

猫的天文摄影笔记：天文摄影中的图像复原

可喵喵

这个例子是狒狒用 10 寸牛反和 ATR 拍的 NGC253。我用这个片子来讲清楚七篇笔记里面没有涉及到的一些东西，例如图像的复原和降噪处理。这个技巧是在可喵退坑时 HG 发现的，但那时的我已经根本没有时间来做天文摄影了。所以可喵并不是第一只深入研究反卷积的猫，在此对 HG 表示感谢。

1 卷积和天文摄影

所谓的图像复原，就是在后期处理过程中消除因为大气视宁度影响带来的模糊，主要的算法为反卷积运算。作为铺垫，本文会先讲明白什么叫卷积，然后才能让读者理解为什么图像复原需要做“卷积运算”的逆运算-反卷积。

已知两个函数 $f(x, y)$ 和 $g(x, y)$ ，定义函数：

$$h(x, y) = \int_{-\infty}^{\infty} f(x - x', y - y')g(x', y')dx'dy' \quad (1)$$

为 f 和 g 的卷积。

这个式子初看起来可能不是很好懂，但我们今天是在讨论图像处理，所以我们可以直把它放到图像里去理解。如图 1 所示，有一张大图（白色矩形）和一张小图（浅蓝色矩形），下面我们把前者称为“原图”，后者称为“卷积核”。大图的原点在左下角，小图的原点在蓝色点的位置。于是图中红色向量 + 黄色向量 = 绿色向量， $(x, y) + (-x', -y') = (x - x', y - y')$ 。图像的本质是一个二元函数（也有的说法是一个信号，都一样），其中原图的函数我们定义为 f ，“卷积核”的函数定义为 g 。现在我们把卷积核覆盖在原图上。在 A 点位置，原图的亮度为 $f(x - x', y - y')$ ，AB 关于浅蓝色原点对称，在 B 点的位置，卷积核的亮度为 $g(x', y')$ 。两者做乘积

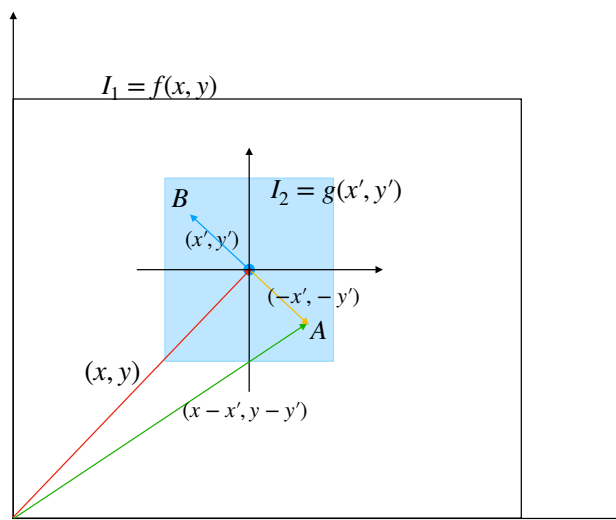


图 1: 卷积的几何意义

后，就是 $f(x - x', y - y')g(x', y')$ ，如果我们把所有的 (x', y') 都遍历一次，结果就是 $f(x - x', y - y')g(x', y')$ 关于整个空间的积分，也就是蓝色点 (x, y) 位置的卷积值了。在大多数情况下，卷积核的数值是呈中心对称的，所以 $f(x - x', y - y')g(x', y')$ 也可以用 A 点原图的亮度乘以 A 点卷积核的亮度来代替。

简单的说，做卷积运算分三步：

第一步，把卷积核覆盖到原图的某一位置。

第二步，把每个位置的原图亮度和卷积核亮度相乘，得到一个新的数值。

第三步：把卷积核所有位置的新数值相加，得到蓝色原点处的卷积结果

然后我们把蓝色原点移动一个像素，重复以上 123 步，得到一个新位置的卷积结果。让卷积核的原点覆盖遍原图中的每一个点，就可以得到一张新的图，这就是图像的卷积结果了。

