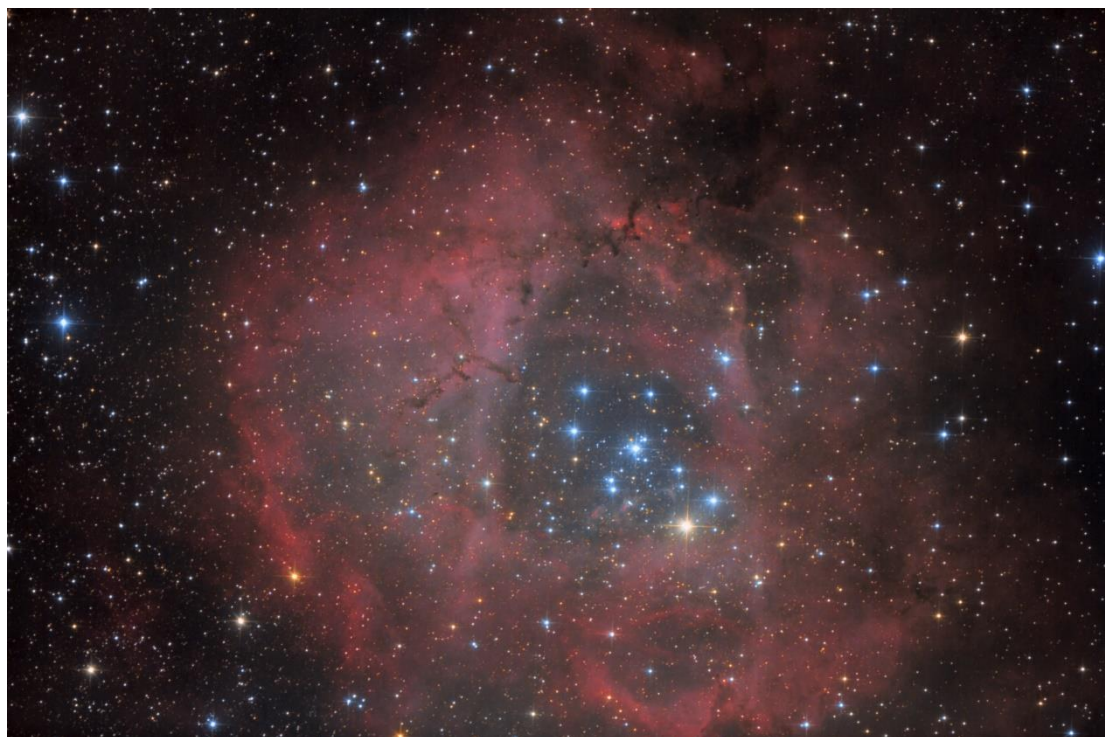


星芒修复、保留星色——PI LE 上 Masked Stretch 的实现

@HG

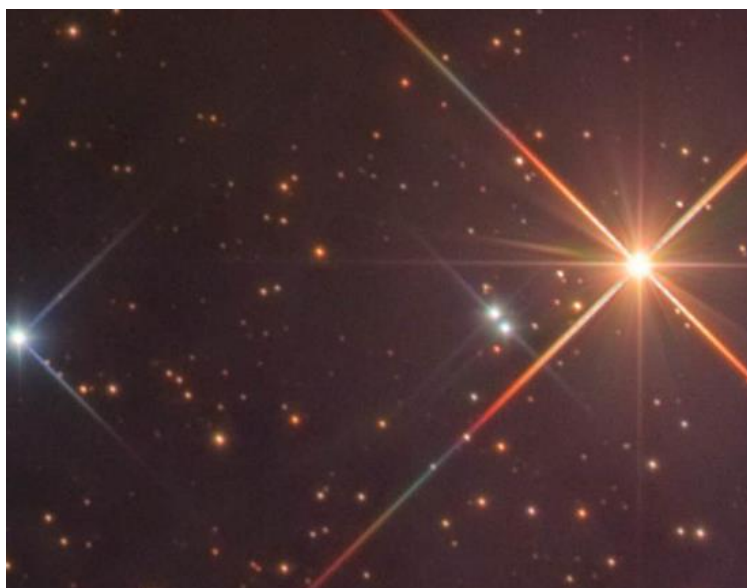
很久很久以前，有人问我，HG 啊，为什么你处理的片子星点总是不太爆呢，为什么我的片子亮星就是一片惨白一大坨呢？为什么你的片子星点颜色这么丰富呢？



