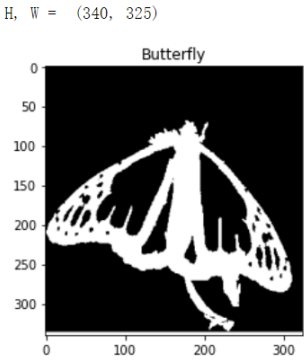
《图像分析与理解》课程——形态学实验

212138 卓旭

Part 1 – 二值图像

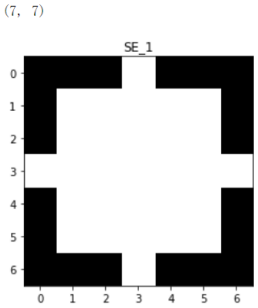
1. 读取 Butterfly 图像

按行扫描 butterfly.txt，结果如下，白色为有效区域，尺寸正确。该图记为I_0。



2. 开操作与闭操作

读取结构元 SE_1 如下，以中心点为原点：



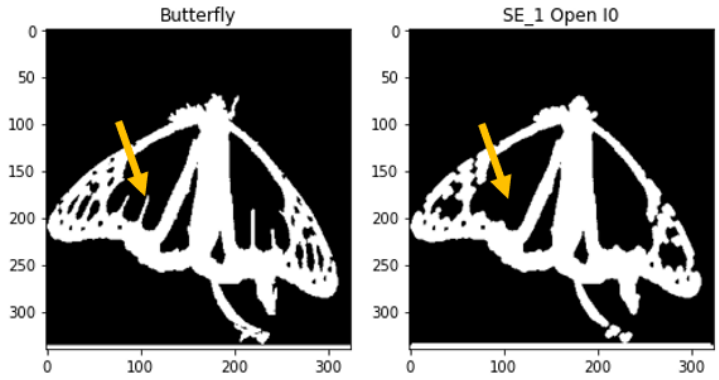
为方便实验，在代码中编写了如下函数：

函数	功能	遵循的定义		函数	功能	遵循的定义
binaryErode	腐蚀	$A \ominus B = \{p \mid (B)_p \subseteq A\}$		binaryComplement	补集	$A^c = \text{not } A$
binaryDilate	膨胀	$A \oplus B = \left(A^c \ominus \hat{B}\right)^c$		union	并集	$A \cup B$
binaryOpen	开	$A \circ B = (A \ominus B) \oplus B$		intersect	交集	$A \cap B$
binaryClose	闭	$A \cdot B = (A \oplus B) \ominus B$				

本实验的实现中，对于图像边缘的像素，无法以它们为中心放下整个结构元时，选择放弃对该像素的处理（若进行 0-Padding 或 1-Padding，都会对后续 Convex Hull 的构造过程产生影响）。

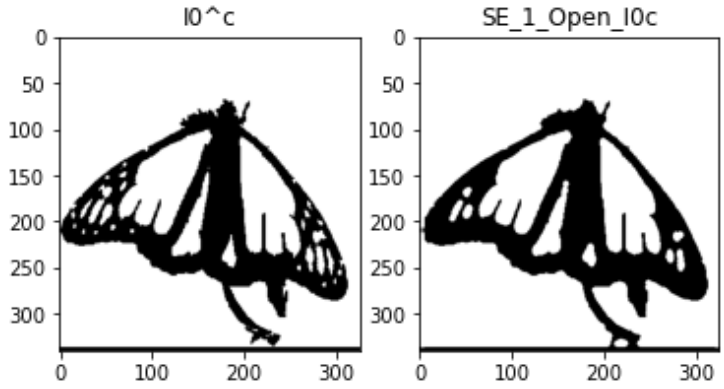
以腐蚀为例，串行的运算过程类似于二维卷积，需要四重循环，效率不高。为此，我编写了 CUDA 版并行加速的腐蚀函数，并借助 PyCUDA 引入 Python 以便调用，具体代码在 cuMorph.py 和 cuMorph.cu，执行前需要配置 MSVC 编译器的路径。