我们来看一个卷积在图像运算中的具体例子：

高斯模糊 (图 2)。

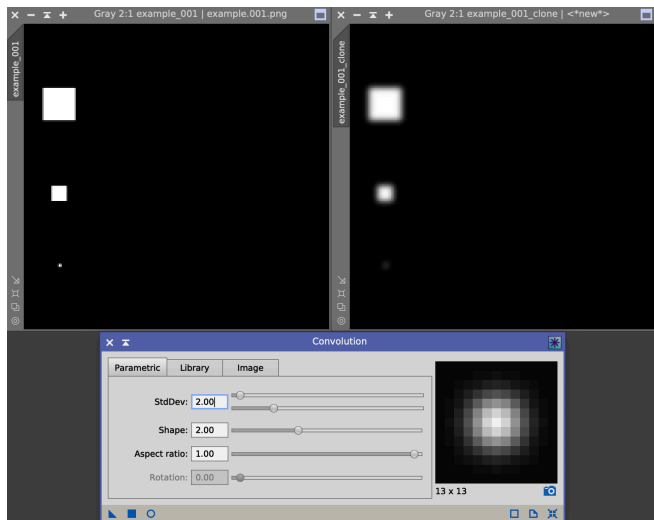


图 2: 高斯模糊

在这个例子里，我分别选用了不同大小的正方形和一个高斯卷积核（是不是很像一个星点）来做卷积运算，得到的结果是分别把这几个图形模糊掉了。如果把最下面那个最小的正方形视作一个最理想的星点（其实就是一个 δ 函数），不难看出一个理想星点与这个高斯核做卷积的结果，就是高斯核本身。这时候我们说：“星点”（Point）被这个高斯核“扩散”（Spread）了。于是，这一类特殊的卷积核就有了它们独特的称谓：点扩散函数（Point Spread Function）。它作用在一个理想的点上的结果就是把它扩散成一个高斯波包；而它作用在一个图像上的结果则是把这个图像模糊掉了。于是这个操作也被称为高斯模糊。

接下来，某猫从统计学的角度而不是波动光学的角度来解释一个事实：大气抖动和赤道仪跟踪误差带来的效应，等同于给一个理想的图像做卷积运算。

如果排除大气干扰和赤道仪追踪误差，恒星在焦平面上的成像是理想的点。由于大气扰动等因素的存在，这个点的位置是相对像平面在实时变化的。如果你有过导星或者拍摄行星的经历就会知道，星点打在像平面上的位置是随机的，但位置的分布大致上是一个高斯分布。在这样的原理的作用下，

星点的图像落在拍摄过程中每一个点的相对偏离量 (x', y') 都对应有个概率 $P(x', y')$ 。这个 $P(x', y')$ 在统计物理里面通常被称为 Probability Distribution Function (PDF)，意为概率分布函数，它的含义是星点在某个时刻的位置偏离为 (x', y') 的概率密度。

假设此时的图像为 $I(x, y)$ ，图像在某一时刻的整体抖动后的结果我们就可以用 $I(x - x', y - y')$ 来描述。我们知道，长曝光的本质，是在曝光时间之内的无数个静态瞬间图像的叠加。当图像偏离原来位置 (x', y') 时，它在整个曝光过程中贡献了多少亮度？我们可以用 $I(x - x', y - y')P(x', y')$ 来表示，这里的 P 是概率密度，也有了权重的含义。所以最后我们得到的长曝光图像，是所有时刻的亮度分布 $I(x - x', y - y')$ 乘以它的权重 $P(x', y')$ 之后求和的结果，这个求和也可以写成积分。

综上所述，图像和一个点扩散函数的卷积 $\int_{-\infty}^{\infty} I(x - x', y - y')P(x', y')dx'dy'$ 就是我们长曝光之后得到的图像。严格来说，这里的 $P(x', y')$ 应该写作 $P(-x', -y')$ ，在卷积核不对称的时候需要这么写（例如不规则拉线）。

通过以上的定性分析，我们理解了为什么大气的扰动和赤道仪的跟踪误差可以用数学上卷积的形式来表达。于是，想复原一张被大气扰动之后的图像，我们需要做它的逆运算：反卷积（Deconvolution）。在公式 1 中，如果已知了最终图像 $h(x, y)$ 和卷积核 $g(x, y)$ ，求原始图像 $f(x, y)$ ，这种运算就是反卷积。

