，再按照划分将该图案排列拼接，然后将围成的空隙与数据集里的图案比较的新思路之后，我觉得划分算法可以更简单直接。从候选的划分点中任找四个点，然后按这四个点的划分进行实验，所以如果一开始候选点中如果能包含groundtruth的划分点，那就很容易实现一个较好的结果。在确定候选点的时候，我采用了骨架加轮廓上高曲率点的标准，如图，蓝色点是提取骨架的结果，红色点是根据轮廓曲率得到的高曲率的点。离蓝色点中的端点距离最近的红色点，将成为优先考虑的划分点，这周的工作就做到这一步。

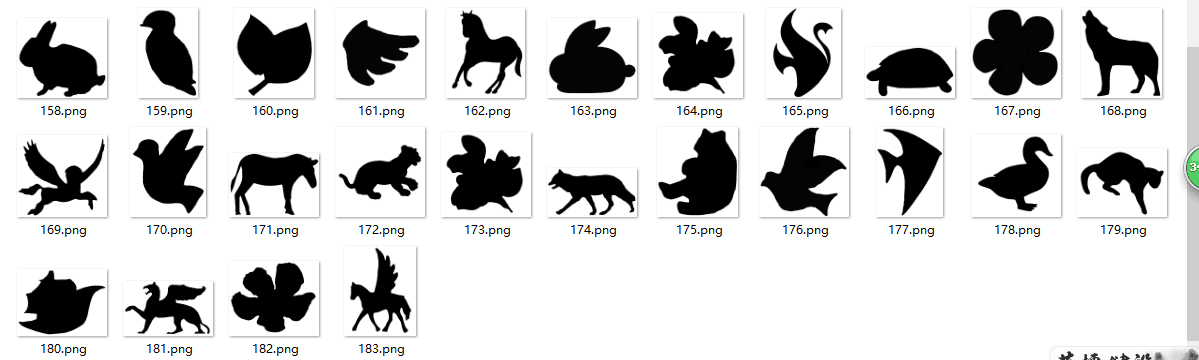
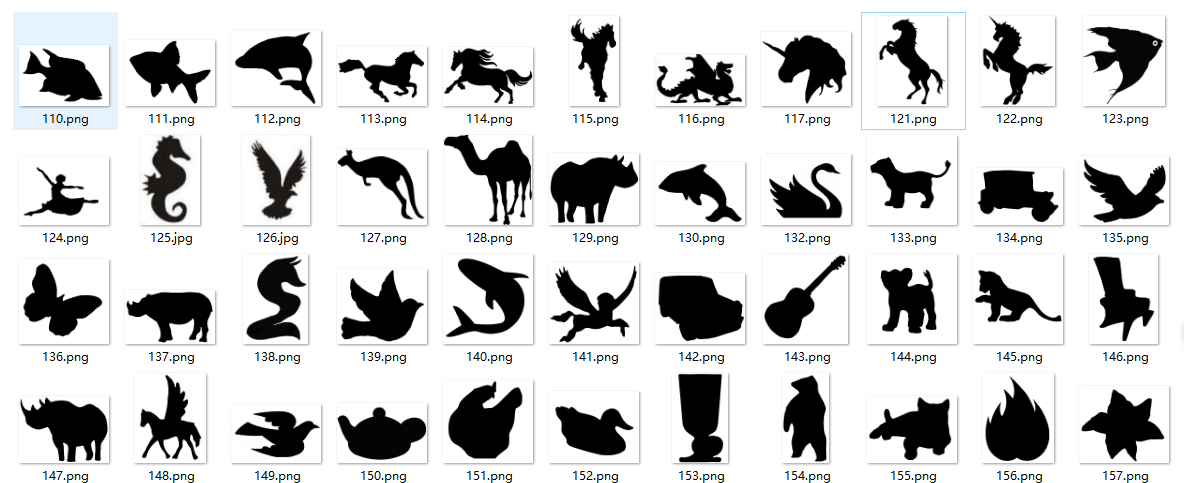