当时我给的答复是，你拉伸的时候就要注意星点的问题了。

我也没有多说，因为那时候我也不知道他是怎么把星点弄爆的。我的拉伸手法其实没有什么特别的，就是纯粹的 Histogram Transformation，之后可能会配合一下 Curves Transformation，非常基础而简单，但是最终结果看起来要比大部分人要好。

然后我为此还沾沾自喜了一段时间，直到看到这种东西。



还有这种。

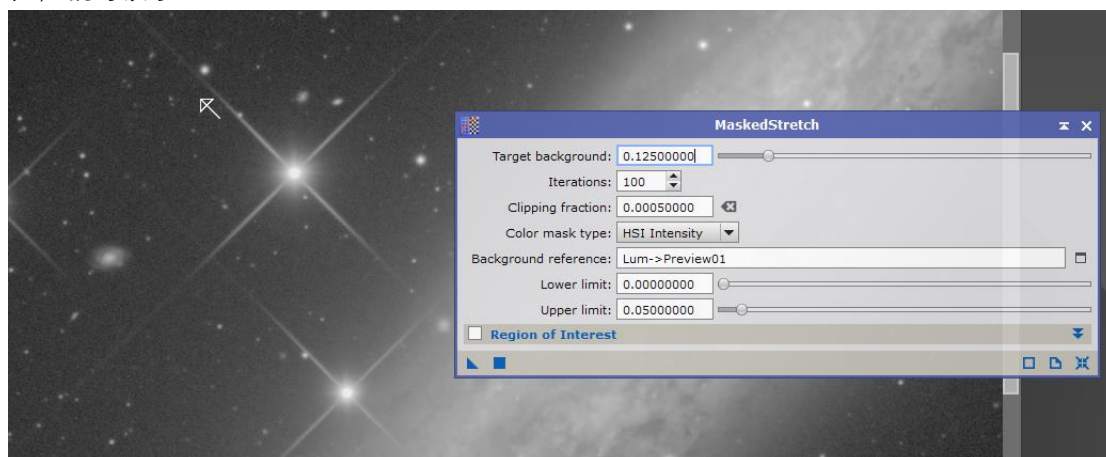


我……



后来知道 P11.8 里有个 Masked Stretch 的东西就是做这事的，虽然会有降低反差的副作用，但是对于光晕的抑制确实有奇效。

这玩意在 PI1.8 里面操作非常简单，框一块背景出来作为参考，填一个背景亮度目标值，就可以了。



以上是 2017 年 7 月的事情。

既然都一年多了，我怎么忽然又想起这事情呢？因为……可能我有点脑抽。
这玩意的原理其实非常简单。我们看看 Inside PixInsight 里面怎么说。

MaskedStretch

The MaskedStretch process by Carlos Milovic and Juan Conejero is a new implementation of an earlier script (Sonnenstein, Serrano, del Pozo). MaskedStretch (MS) is designed to automatically stretch the all-important midtones of astro-images while preventing saturation of highlights. This is accomplished by applying a series of weak stretches iteratively, masking the image successively with the result of the previous iteration. As the stretching begins, the masks are weak so the stretching boosts the image globally. Once the image reaches a reasonably bright level however, the masks become stronger to selectively protect bright structures (Fig. 13.6).

This is accomplished by applying a series of weak stretches iteratively, masking the image successively with the result of the previous iteration. As the stretching begins, the masks are weak so the stretching boosts the image globally. Once the image reaches a reasonably bright level however, the masks become stronger to selectively protect bright structures.