进行开操作，先腐蚀后膨胀，结果如下右图：



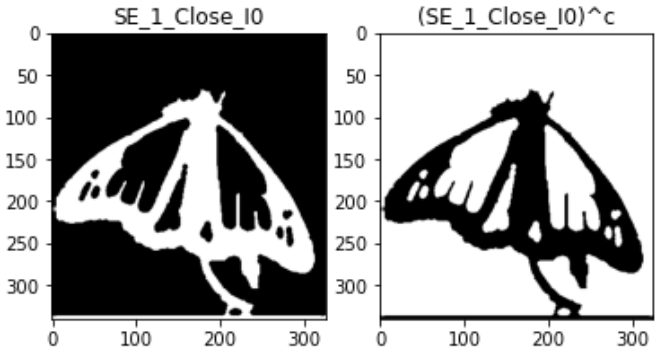
相较原图（左图），最明显的变化是箭头所指部分的细小线状结构被消除了。这是因为在腐蚀时，这些部分不能满足结构元要求，被消去，等到再膨胀时，自然无法恢复出这些结构。这说明开操作能够消去一些细微的结构并保留图像整体。由于 1-Padding 的原因，图像下方的白线仍然保留，尽管它的高度小于结构元要求。

若计算$I_0^c \circ SE_1$，则得到如下结果（左图为I_0^c，右图为$I_0^c \circ SE_1$）：



与前文对开操作的分析类似，蝴蝶翅膀里的细小结构也被消去了。这一例子说明开操作也能够填充一些微小的孔洞（主要是膨胀步骤的作用）。

若计算$I_0 \cdot SE_1$，则得到如下结果（左图为$I_0 \cdot SE_1$，右图为$(I_0 \cdot SE_1)^c$）：

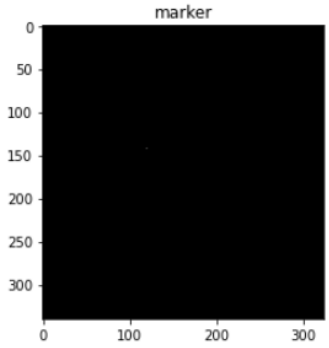


由于$(I_0 \cdot B)^c = I_0^c \circ \hat{B}$，又本例中结构元中心对称，$SE_1 = \widehat{SE_1}$，从而有$(I_0 \cdot SE_1)^c = I_0^c \circ SE_1$，观察到两图相同，验证了该性质的正确性。

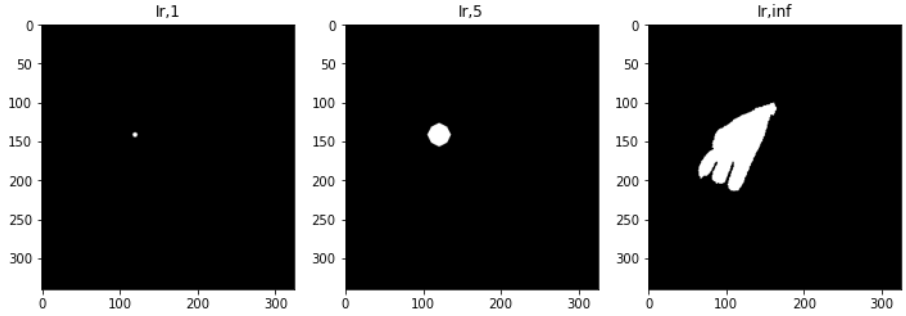
3. 形态学重建

读取 marker.txt，如下，记为I_m（在(row, col) = (142, 120)处有一个点为 1）。

(340, 325)



