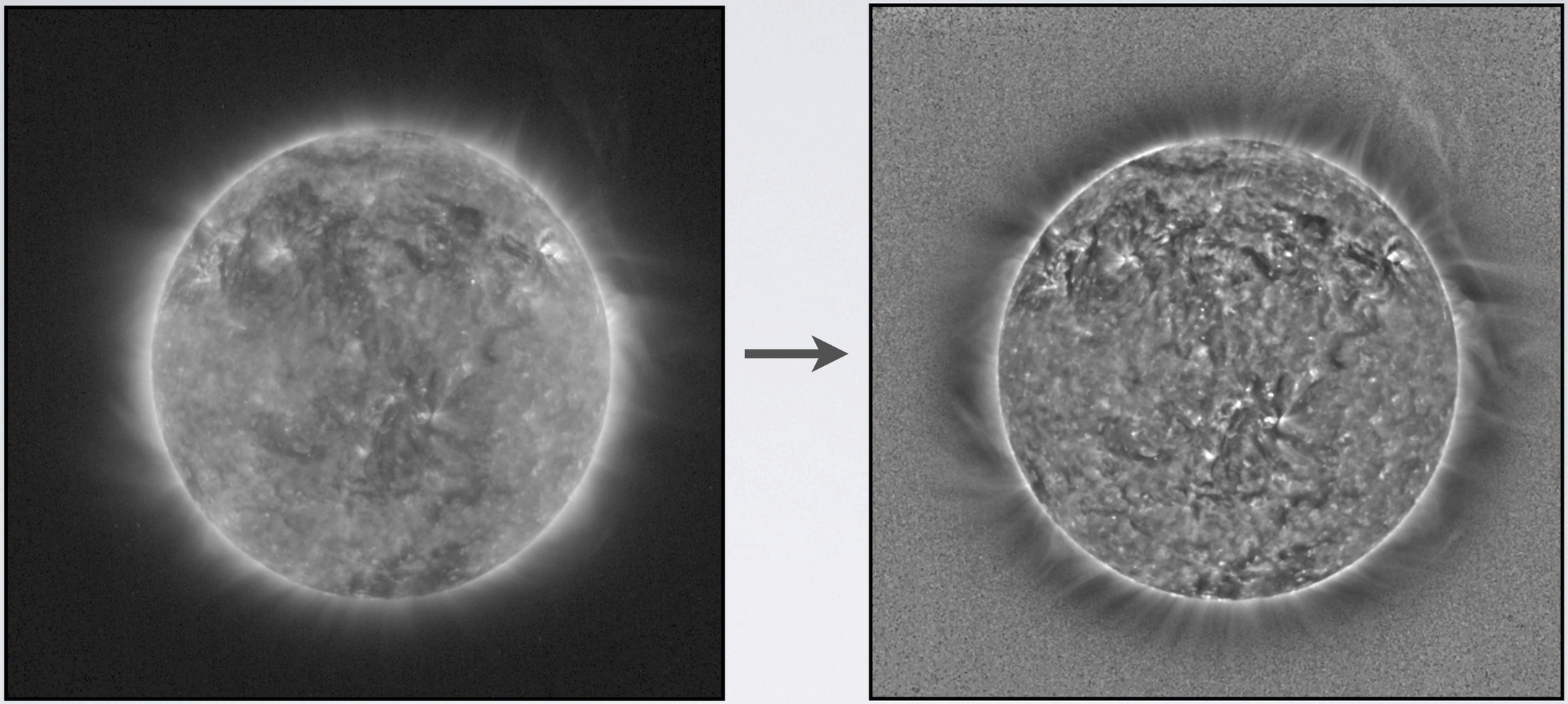


INTRODUCTION TO IMAGE PROCESSING

Dan Seaton, Royal Observatory of Belgium

```
IDL> image = readfits(<filename>, header)
```

IMAGE SCALING

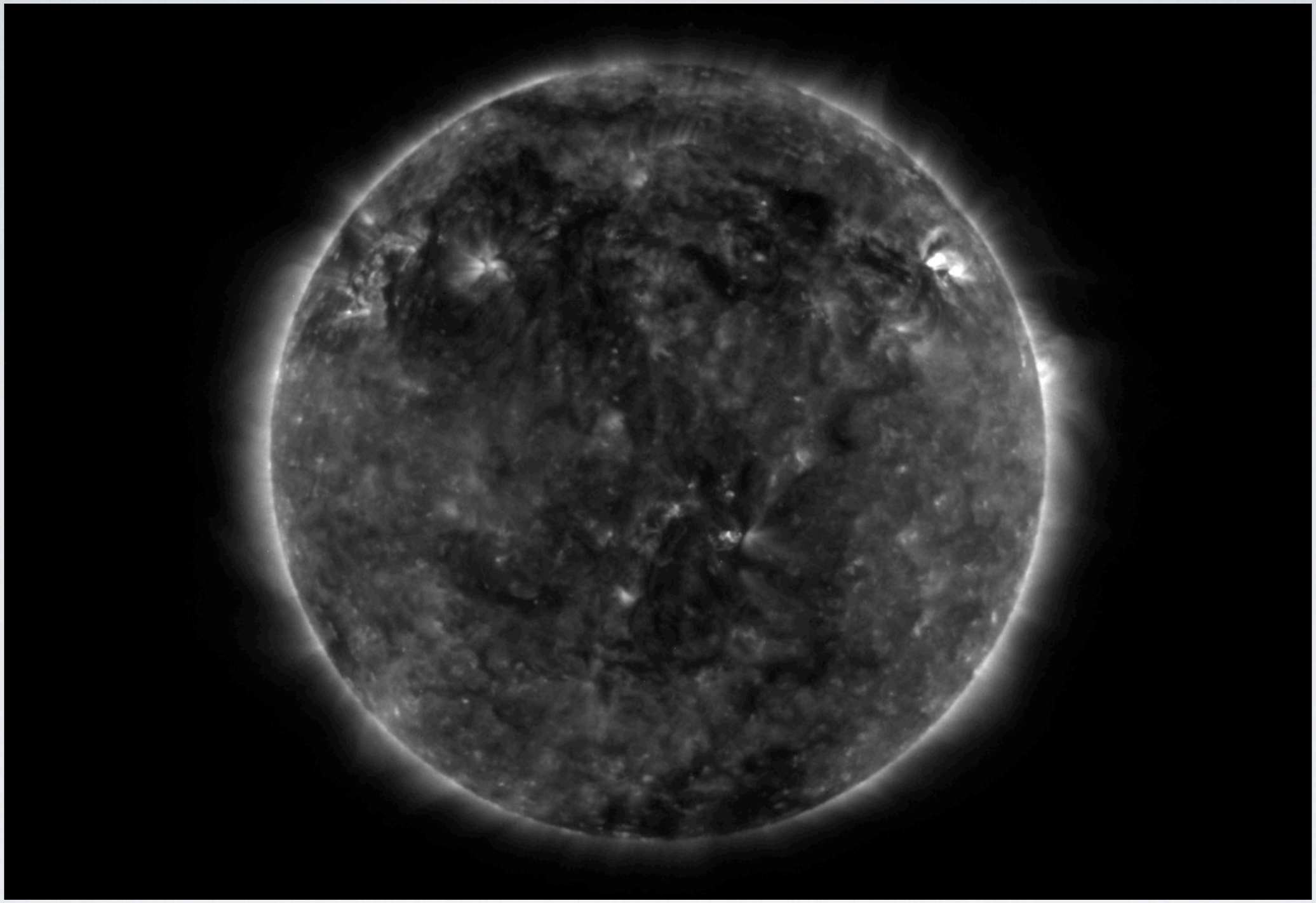


IMAGE SCALING

Linear

```
IDL> TVSCL, image
```

```
IDL> tv, bytscl(image, 10, 100)
```

Scales image from *data space* (many bits)
to *display space* (8 bits)

```
IDL> tv, bytscl(image, 10, 100)
```

```
IDL> tv, bytscl(image, 10, 100)
```