最后来回答一个问题：什么样的照片适合做反卷积？星点的半高宽大于 2.0 像素的照片，可以通过反卷积来提高图像的细节。在这里我想通过 Nyquist 采样定理来理解这件事：如果我们对一个连续的信号采样，在什么样的采样频率（在天文摄影里是空间频率）下才是真正无损的？只要大于信号中包含的最高频率的两倍就行了。所以，在天文摄影中，相机的像素实际上就是在做空间采样，而望远镜成的像是一个模拟信号。当采样频率（像素的空间频率）高于这个信号中最高频的部分（星点）的两倍时（FWHM=2.0），我们就可以还原出原始信号了。在

这种情况下，采用更密集的像素也不能得到更多的细节，天文摄影中一般称为“过采样”。反之，FWHM 小于 2.0 的就是欠采样。

对于一张欠采样的照片，由于像素的空间频率太低，从物理上隔绝了照片中该有的一些细节，这种情况下就算做了反卷积也不可能得到比原始照片更多的细节。但过采样的照片对于望远镜成像的信息的采集是完整的，只是图像中的细节被大气扰动抹去了，这时候才有理由采用反卷积的算法还原这些细节。有些人误解了这段话的意思，于是把望远镜的对焦故意对不准，以获得高一些的半宽以便做反卷积。这种就是为了还原而还原的沙雕行为，因为做了反卷积之后的图像也不可能比一开始对焦就对好的图像更清晰。哈哈哈哈哈

天文摄影中的信号不可能全部都是正弦叠加，所以这个 2.0 的标准也不像 Nyquist 定理中卡的这么死。按天文摄影的实际情况，星点的半高宽在大于 2.5 时比较适合考虑做反卷积。

2 认识 PixInsight 的反卷积工具

由于某猫的知识水平有限，在这里我没有能力讲出反卷积的具体图形算法来。但工具使用的经验还是可以讲出来的。我们调出 PixInsight 里的 Deconvolution 模块，发现有这么几个栏目：

1) PSF 点扩散函数。这一栏中有 Parametric PSF, Motion Blur PSF 和 External PSF。详细解释一下这三个选项：第一个是参数化的 PSF, stdDEV 是 PSF 函数的半高宽, Shape 的数值表示你可以让这个 PSF 不一定符合高斯型（虽然高斯是最常见的），Aspect Ratio 是这个 PSF 的不对称度，调整这个数值可以让 PSF 拉长或者变胖。Motion Blur PSF 主要是修正由于曝光中的运动带来的图像失真。打开它之后也是一个给定参数和条条框框的选项卡。之需要注意的是，这两个都是在你看不到星点的时候才用得上的，例如拍摄行星和日常物体。对于看得到星点的深空天体摄影，我们有更方便的方式来取得公式 1 中的卷积核 g ：那就是 External PSF。