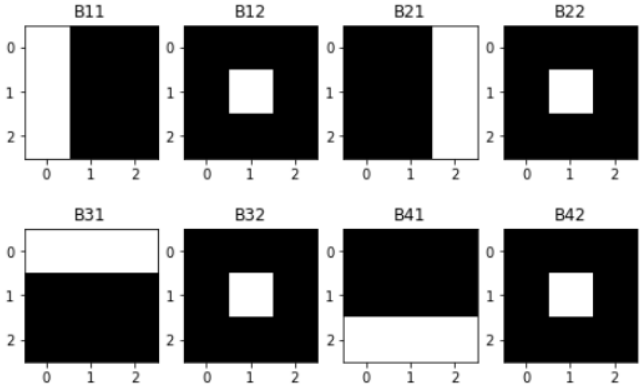
记$I_{r,0}=I_m$。迭代计算$I_{r,k}=\left(I_{r,k-1}\oplus SE_1\right)\cap I_0^c$，直到收敛（使用代码中的 same 函数判断），总迭代轮数为 28，各步骤结果如下（$I_{r,1},I_{r,5},I_{r,\infty}$）：



可以看到该算法符合预期地重建出了种子点生长出去的翅膀左边部分。

4. Convex Hull

记$I_{r,\infty}=I_1$。读取 convex_hull_se 下的结构元如下：



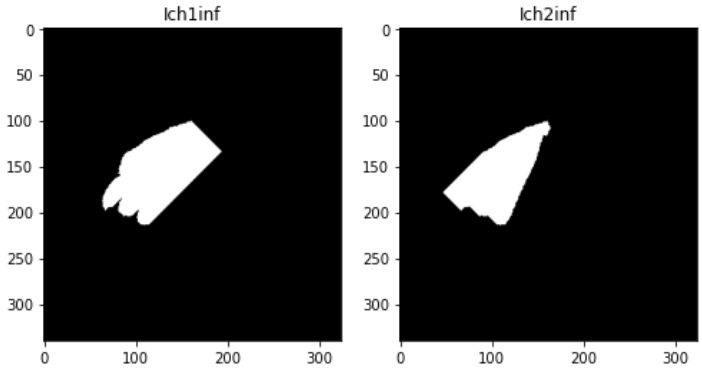
一对结构元（以(B_{11},B_{12})为例）的 Convex Hull 分量的构造过程为如下的迭代过程，直到收敛：

$$I_{ch,1,k}=\left[\left(I_{ch,1,k-1}\ominus B_{11}\right)\cap\left(I_{ch,1,k-1}^c\ominus B_{12}\right)\right]\cup I_{ch,1,k-1}$$