Cool Trick: To repeat the last line, use the ↑ key

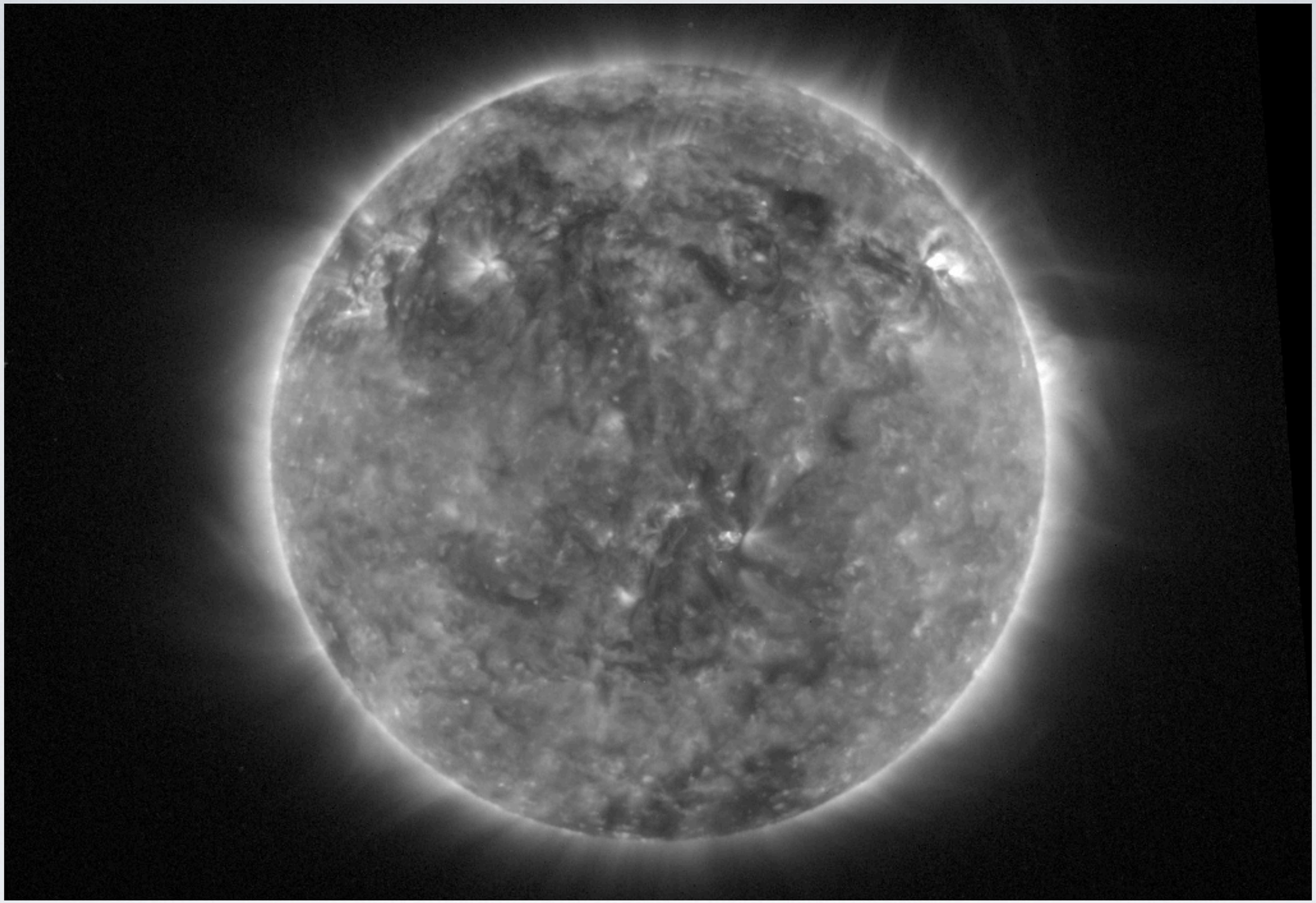


IMAGE SCALING

Square Root

```
IDL> sqrt_image = sqrt(image)
IDL> tv, bytscl(sqrt_image, 0, 15)
```

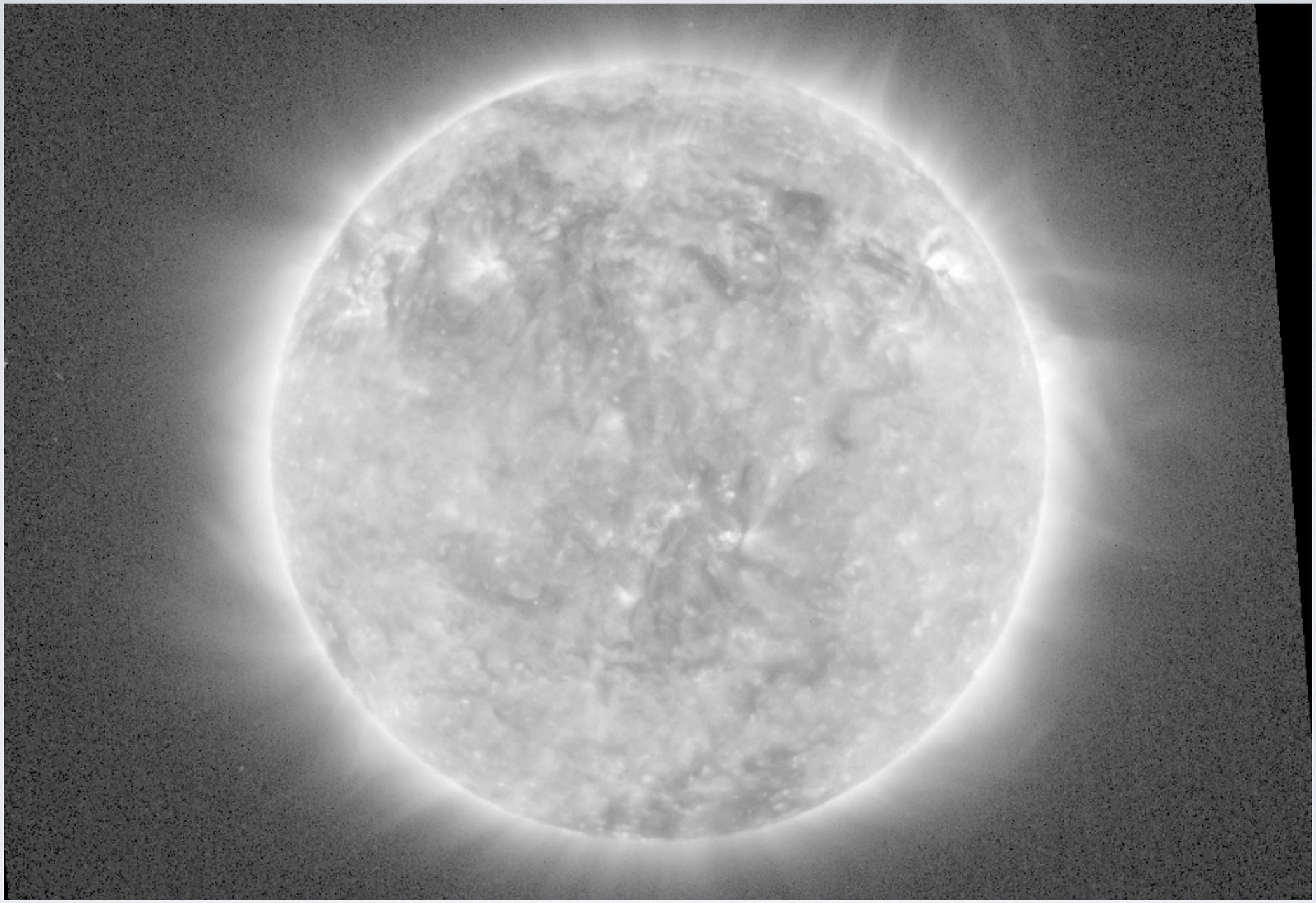


IMAGE SCALING

Logarithmic

```
IDL> log_image = alog10(image)
IDL> tv, bytscl(log_image, -2, 5)
```

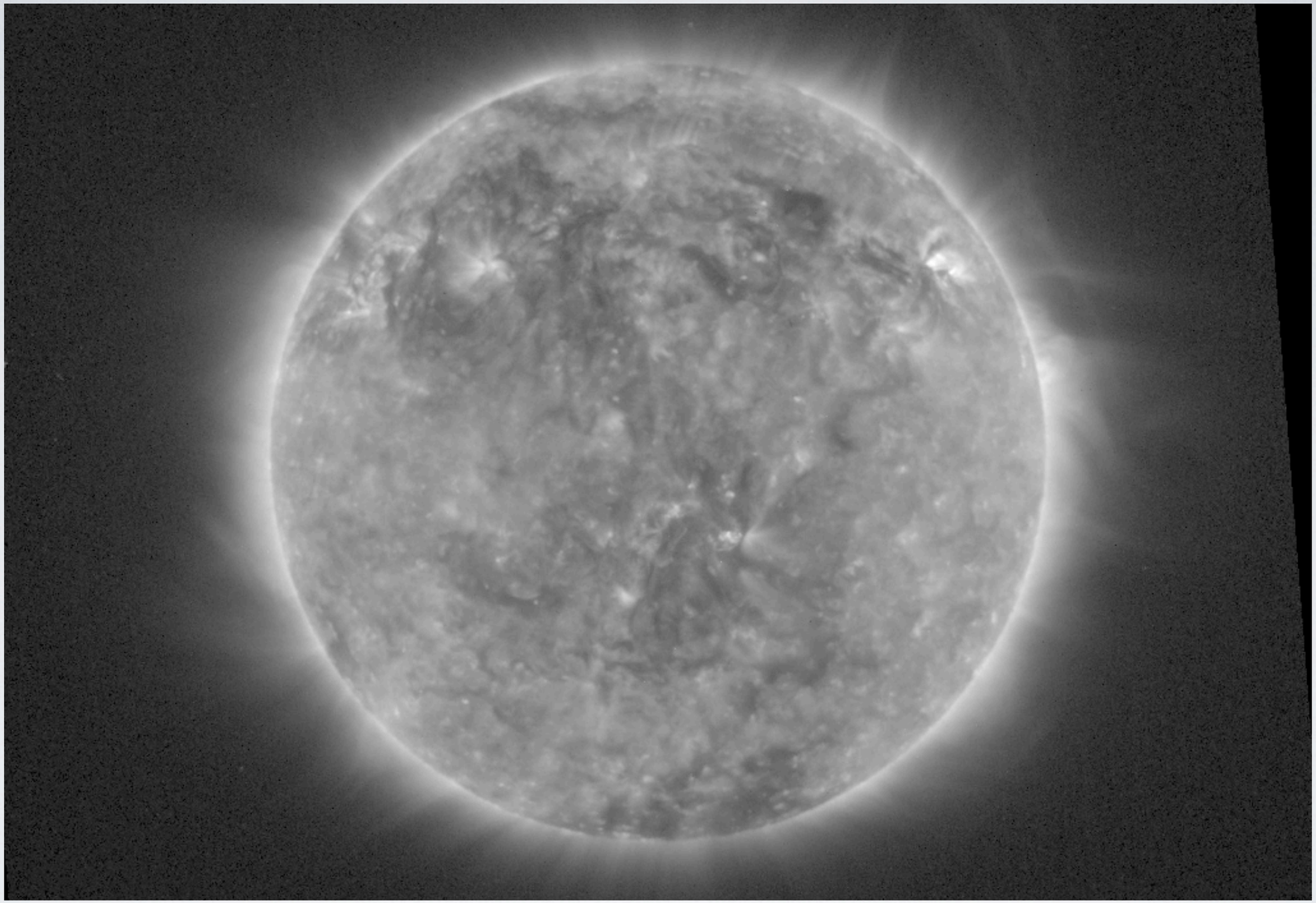


IMAGE SCALING

Other Nonlinear Scaling

Try some other functions for scaling

```
IDL> scaled_image = image^2
IDL> tv, bytscl(scaled_image, 0, 1000)
```