2) Algorithm, 算法。由于卷积的定义式是一

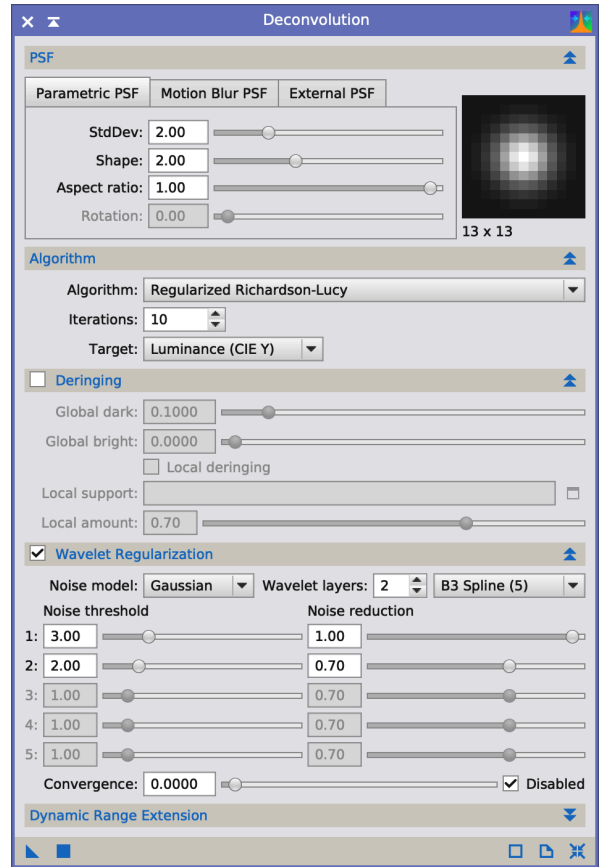


图 3: PixInsight 中的反卷积工具

个无穷空间中的积分，我们不能直接解这个方程来得到 $f(x, y)$ ，人们开发出了一些反卷积的算法，例如 Richardson-Lucy 迭代，通过一步步试探来估计原函数 $f(x, y)$ 。深空摄影中我们一般选择这个默认的 Regularized RL。其他的算法在行星摄影中会用到，要求图像的信噪比很高。这个 RL 算法还有个比较重要的特征：它能保持图像的线性性不变。这在天文测光中比较重要。这个算法可以保证，反卷积的结果就算 Ringing 效应再严重，把两颗恒星分开的前提下也能保证让他们的亮度不发生改变。所以在深空摄影中我们也沿用这种守恒的算法。

3) Deringing, 去振铃效应。在业余天文摄影中，我们一般都不希望做完反卷积的图像星点周围有一圈黑的，那样太影响美观了。所以需要在反卷积的同时去掉振铃效应。这里有暗部和亮部两个拖动条，也有 LocalSupport，我们一般制作亮星的位置图来

充当 LocalSupport。请记住，Deringing 做得好，反卷积的迭代次数就可以很高，迭代次数越高，通常来讲细节就越清晰。做反卷积运算时一般迭代到图像刚刚出现振铃效应为止，所以这一项设置的好坏直接限制了反卷积的上限。

4) 降噪。这里的降噪一般就是小波降噪，需要做一些试错才能得到一张图的经验数值。

3 外置 PSF 和 LocalSupport

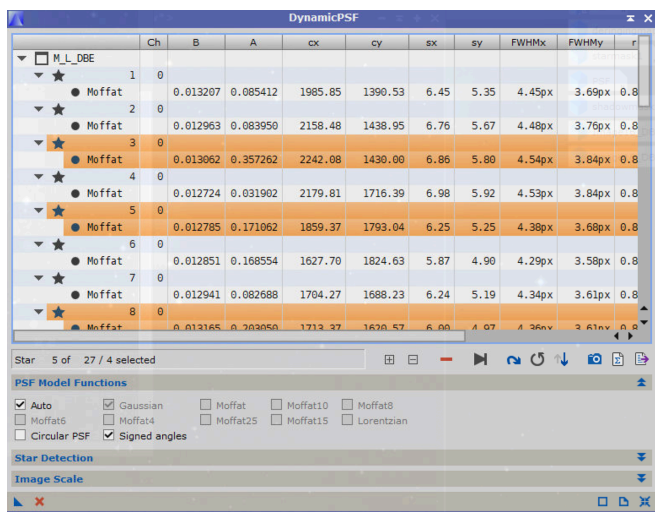


图 4: 动态 PSF

PixInsight 中提供了一个 Dynamics PSF 工具，可以直接从图像中的星点做拟合得到想要的 PSF。这样做的好处在于，我们的 PSF 选取没有盲目性，也不用猜参数。打开动态 PSF 工具之后，用鼠标点击图中的星点备用。这个 PSF 工具可以把星点拟合成高斯型或者 Moffat 型函数（但实际用下来大多数结果都是 Moffat）。下面我逐一解释图 4 中的每一项含义。

Ch: Channel 通道。一般我们只对明度通道做反卷积，所以这一项基本都是 0。红色通道也是 0，绿色通道是 1，蓝色通道是 2。

B: Background, 星点附近的背景数值。

A: Amplitude, 星点的峰值

cx/cy: x/y-coordinate, 星点的坐标值。

sx/sy: x/y-standard deviation. 当拟合形状为高斯或 Moffat 时，x 或者 y 方向的 PSF 函数标准差。这一项是星点大小的一个体现。

FWHMx/y: x 或者 y 方向的星点半高宽。这一项提供了另一种测量星点大小的指标。

r: Aspect Ratio. r 越接近 1，星点越圆，所以这一项也可以叫做星点的圆度。

theta: 如果星点不是正圆的（很少有正圆的星点）那么它的旋转方向是向哪里的？这就是旋转角。

我们选择星点时，最有用的指标是 Amplitude。注意不要选到那些“平头”的星点，也不要选择信噪比过低的星点。如果是给一个星系做反卷积，建议把星点全部选在星系周围。这么做的原因是，离轴很远的星点可能会有彗差，但处于画面中央的星系是不会受此影响的。这时候我们没必要把彗差也考虑进来修正。