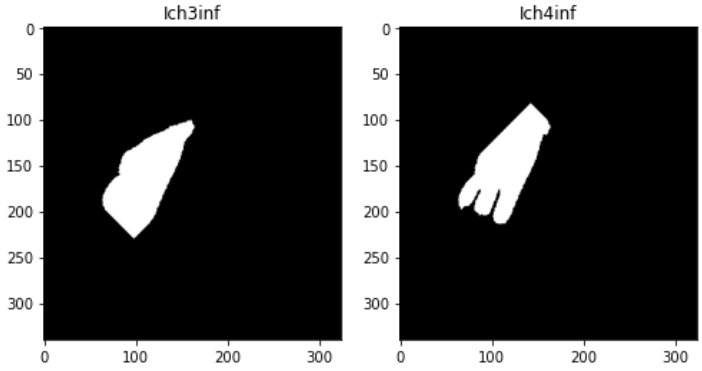
方括号内的运算称为击中-击不中变换。

各对结构元构造出的分量如下：

$I_{ch,1,\infty}$、$I_{ch,2,\infty}$：

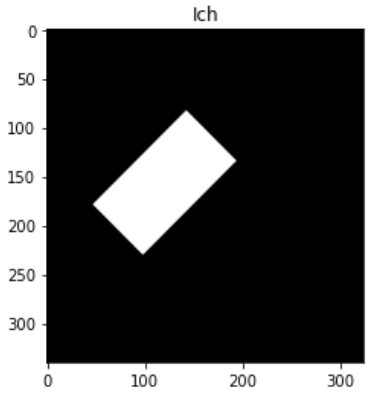


$I_{ch,3,\infty}$、$I_{ch,4,\infty}$：



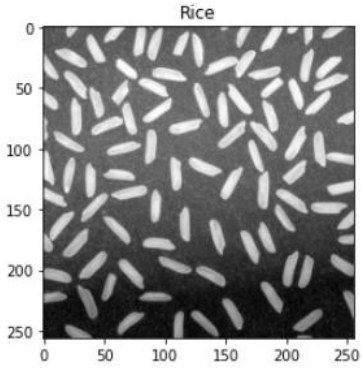
取并集得到最终的 Convex Hull：

$\cup_i I_{ch,i,\infty}$

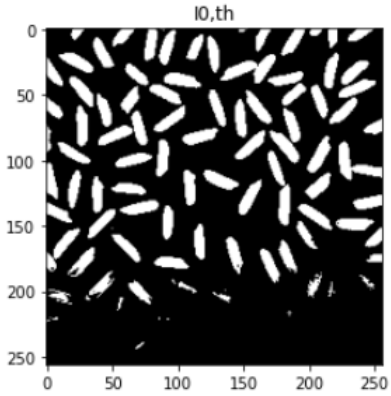


Part 2 – 灰度图像

读取得到原始图像I_0如下：

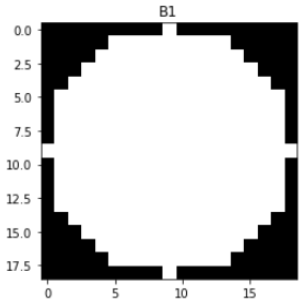


使用阈值 150 进行分割结果如下，白色（1）为前景：



可见，由于图像下部光照与上部不均，固定的阈值不能在全图上取得很好的分割效果。

结构元B_1读取如下：

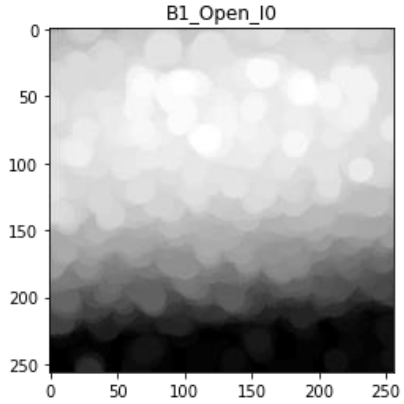


为方便实验，在代码中编写了如下函数：

函数	功能	遵循的定义
grayscaleErode	腐蚀	$[f \ominus b](x,y) = \min_{(s,t)\in b}\{f(x+s,y+t)\}$
grayscaleDilate	膨胀	$f \oplus b = \left(f^c \ominus \hat{b}\right)^c$
grayscaleOpen	开	$f \circ b = (f \ominus b) \oplus b$
grayscaleClose	闭	$f \cdot b = (f \oplus b) \ominus b$
grayscaleComplement	补	$A^c = -A$
tophat	顶帽变换	$f - (f \circ b)$

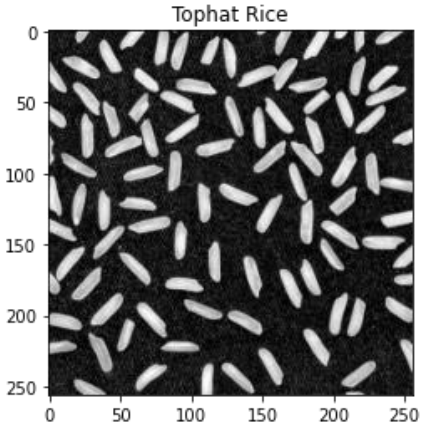
本实验的实现中，对于图像边缘的像素，无法以它们为中心放下整个结构元时，则将原图超出原尺寸的部分认为是 255（白色）。此处如果放弃处理，由于使用的核尺寸较大，效果不好。同样地编写了 CUDA 版本的灰度形态学的腐蚀函数，代码在 cuMorph.py 和 cuMorph.cu。