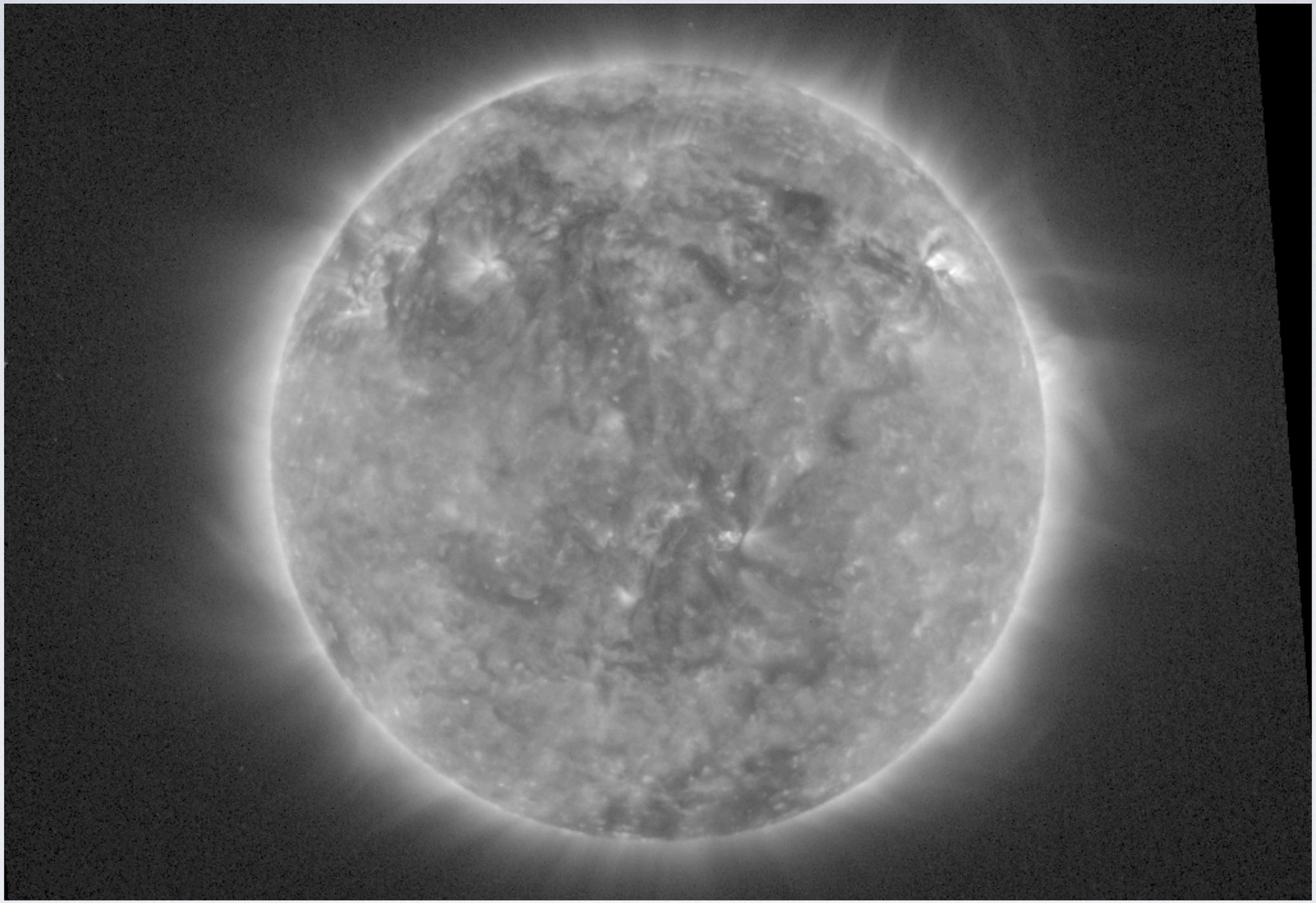
Try some other functions for scaling

```
IDL> scaled_image = image^2
IDL> tv, bytscl(scaled_image, 0, 1000)
```

Question:

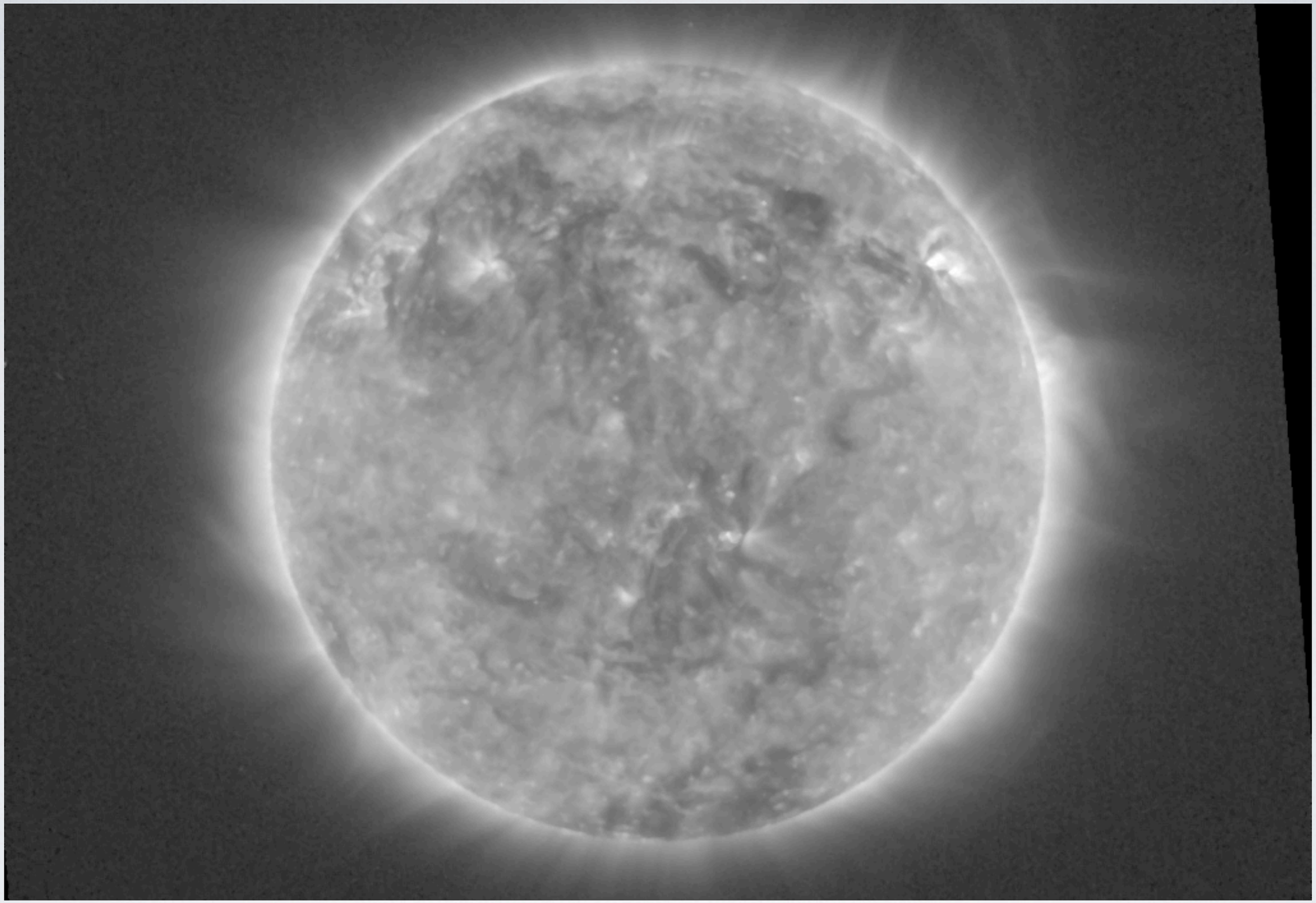
Why don't linear operations affect the appearance of the scaled image much?