这一切都准备完毕，你可以先选择你需要的星点，然后点一下 PSF 工具里的小照相机图标完成动态 PSF 的创建。

下面我们来建立一个 Deringing 用的 LocalSupport 文件，我们要对亮星的 deringing 给点特殊照顾才行。从 PixInsight 里调出 Starmask 模块，如图 5 所示。解释这几个主要选项的作用：

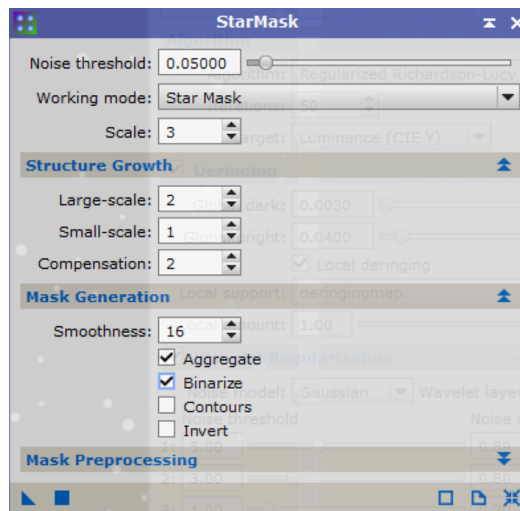


图 5: 星点蒙版生成工具

1) Noise threshold: 噪声阈值。这个数值默认是 0.1，选的越低，就会有越多的暗弱星点（甚至天

体结构) 进入 starmask。设置得高了, starmask 里面就只保留了一些亮星。

2) Scale: 尺度。这一项的数值越大, 生成的星点图里就越容易出现“亮星更大, 暗星更小”的情况。

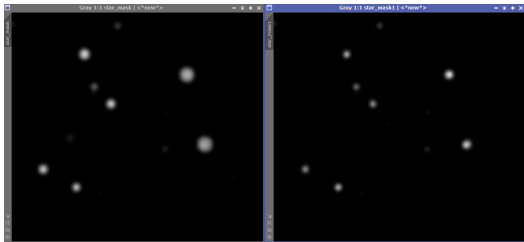


图 6: 左图: scale=5, 右图: scale=2

3) Large-scale 和 Small-scale 都是控制生成的星点的大小的, 增大这个数值, 所有的星点都增大, 但亮星不会比暗星增大的更明显。

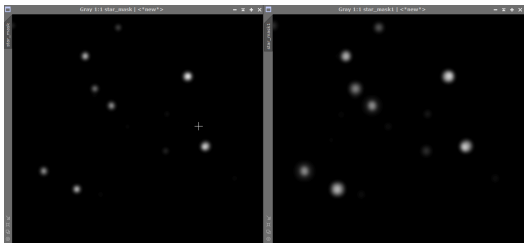


图 7: 左图: large-scale=2, 右图: largescale=7

我们主要就调节这几个数值, 让 LocalSupport 完全覆盖图中主要的亮星。然后在 Deconvolution 的 Deringing 栏里面填入 Global Dark 和 Bright 的数值, 这两个数值是全局的去振铃效应选项, 对于画面中任何一点都成立 (蒙版区除外)。这个数值一般是试出来的, Global Dark 的数值如果太小, 星点周围就会出现一圈暗环, 如果过高, 星点周围就会出现一圈白雾。Global Bright 的数值如果设置的太低, 亮部就会出现很强烈的“塑料感”。接下来是局部消振铃。勾选 Local deringing 的选框, local support 选择刚才生成的 starmask 星点图。我们可以用直方图工具切去这个星点图的最亮部分, 让星点图整体显示出“过曝”的感觉, 这样设置之后 deringing 的效果会更好一些。

但仅仅做这些还是不够的, 依然会有星点成为

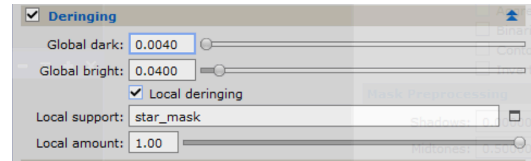


图 8: Deringing 数值设置

漏网之鱼。再者, 背景处的全黑部分根本不含有细节, 根本不需要进行反卷积运算, 如果做了反卷积, 背景反而会出现颗粒状噪声。所以下面我们来制作一张 LightMask 明度蒙版以盖住背景。我们复制一