进行开运算$I_0 \ominus B_1 \oplus B_1$结果如下：



可见，该运算提出了大致的背景颜色分布。

做顶帽变换$I_{\text{top-hat}} = I_0 - (I_0 \ominus B_1 \oplus B_1)$后结果如下：



可见，背景光照被有效地均一化了。通过保持整个运算过程中数据类型始终为有符号的 int32，最终取顶帽变换结果图中的最小值，为 0，不是负值。这一结果是可以预期的。考虑中心对称的结构元b覆盖的一个 Patch，有：

$$f - (f \circ b) = f - \big((f \ominus b) \oplus b\big)$$

而

$$f \ominus b = \min_{(s,t)\in b}\{f(x+s, y+t)\} \le f$$

又

$$(f \ominus b) \oplus b = \max_{(s,t)\in b}\{[f \ominus b](x-s, y-t)\}$$

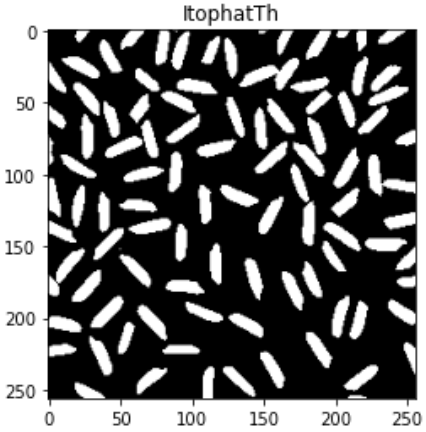
$\hat{b} = b$，则

$$\sup\,[(f \ominus b) \oplus b] \le f$$

故

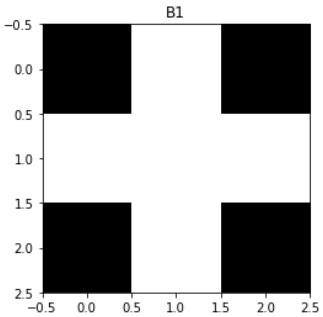
$$f - (f \circ b) \ge 0$$

设置阈值 60 来分割顶帽变化后的图像，得到结果如下：

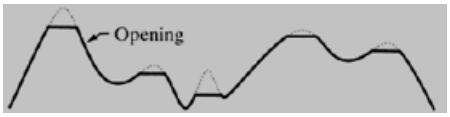


可以看到阈值分割的效果有很大提升，背景的不均匀的带来的影响被去除了。

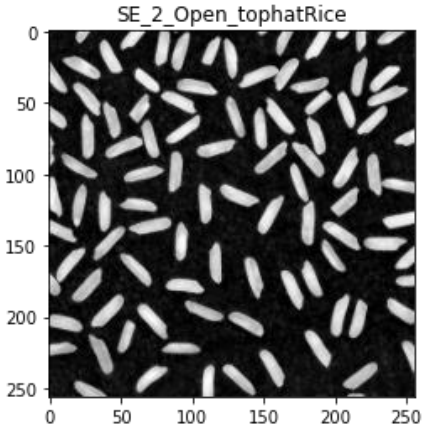
为去除$I_{\text{top-hat}}$中的小亮点，给出的SE_2如下：