翻译的话，机翻就能理解。

我当时看到这段的时候……



特么原理这么简单？

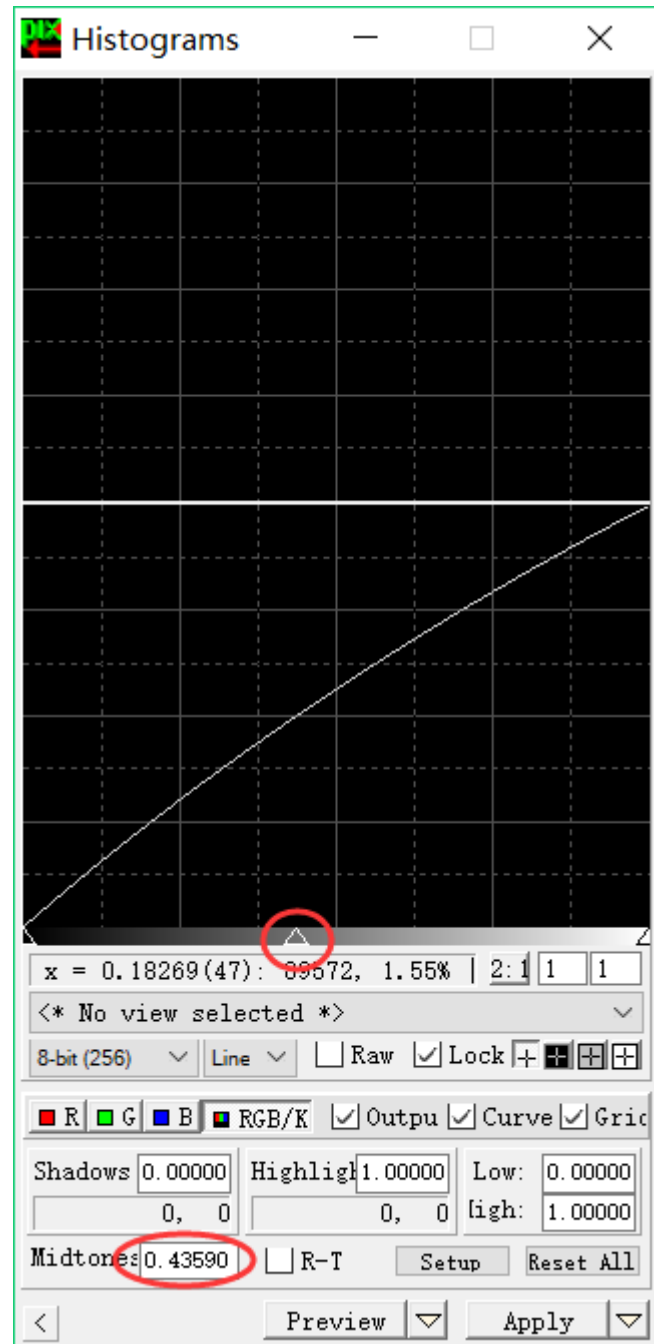
如果手动来做的话就可以这样：每拉伸一次，就拿拉伸结果作为蒙版蒙上去（实际上

就是蒙了它本身)，然后再进行下一步拉伸。

好。

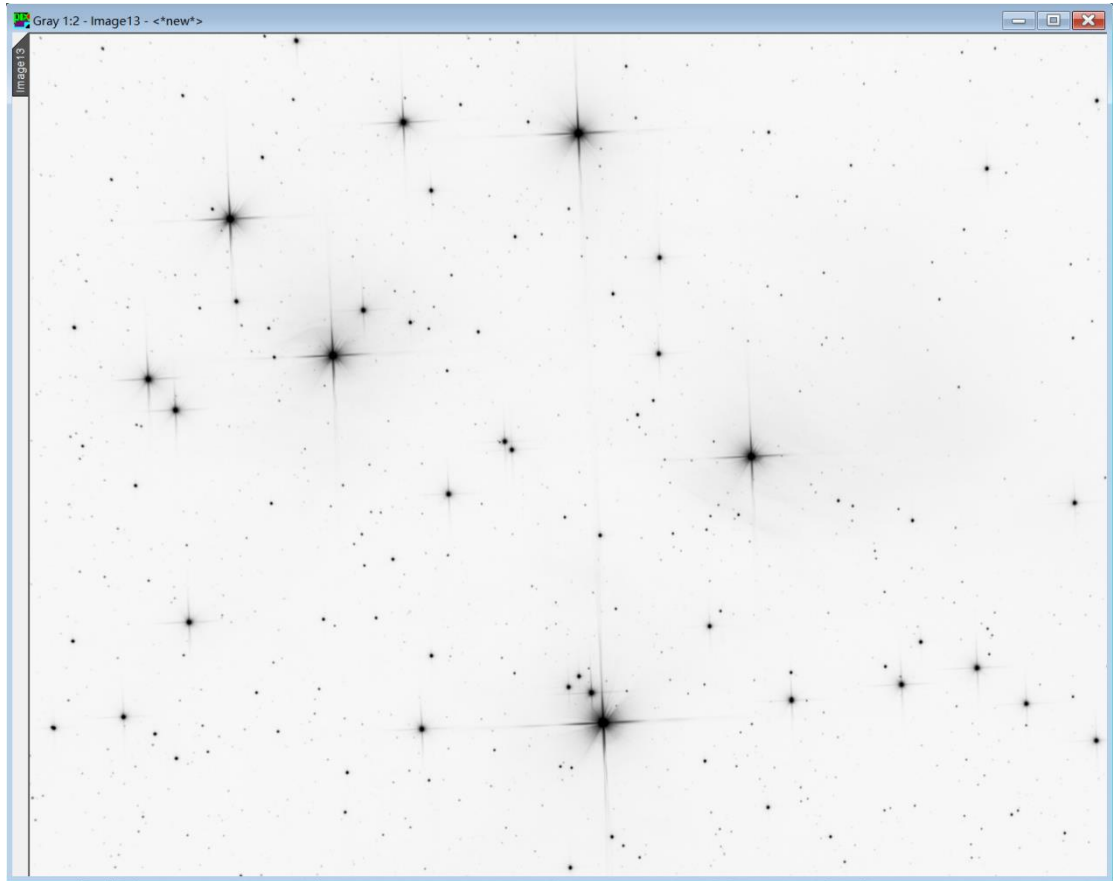
我们打开 PI LE，然后打开做好 DBE 需要拉伸的图像（Image01）。

先确定一个 Histogram Transfer，力度一定不能大，很轻微就行。



然后拉伸一次。