**周报**

2018.07.30—2018.08.05

本周收集了更多的的剪影图像来扩充数据集，并在其中添加了很多比较相似的样本，总数量接近200了，最终目标是达到300以上，如果有时间，还会继续扩充，越多越好。



本周后几天在完善划分的算法，从提取骨架到做初始化分的步骤不够鲁棒，对于刚刚做的数据集里凸起比较尖锐且密集的图像，提取的骨架不是很合适。

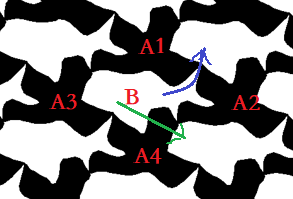
周六讨论并敲定了社会实践的具体行程安排，周六晚上就整装出发了。

**周报**

2018.07.23—2018.07.29

本周前三天都在更多收集的剪影图像来扩充数据集。当数据量到80以上时，添加新数据就不是那么容易了。我一开始搜集了大约300多个包括动植物，生活用品，交通工具，建筑在内十几种类型的黑白剪影图，然后排除大部分语意不够清晰的剪影图，这样的图很可能会因为轻微的变形就丧失含义。再然后删掉一部分轮廓语意差别不大的图案，最后剩下也就50个不到的轮廓。这些图案普遍存在的问题就是噪声点很多，对轮廓提取的算法造成了不小的干扰。我试图用ps的魔棒工具和套索工具对轮廓进行初步提取来避开，然后又发现这些图案本身清晰度都不高，使用诸如PS或PhotoZoom这些放大方法，放大的图片后依然有明显的模糊感，边缘的重影以及噪点，导致ps提取之后出现很多新的噪声点，本来该光滑的边缘也变得坑坑洼洼，最终得到的点数据也出现不小的误差。

后来，我找到了bigjpg.com这个网站，号称使用最新人工智能深度学习技术——深度卷积神经网络（Deep Convolutional Neural Networks, based on waifu2x），会将噪点和锯齿的部分进行补充，实现图片的无损放大。实际对黑白剪影图，卡通图案以及照片的放大效果都相当不错。受限于缺少更多的优质剪影图来源，现在数据集的数量也只达到100左右，新数据的添加速度比较慢。

后三天在实现之前提到的一类物体的平铺，如右图。对于同一类A来说，A2可以沿蓝色箭头指向走一定度数环绕A2作平铺，当A1A2相对位置确定时，A3A4也确定，然后A3A4整体可以沿着绿色箭头指向做一定平移，同时始终保证与A1A2的对应部分相切。算法暂时只考虑了这两个自由度。

**周报**

2018.07.16—2018.07.22

周一讨论之后，任务变为先建立用于检索的数据集。回去后首先跟明海讨论了一下，统一了数据集里数据的格式，并对代码进行了修改。然后搜集更多的可行图案。

周二暑期学校开始注册，我们会场组从中午开始布置晚宴会场，检查图书馆报告厅设备等。

周三周四周五白天听报告，下午以及晚上负责3D打印的DIY指导工作。因为我是临时顶上当小组长，所以也多花了一段时间来熟悉和掌握整个DIY活动的各个流程。整体做下来感觉工作量还是挺大的，不仅需要小组之间配合好，有时候还需要他们多花点休息时间，才能很好的完成整个diy的任务。