根据所学，灰度图像的开操作可以“削峰”来消去这一亮点。



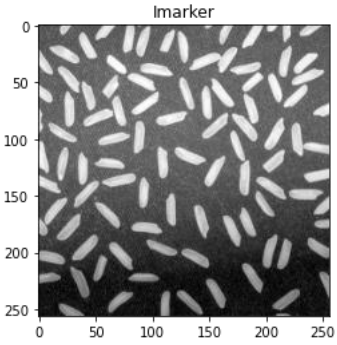
计算$I_{\text{top-hat}} \circ SE_2$结果如下：



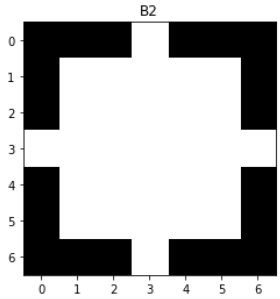
对比前面的图像，该亮点被抹平了。

下面开始进行灰度图像的形态学重建 H-dome 算法。

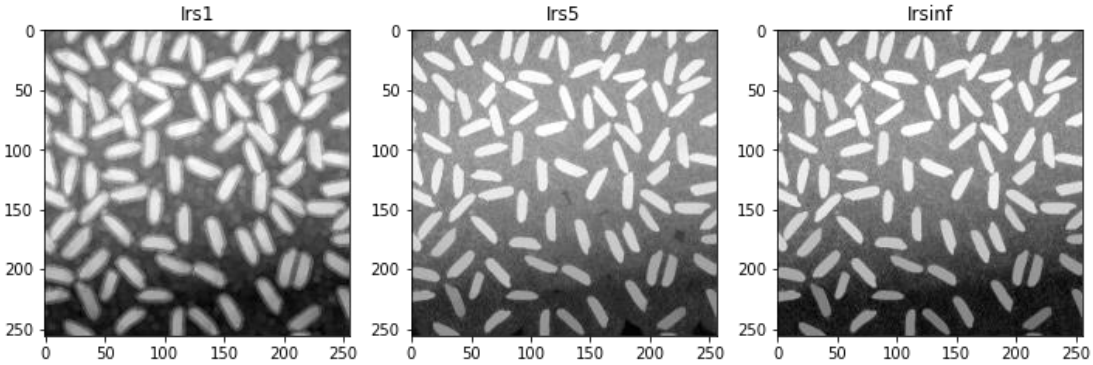
$I_{\text{marker}} = I_0 - 45$如下，作为 marker 图像，而 source 图像是I_0：



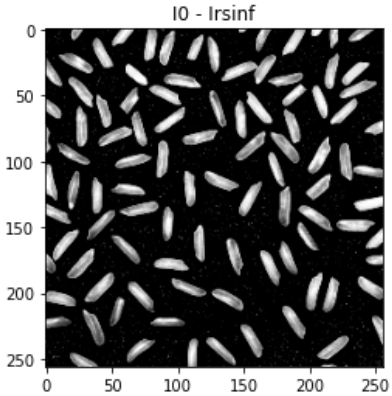
初始化$I_{r,0} = I_{\text{marker}}$，迭代计算$I_{r,k} = (I_{r,k-1} \oplus B_2) \wedge I_0$，$\wedge$表示点对点求最小值。$B_2$为如下结构元：



仍用 same 函数判断两图相同，迭代在 20 轮后收敛，得到$I_{r,1}, I_{r,5}, I_{r,\infty}$如下：

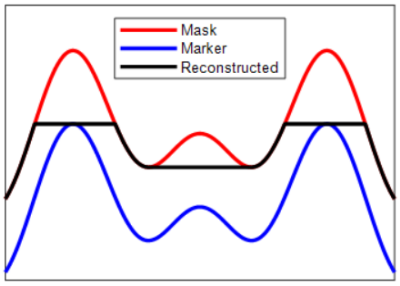


$I_0 - I_{r,\infty}$如下图：



上述结果的解释是，灰度形态学重建得到了 source 图像的“支撑”，而支撑水平以上的变化则被削去，所以可以观察到$I_{r,\infty}$中米粒上的细节没有了。因此，$I_0 - I_{r,\infty}$可以去除支撑用

的背景，只留下米粒及其细节。一个形象的示意图如下：



https://www.mathworks.com/help/images/understanding-morphological-reconstruction.html