之后 Image→Duplicate，把结果复制出来（如果没进行别的操作这图会是 Image02），对结果反色（Ctrl+I），得到如下图的结果，然后再回到原图，右键，Mask→Select Mask（或者直接按 Ctrl+M），选择我们刚才复制出来的那张图（Image02）。



蒙版就位以后我们再应用相同的一次 Histogram Transfer。之后再对 Image01 复制到 Image03，同样反色然后把 Image03 作为蒙版，再次拉伸 Image01。如此循环往复直到影像亮度达到理想效果。

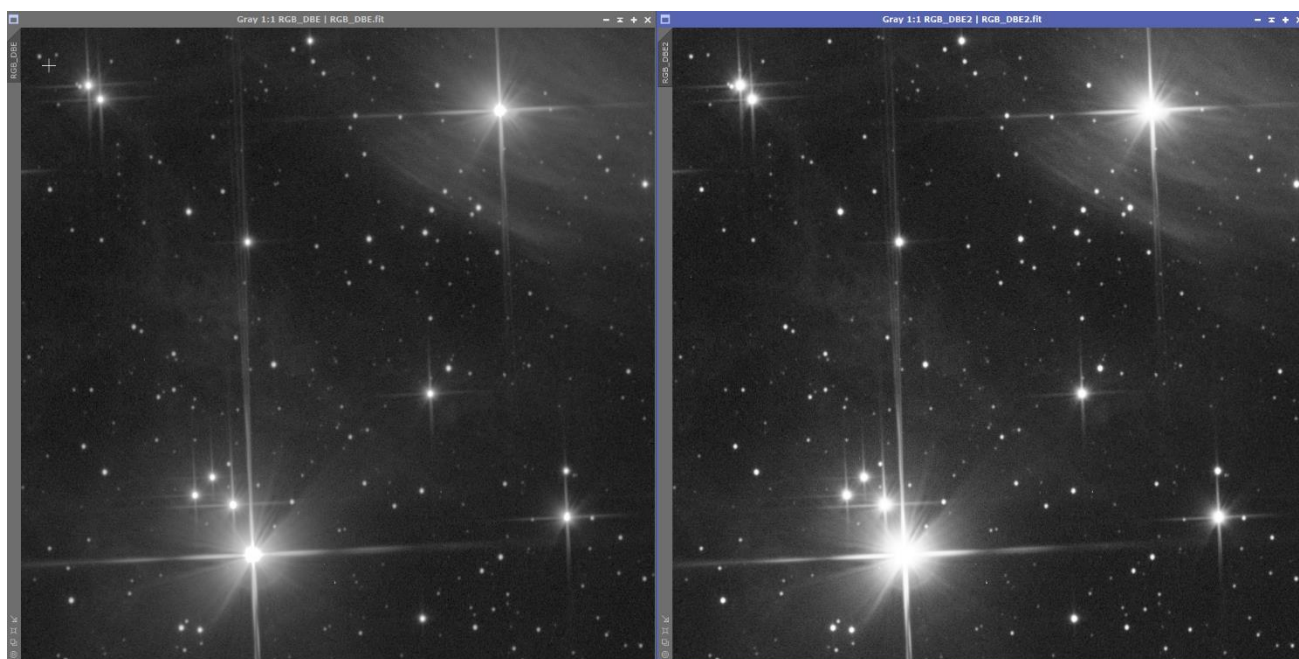
这里有个问题就是，在开始拉伸前我们可以做一个 Shadow Clip，具体操作我就不再多说了。