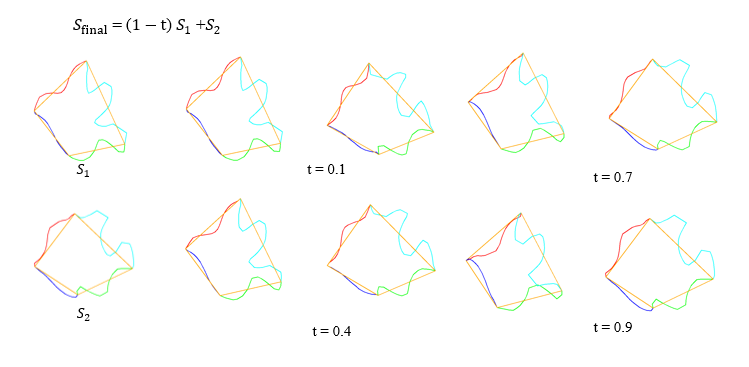
周六整个会场组开始准备毕业典礼，暑期学校结束后负责收拾整理会场。

周日上午负责保研面试的现场记录，下午听了报告。一周下来只有比较零散的时间来完成自己的工作。

**周报**

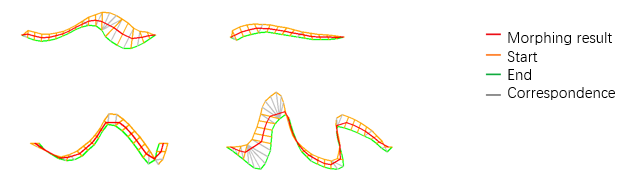
2018.07.09—2018.07.15

1. 在shape morphing的过程中，通过控制变量t来控制形变程度，但此时变形结果的优劣都需要人主观判断，因此整个流程还无法实现自动化。

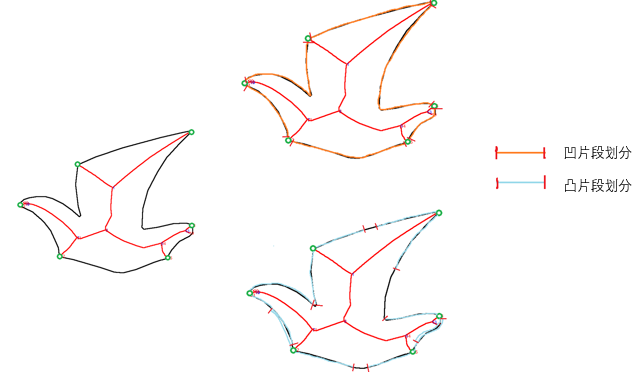


为解决这个问题，设计一个评价函数，为 S1的面积变化量，通过这个公式对S1，S2两个形状的形变程度做一个评估。

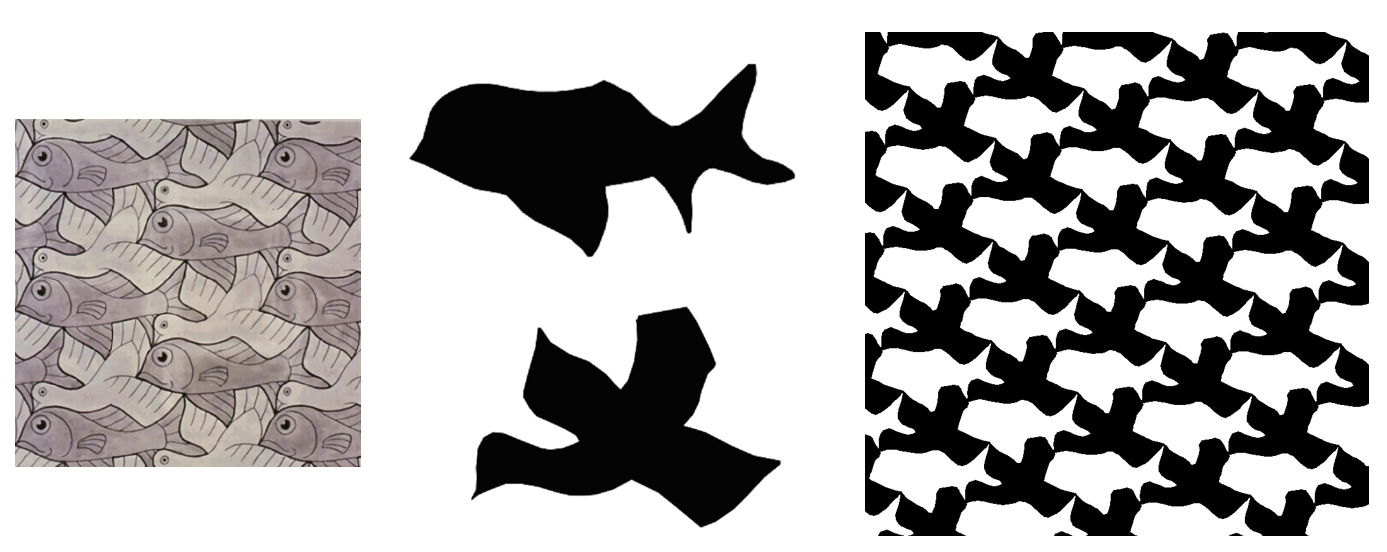
同时在变形过程中，我发现最终的变形会因为融合两个形状而出现边缘变化不够平滑的情况，初步的改进措施是增加采样点数，使多边形边缘的转折不至过于突兀。另外，可以通过一些插值算法，在其变形结果上进行插值，使其变得平滑。