SPATIAL FILTERS



UNFILTERED

```
IDL> tv, bytscl(sqrt(image), 0, 15)
```



MEDIAN FILTERED

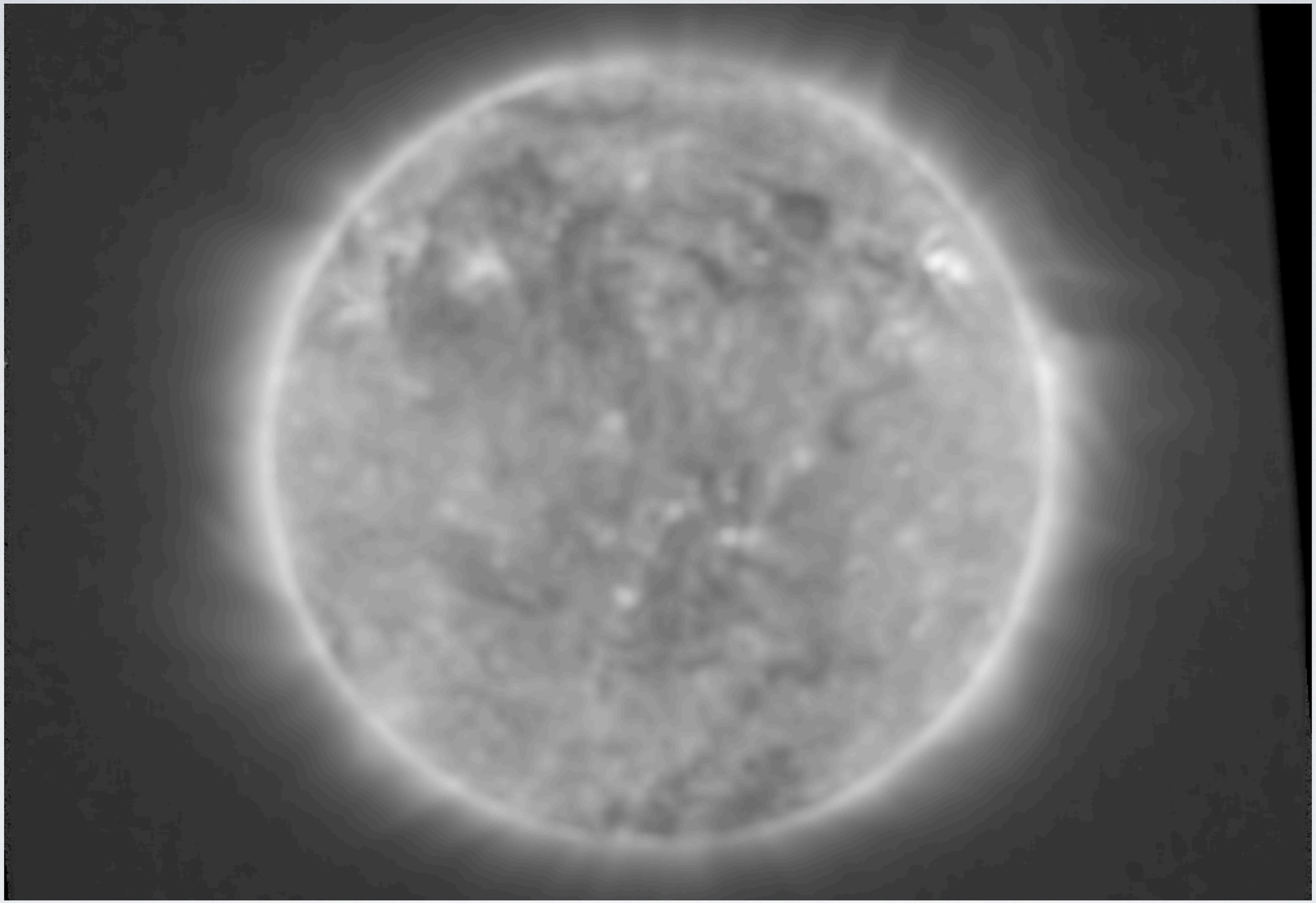
3×3 Pixel Filter (Removes Noise)

```
IDL> filtered_image = median(image, 3)
IDL> tv, bytscl(sqrt(filtered_image), 0, 15)
```

```
IDL> filtered_image = median(image, 3)
IDL> tv, bytscl(sqrt(filtered_image), 0, 15)
```

Question:

What happens when you vary the 3 in the *median* command?



SMOOTHED

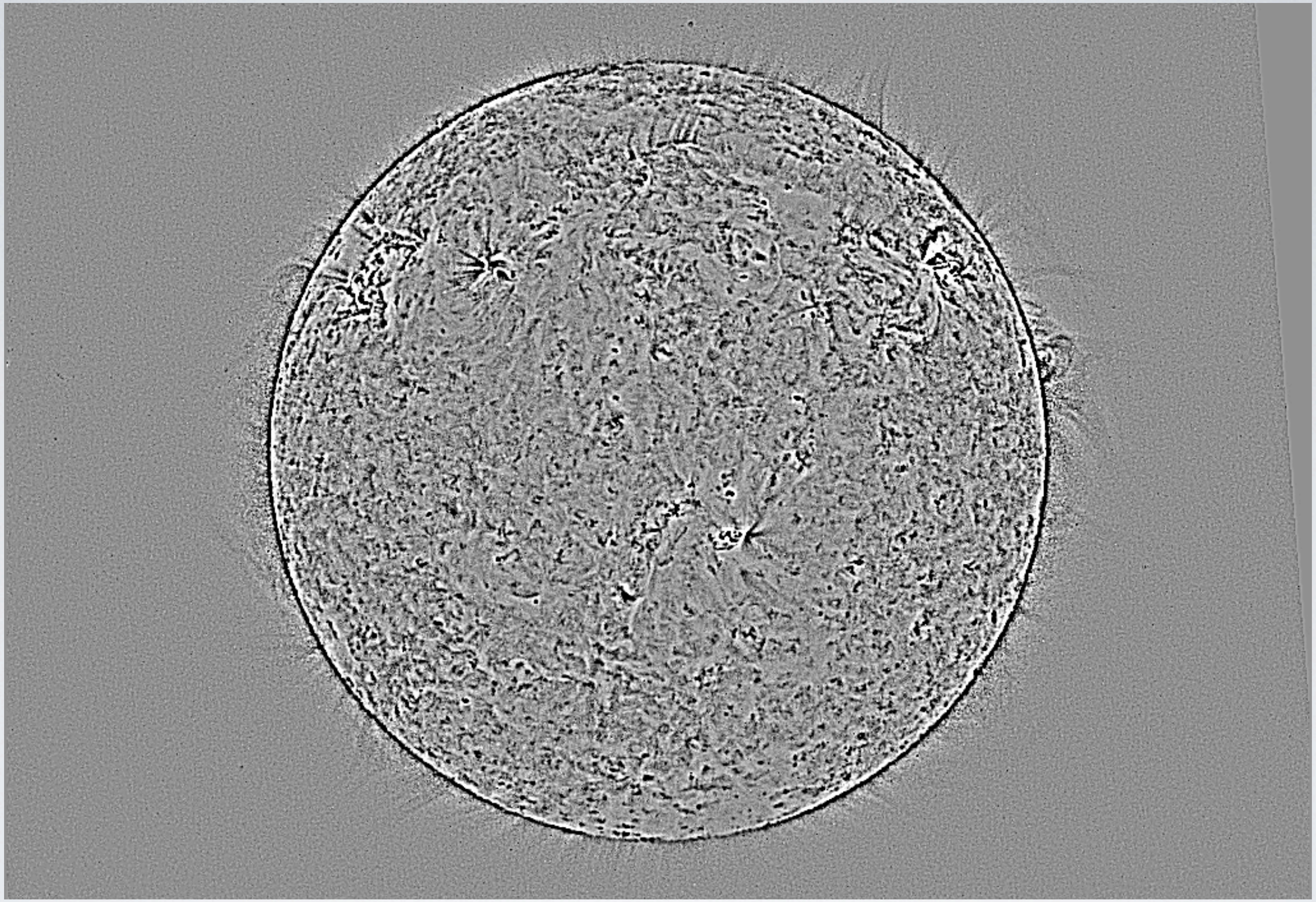
10×10 Pixel Boxcar (Low Frequencies)

```
IDL> filtered_image = smooth(image, 10)
IDL> tv, bytscl(sqrt(filtered_image), 0, 15)
```

```
IDL> filtered_image = smooth(image, 10)
IDL> tv, bytscl(sqrt(filtered_image), 0, 15)
```

Question:

How are *smooth* and *median* different?



HIGH FREQUENCIES

(Normal Image) - (Smoothed Image)

```
IDL> hifq_image = image - smooth(image, 10)
IDL> tv, bytscl(hifq_image, -10, 10)
```

Question:

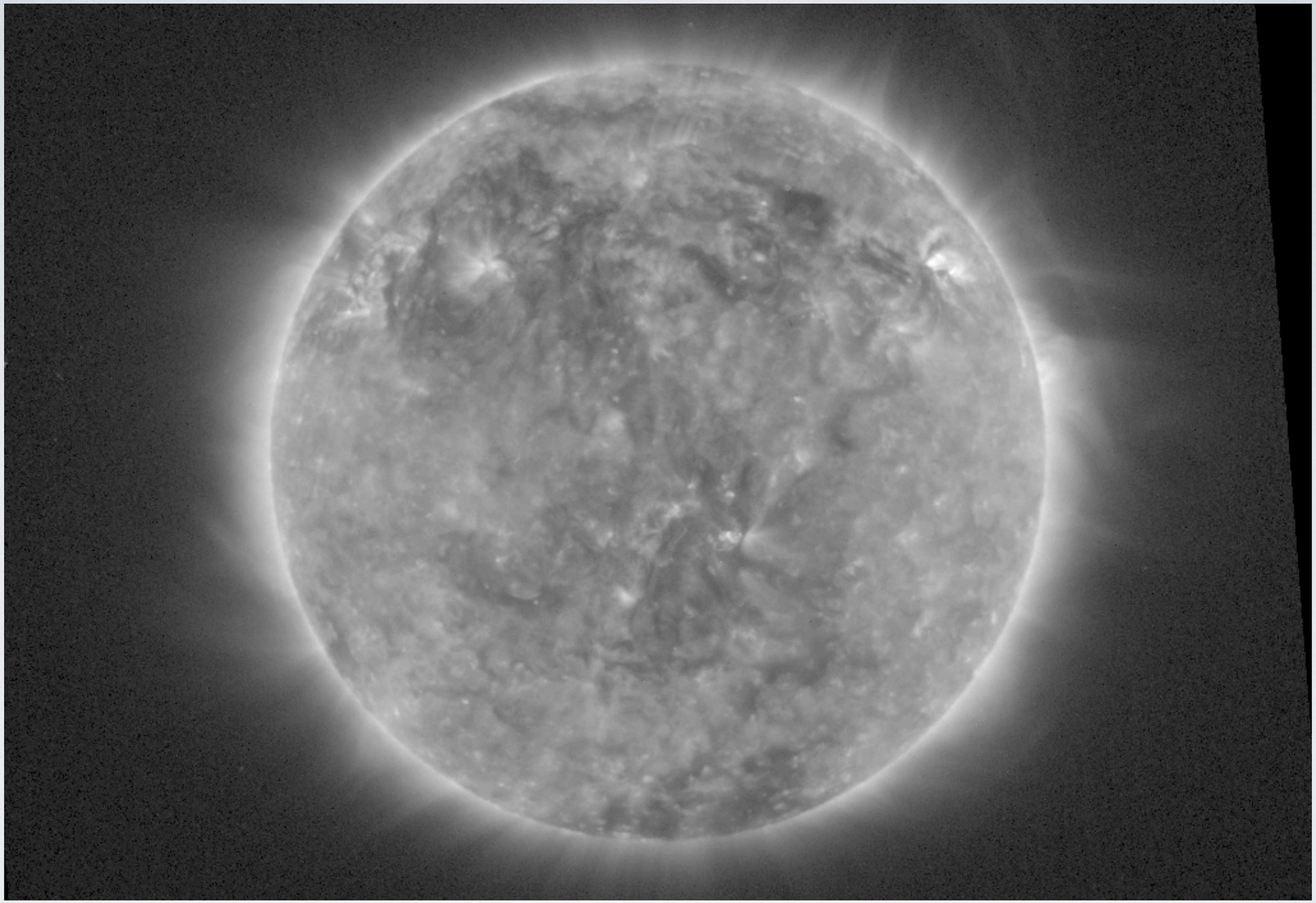
Why can't we take the *sqrt* of this image?

```
IDL> hifq_image = image - smooth(image, 10)
IDL> tv, bytscl(hifq_image, -10, 10)
```

```
IDL> hifq_image = image - smooth(image, 10)
IDL> tv, bytscl(hifq_image, -10, 10)
```