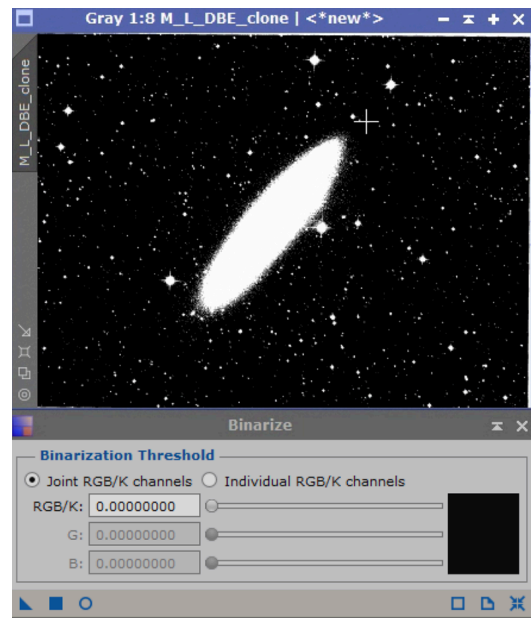


图 9: 制作二值化蒙版

张图, 调出 PixInsight 的二值化工具 Binarize, 选择一个合适的数值做二值化, 背景全黑, 星系全白, 如图 9所示。把这张蒙版应用到要被反卷积计算的图上, 背景就不会做任何运算了。

我们可以把这张蒙版做一个模糊, 即去掉小波变换中的第一层。至此, 在反卷积的同时我们使用了 Local Deringing 和亮度蒙版。但这仍然不够。Local Support 只管亮星周围的 deringing, 而没有办法管暗星。接下来我们使用一种很暴力的方式强制反卷积算法不在暗星的周围运算, 避免出现振铃效应。

从图 10中我们注意到, 这张星点蒙版与星点之间还是有一些间隙的。我们可以考虑缩小这个间隙。

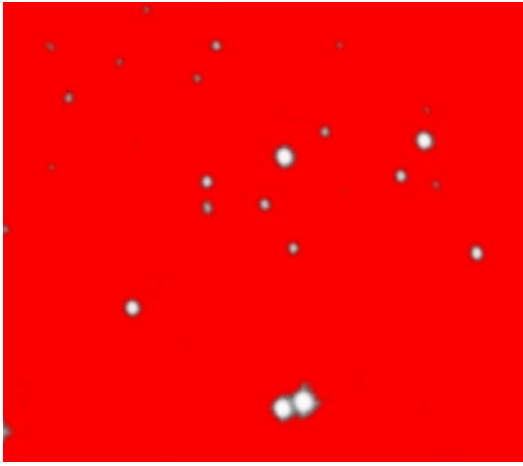


图 10: 亮度蒙版在遮住背景时, 对星点的外围还保留了一些空间

先把原图做非线性拉伸, 再如《天文摄影中的频率》篇里那样制作一张星点蒙版, 但注意这张星点蒙版要完全遮盖住刚刚制作的亮度蒙版。这里面使用最大值滤镜时可以把扩张 (Dilation) 滤镜的孔径设置成 5x5。

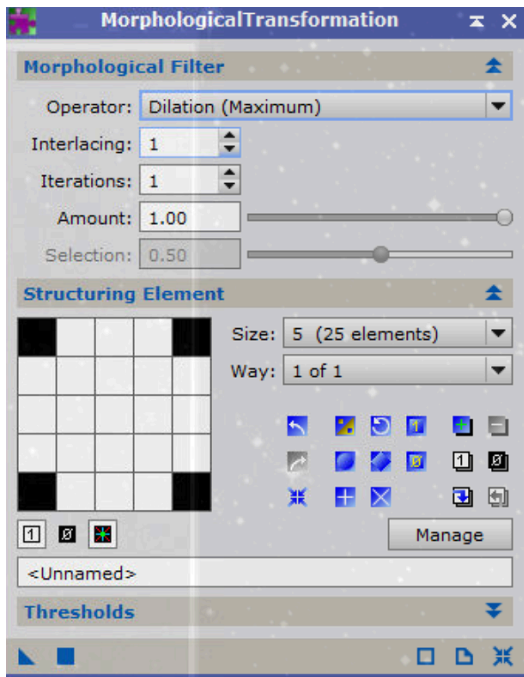


图 11: 扩展滤镜的应用

把这张星点蒙版套在之前的亮度蒙版上, 调出 Morphological Transformation (形态学变换) 工具, 做一次或两次最小值, 使得亮度蒙版中的星点缩小