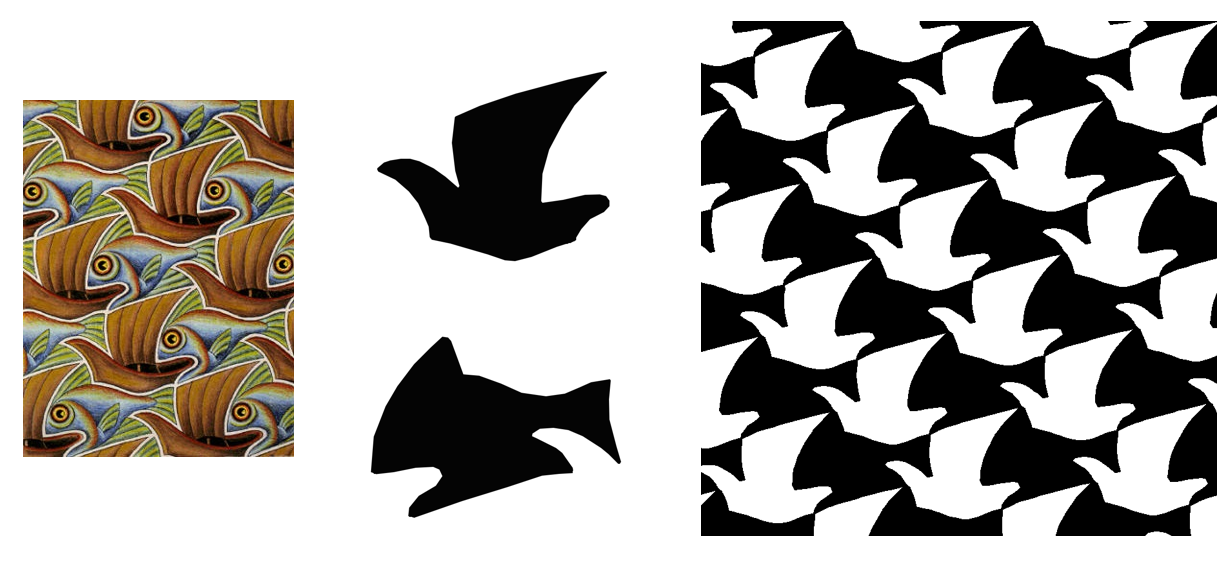


2. 现阶段在做groundtruth时，因为划分算法还没有完全实现，仍需要提前手动划分片段，整个过程十分不智能。因此这阶段的主要工作是完善划分算法。通过跟彭昊的讨论，觉得可以根据提取的骨架对轮廓做初始的划分。首先确定几个凸起点，然后分别得到多段候选片段划分，其中包括任意两点间的凹片段以及以凸起点为中心的凸片段。通过左右微调及合并，最终得到4段局部最优的对应轮廓划分。



根据Escher的画作得到一些密铺的结果，但是观察可得，这些形状都比较刻意，在没有纹理的情况下，最终得到的密铺结果不够吸引人。因此如何寻找一个美观优雅的groundtruth是当务之急，同时也是一个困扰我很久的难题。





在与老师讨论后，觉得自己看问题还是比较局限，总是试图用四边变换来处理规则密铺，同事平常的工作也陷入到具体的某个算法实现上，整体上对这个问题的把握不够。老师提供了一些解决思路，比如建立基础图形库，保存一些具有特殊结构的图形，使得其中任意两个图形都在一定程度上进行匹配或对齐。或者先从一类物体的平铺着手，通过对相同图形间的间隙进行调整变换，来得到第二类密铺图图案。接下来的工作，我将先从后一种方法入手，做一些任意平铺的结果。