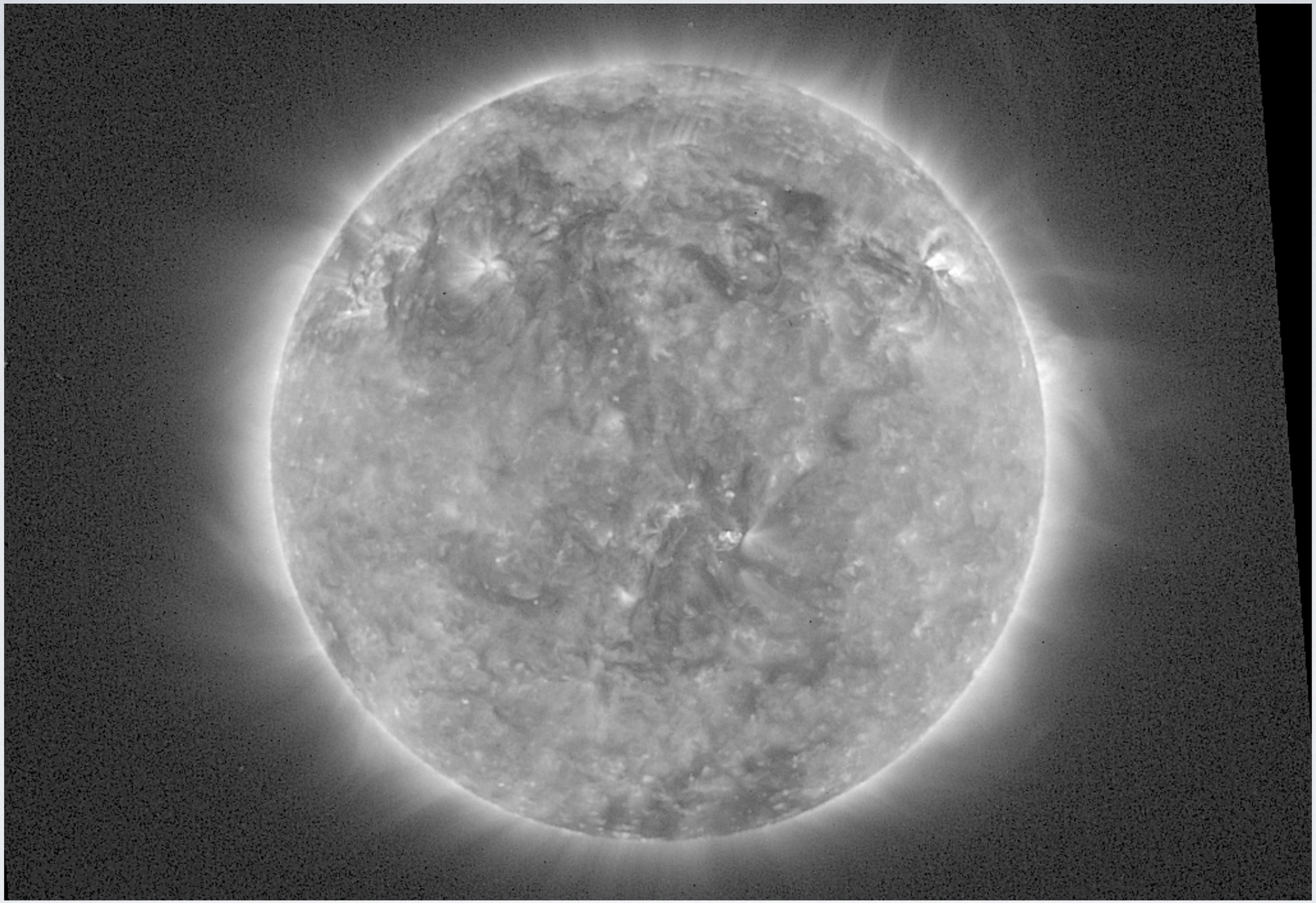
Question:

How does the 10 in *smooth* affect the appearance of the image you display?



NEXT: UNSHARP MASKING

Amplify High Frequencies



UNSHARP MASKED

(High Frequencies) × Const. + (Normal Image)

```
IDL> hifq_image = image - smooth(image, 10)
IDL> sharp_image = image + hifq_image
IDL> tv, bytscl(sqrt(sharp_image), 0, 15)
```

```
IDL> hifq_image = image - smooth(image, 10)
IDL> sharp_image = image + hifq_image
IDL> tv, bytscl(sqrt(sharp_image), 0, 15)
```

Question:

How does the 10 in *smooth* affect the appearance of the image you display?

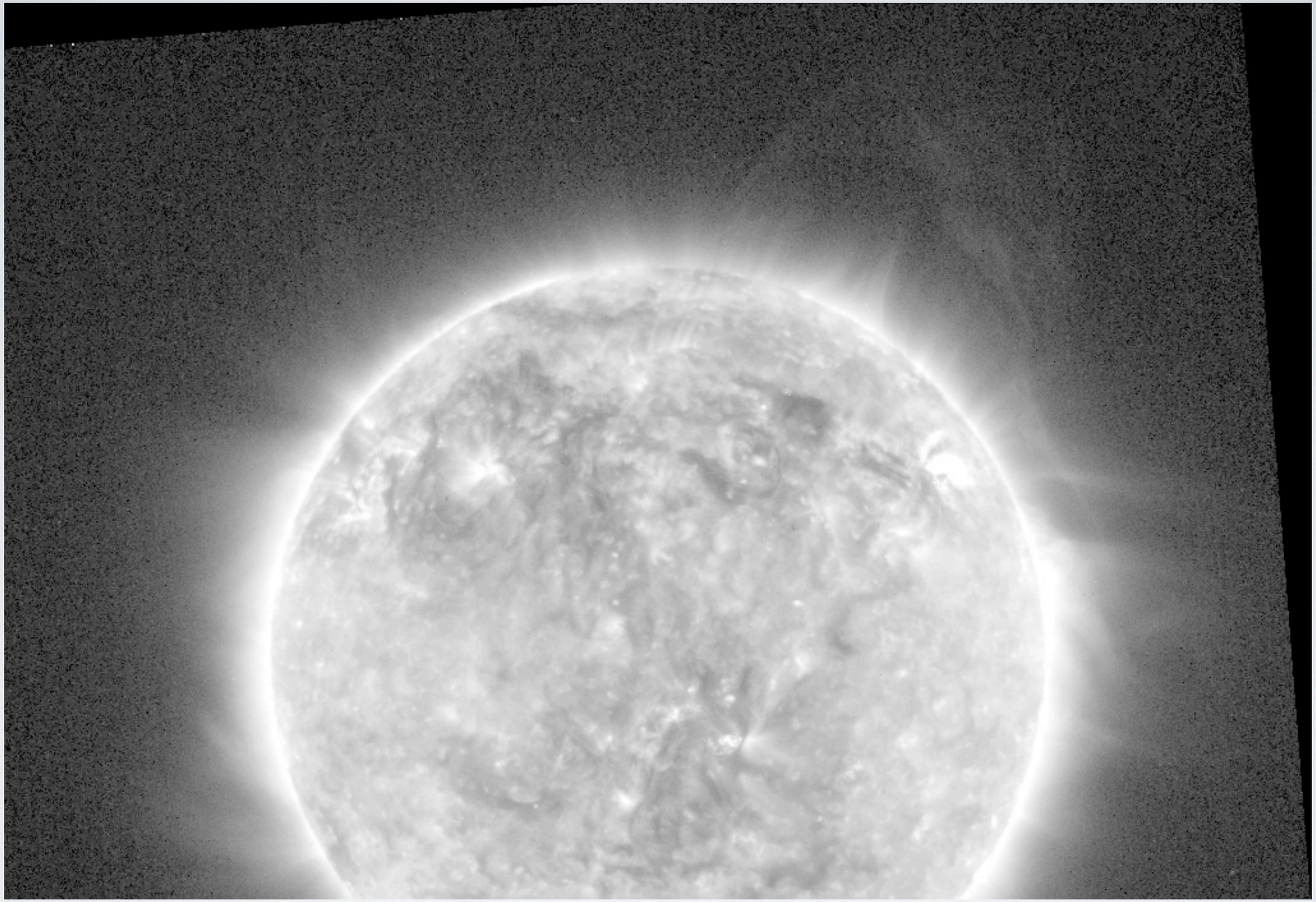
Advanced Sharpening:
Adjust the amount of sharpening by adding a *constant* in front of the *high frequency components*.

```
IDL> hifq_image = image - smooth(image, 10)
IDL> sharp_image = image + 0.5 * hifq_image
IDL> tv, bytscl(sqrt(sharp_image), 0, 15)
```

Advanced Sharpening:
Adjust the amount of sharpening by adding a *constant* in front of the *high frequency components*.