然后我这个是做了二十次，刚开始的时候误操作了一两次。看看 History。

#	Process Id	Start Time	Secs.	Mask
✓ 01	ConvertToGrayscale	2018 10月 11 12:27:49	0.16	
✓ 02	HistogramsTransform	2018 10月 11 12:27:57	0.45	
✓ 03	HistogramsTransform	2018 10月 11 12:28:14	0.52	Image02 <*broken*>
✓ 04	HistogramsTransform	2018 10月 11 12:28:24	0.43	
✓ 05	HistogramsTransform	2018 10月 11 12:28:53	0.53	Image05 <*broken*>
✓ 06	HistogramsTransform	2018 10月 11 12:29:10	0.53	Image06 <*broken*>
✓ 07	HistogramsTransform	2018 10月 11 12:29:25	0.52	Image07 <*broken*>
✓ 08	HistogramsTransform	2018 10月 11 12:29:38	0.53	Image08 <*broken*>
✓ 09	HistogramsTransform	2018 10月 11 12:29:58	0.53	Image09
✓ 10	HistogramsTransform	2018 10月 11 12:30:06	0.54	Image10
✓ 11	HistogramsTransform	2018 10月 11 12:30:15	0.51	Image11
✓ 12	HistogramsTransform	2018 10月 11 12:30:23	0.52	Image12
✓ 13	HistogramsTransform	2018 10月 11 12:30:36	0.54	Image13
✓ 14	HistogramsTransform	2018 10月 11 12:30:46	0.53	Image14 <*broken*>
✓ 15	HistogramsTransform	2018 10月 11 12:31:05	0.54	Image15
✓ 16	HistogramsTransform	2018 10月 11 12:31:14	0.52	Image16
✓ 17	HistogramsTransform	2018 10月 11 12:31:26	0.53	Image17
✓ 18	HistogramsTransform	2018 10月 11 12:31:35	0.52	Image18
✓ 19	HistogramsTransform	2018 10月 11 12:31:46	0.52	Image19
✓ 20	HistogramsTransform	2018 10月 11 12:32:05	0.61	Image20
✓ 21	HistogramsTransform	2018 10月 11 12:32:18	0.60	Image21

< Image01 > Report

结果对比如下图。我是拿 PILE 做的，为了对比效果放到 PI1.8 里面显示。



.....

工作量挺大的。

大吗？

PI1.8 一般是 100 次起步。

做的次数越多，结果越自然，Ringing 效应越弱。如果我们想提升我们在 PILE 里面做的次数，就必须降低拉伸力度，也就是说中间滑块要往中间靠而不是往左边靠。

还有一个问题就是这玩意对于星点凝聚度要求比较高，如果因为对焦失误或者视宁度差等原因，星点很弥散，Masked Stretch 出来的结果就会比较恶心。

其实我写的时候我觉得这玩意又会成为屠龙之技。首先适用范围不是很广，大部分情况下 Histogram Transfer 就能满足需求，其次，**这操作太麻烦了吧喂！**

就酱，当做一个有趣的路子吧。同时也说明了 PILE 的玩法非常非常多，欢迎大家把 PI1.8 的工具移植到 LE 上，或者是**开发全新的操作**。