到恰好包住暗弱的星点为止。使用变换过的亮度蒙版, 我们就可以放心地使用反卷积工具了。这时候无论是亮星还是暗星都受到了保护。

反卷积的迭代次数选择, 一般是几十次就够了。前几十次迭代的效果会非常明显, 越往后迭代同样的次数越是收效甚微。下面这张 NGC253, 就是按此程序做反卷积的结果。



图 12: 例子: NGC253

4 还有更骚的操作吗?

有! 刚刚我们只是考虑了限制反卷积的空间。但如果你做反卷积的对象不是星系, 而是一些形态复杂的星云, 这个办法有可能失准。因为 Shadowmask 并不具有筛选频率的特征, 它根本就不是用来筛选星点的。如果说它能做什么, 只有背景上的星点它能筛选出来 (很多时候还不完整)。所以我们可以转换一下思路, 不如直接在原片里就把星点干掉。

于是顺理成章地, 我们就得到了最佳化的反卷积思路: 在反卷积之前先使用一个星点蒙版套住所有的恒星, 然后做两到三次缩星, 让星点变成如图 13 这样的形态。星点蒙版制作的时候, 不必保证图像的线性, 反正它也只是个蒙版而已。对了, 在这一步的时候, 记得复制一张没做最小值时的原图, 之后要用。

至此, 我们做反卷积需要这么几张辅助图像:

图 1: PSF

图 2: hadowmask

图 3: 亮星的 starmask, 由 PI 的 starmask 工具生成

图 4: 暗星的 starmask, 由笔记《天文摄影中的频率》里提供的方法生成。

图 5: 复制一张没有经过任何操作的原图 当我



图 13: 缩星

们再次进行反卷积时, 可以把 deringing 的 global dark 和 global light 全设置成 0, 然后进行肆无忌惮的反卷积, 一直迭代到 PI 出现粉色字的 warning 提示为止。图 ~ 13 的左半边是没做过反卷积的效果, 右半边是做了反卷积的效果, 可以很明显地看到一个模糊一个清晰。于是图像复原的效果就达到了。需要注意的是, 这张 IC342 的图来自于一个星点有些前采样的素材 (裕众 102APO pro+QHY695), 如果采样比较丰富, 效果可能还会更好。

这一步做完了以后, 大家关心的问题是这个最小化之后的星点怎么办。接下来我们需要复原这些星点。这就是我们之前留了一张没做缩星也没做反卷积的图 5 的原因。我们这一步的手段, 是使用图 5 的星点来替换原图里的。首先把星点蒙版图 3, 再做一次形态学扩张滤镜 (Dilation), 然后把亮星蒙

版 (图 3) 套在待操作的图上, 在 Pixel Math 里输入图 5 的名字, 作用到待操作的图上。表示蒙版之内的区域数值等于图 5 的, 蒙版之外 (蒙版为全黑区域) 的数值不改变。这样就把星点贴回去了。

前面那段, 把星点蒙版再做一次形态学扩张, 是因为做了反卷积以后, 星点残余的痕迹与天体衔接的地方很可能出问题。为了保险起见, 可以做一次扩张滤镜, 让原片里星点周围更多的空间来替代操作后的片子。

最终的出图效果, 可能超越了我们平时对一只 102APO 解析力的印象, 如图 14



图 14: IC342 成品

说句题外话。反卷积这里面, 我们也可以做一些额外的操作。例如有一些在星系中央过于微小的恒星, 做星点蒙版时按常规的办法不能完成提取, 我们可以用 ATWT 工具保留前两层, 然后再做一个二值化来提取这样的恒星, 然后用一个最大值运算添加到已有的星点蒙版上去。这些恒星周围出现 ringing 也是会让反卷积大打折扣的。再例如, 对星系反卷积完成之后, 我们可以再对星点进行一次幅度很小的反卷积, 这些都是更骚的操作。还有一些奇怪的操作有待开发, 我会写在网站专门的栏目里。