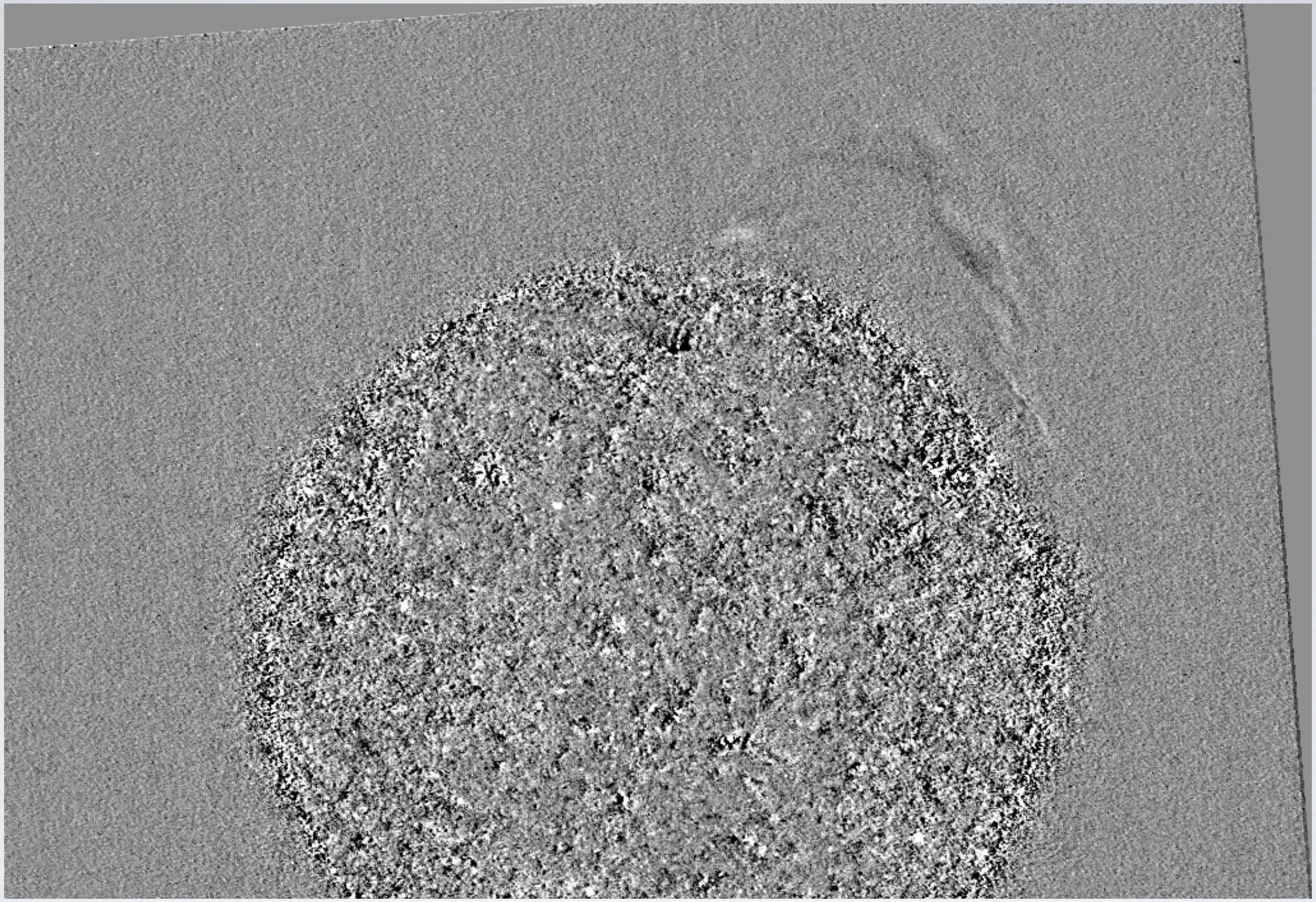
```
IDL> hifq_image = image - smooth(image, 10)
IDL> sharp_image = image + 0.5 * hifq_image
IDL> tv, bytscl(sqrt(sharp_image), 0, 15)
```

Question:
How does the constant affect the result?



NEXT: RUNNING DIFFERENCE

Highlights Dynamic Features



RUNNING DIFFERENCE

(Current Frame) - (Previous Frame)

Get Two (or more) Sequential Images:

http://proba2.oma.be/swap/level1/2010/10/19/

```
IDL> image1 = readfits(<file1>, header1)
IDL> image2 = readfits(<file2>, header2)
IDL> diff_image = image2 - image1
IDL> tv, bytscl(diff_image, -10, 10)
```

Get Two (or more) Sequential Images:

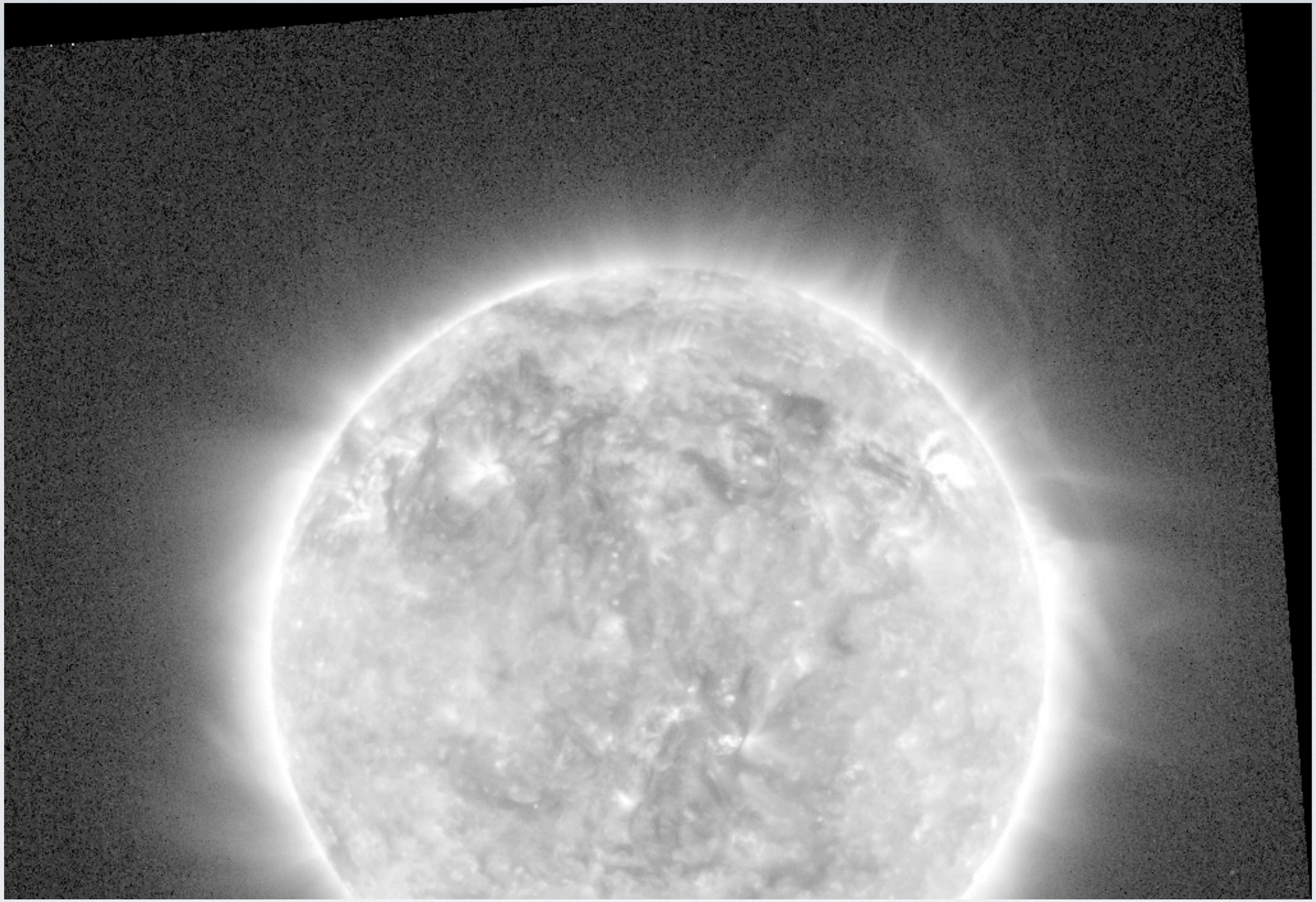
http://proba2.oma.be/swap/level1/2010/10/19/

```
IDL> image1 = readfits(<file1>, header1)
IDL> image2 = readfits(<file2>, header2)
IDL> diff_image = image2 - image1
IDL> tv, bytscl(diff_image, -10, 10)
```

Question:

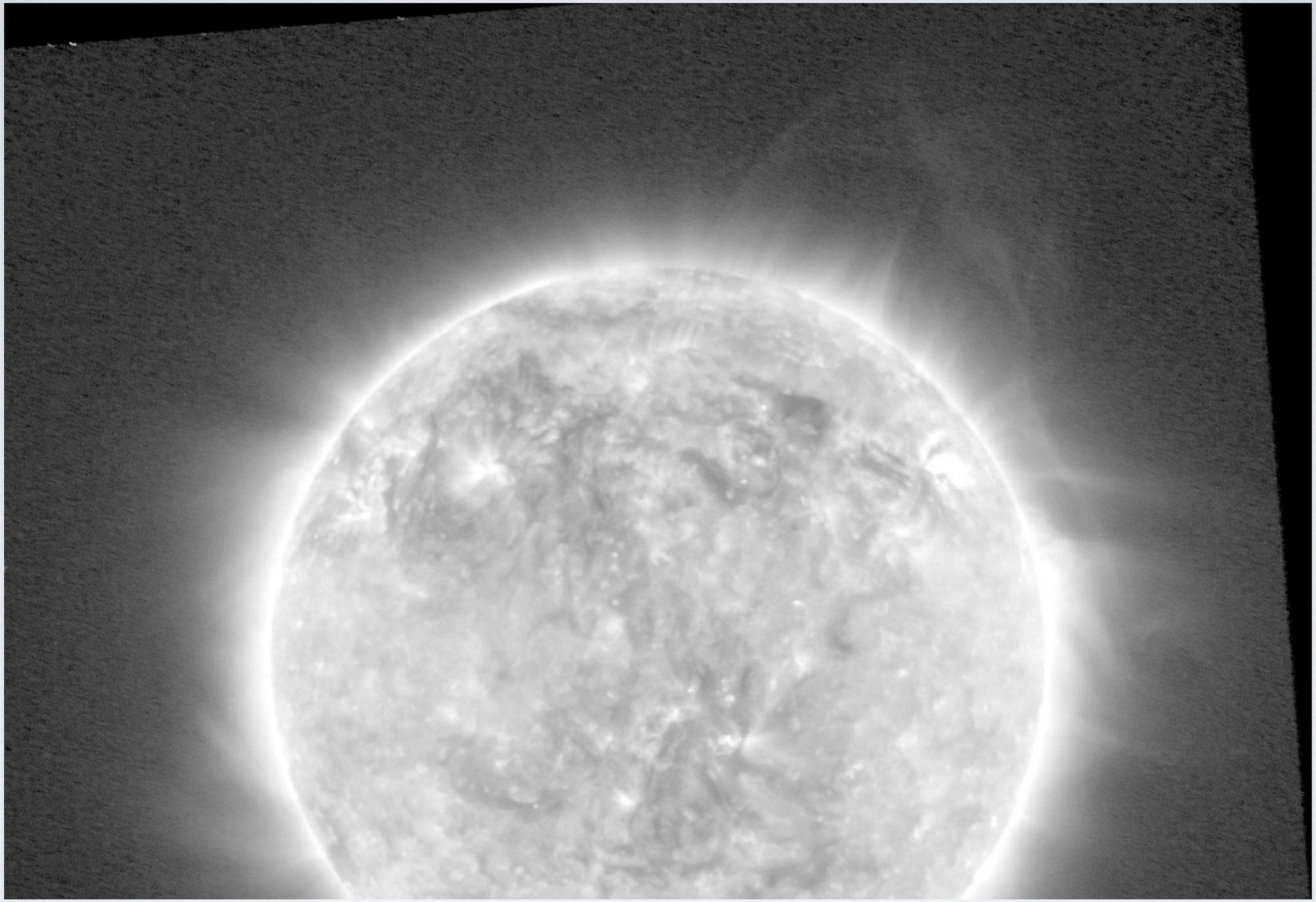
How does the result change when time between the two images increases?

ADVANCED TECHNIQUES



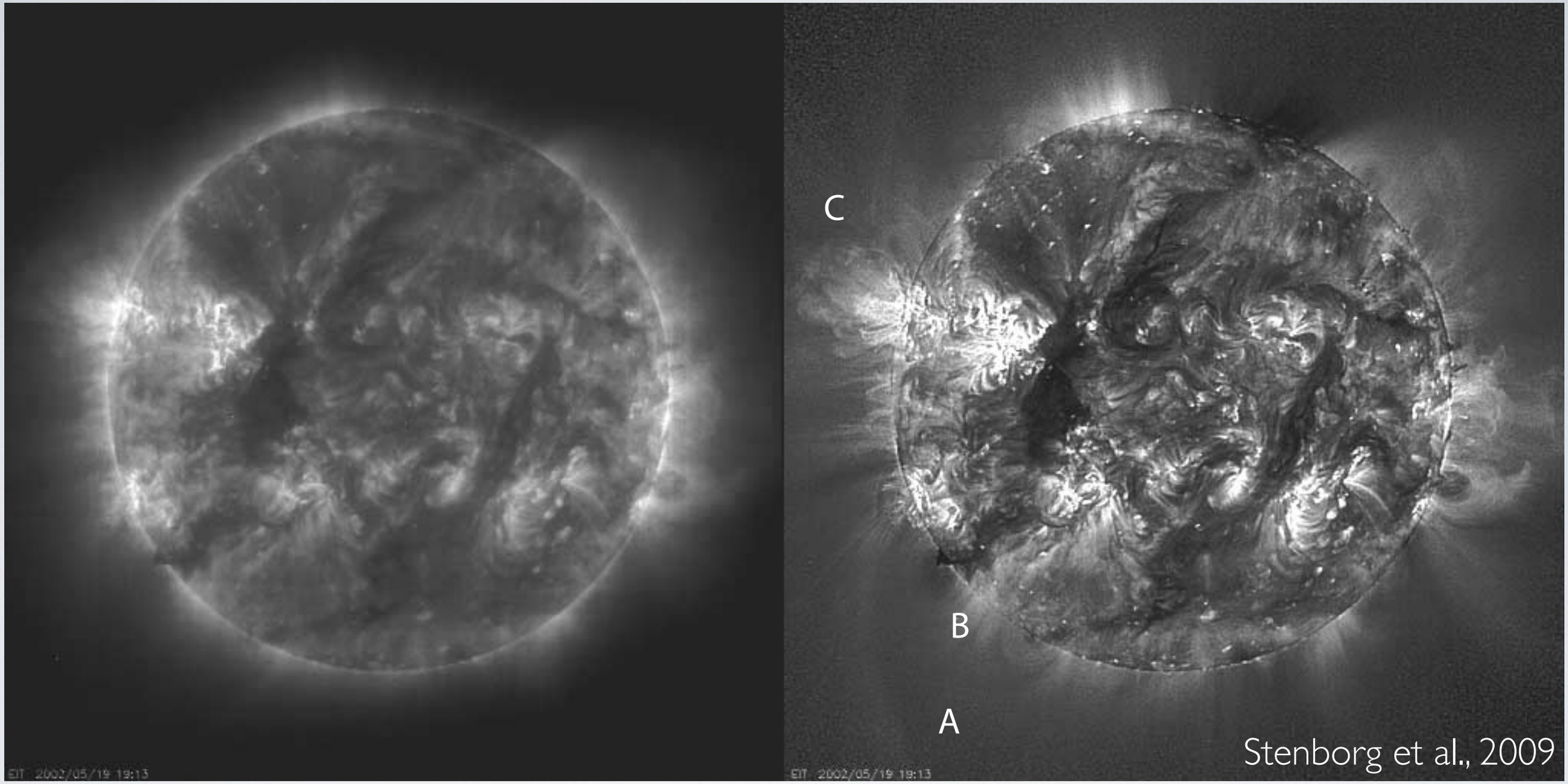
NEXT: SUMMING IMAGES

Improves Signal to Noise



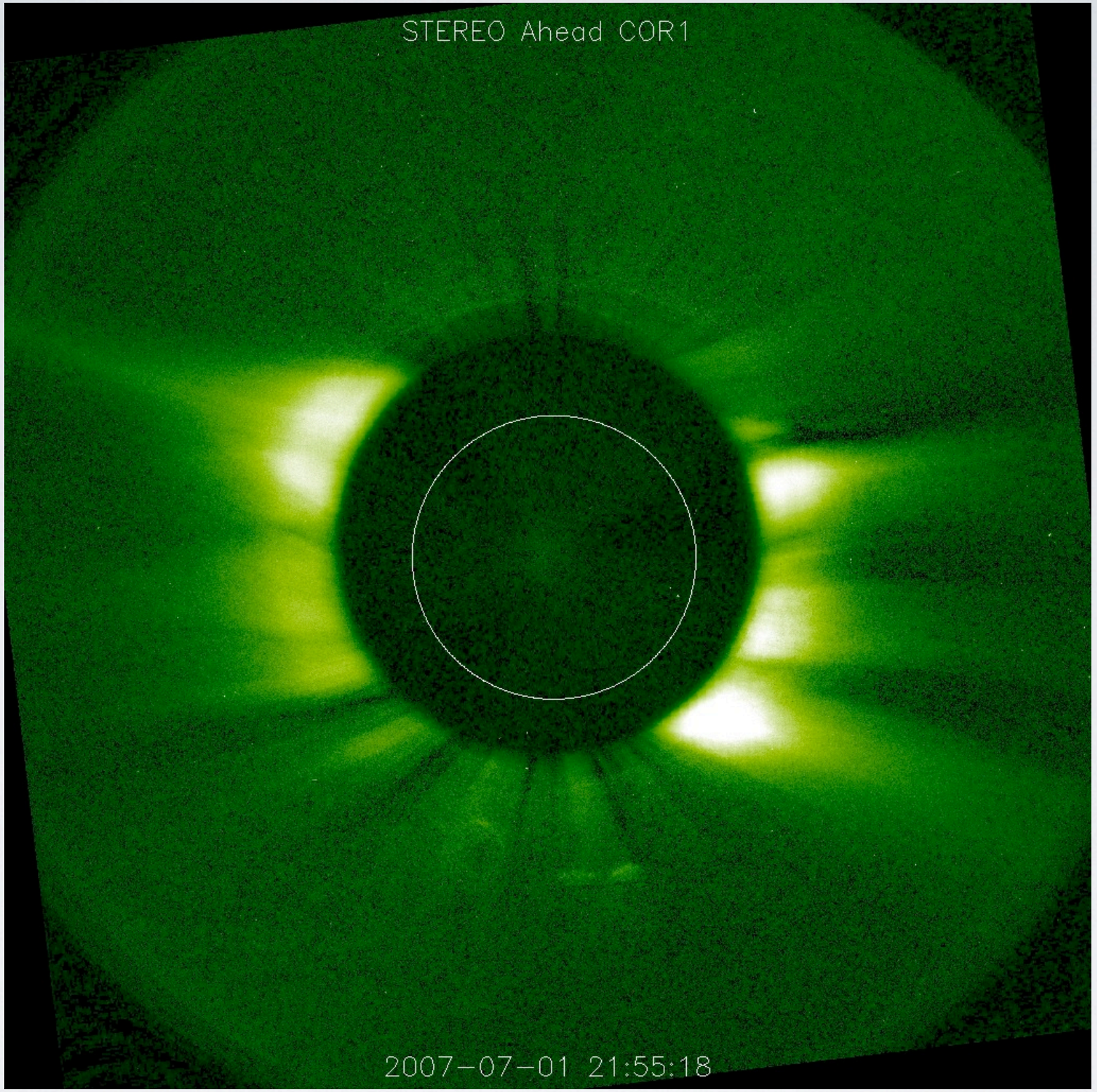
SUMMED IMAGES

5 Frames Added



WAVELET FILTER

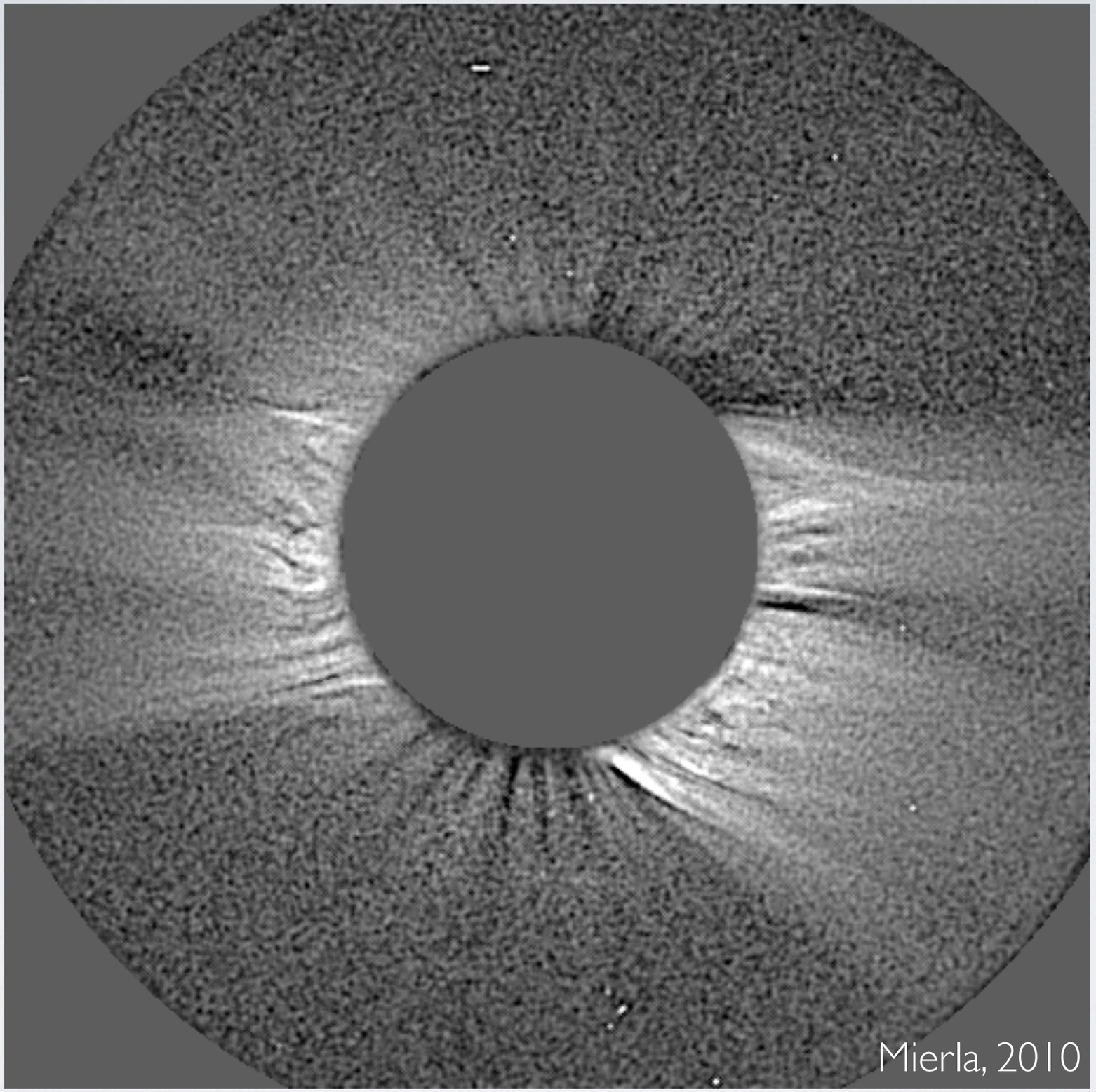
Multiscale Filter, Enhances Coherent Structures



WAVELET FILTER

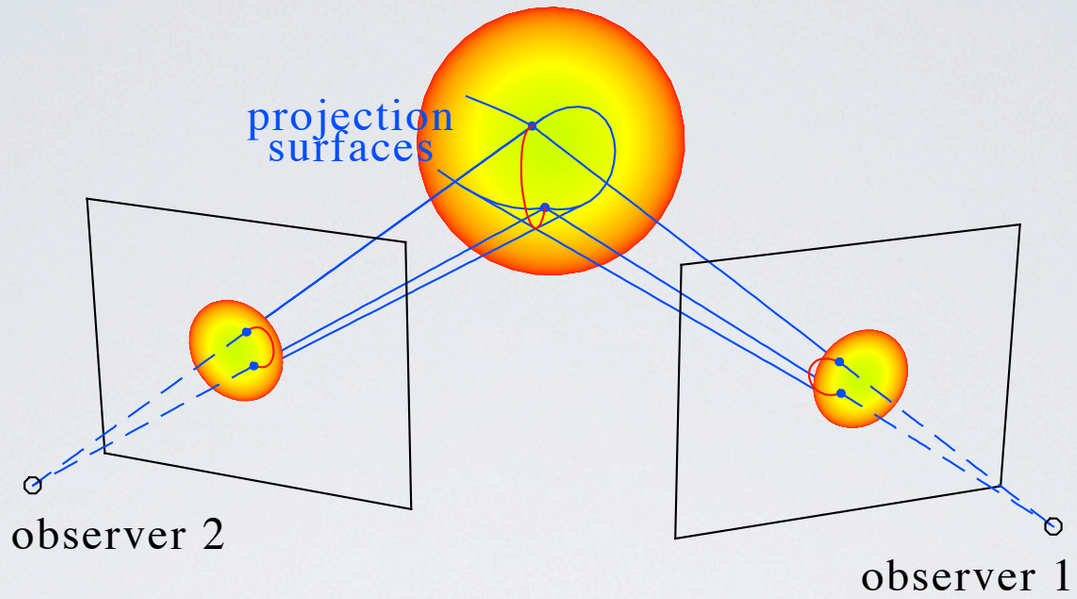
Mierla, 2010

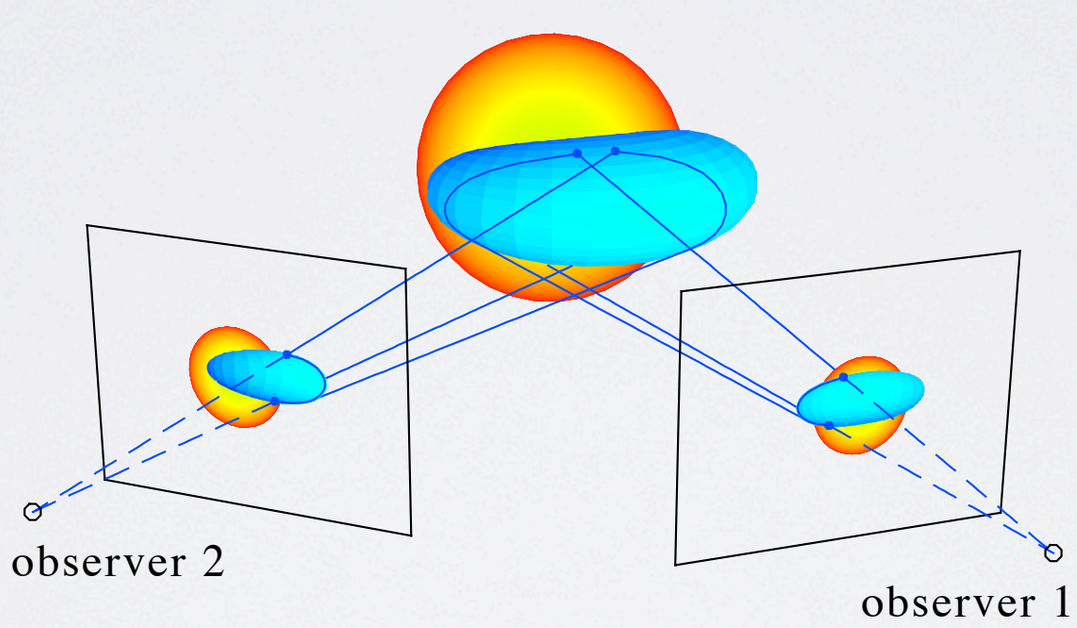
WAVELET FILTER



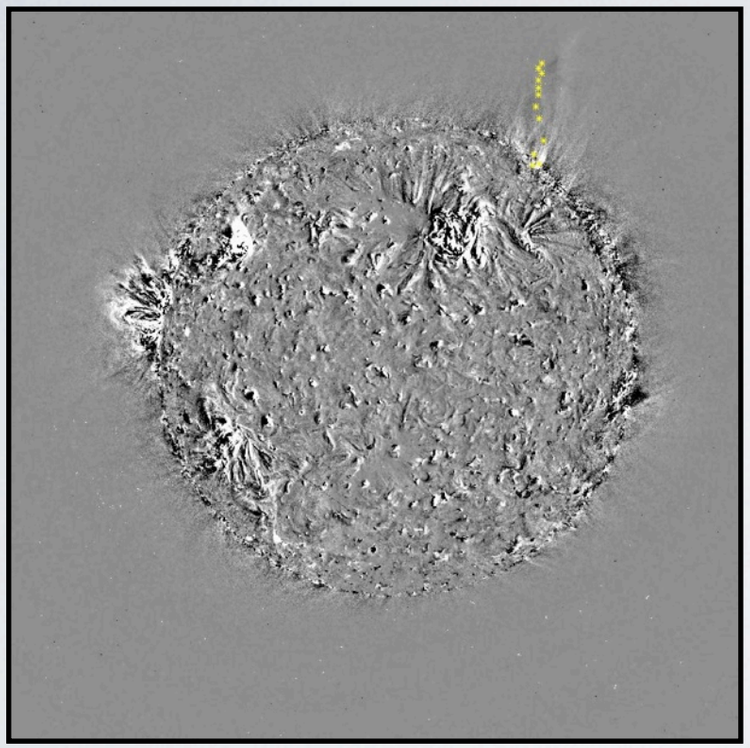
WAVELET FILTER

Inhester., 2007

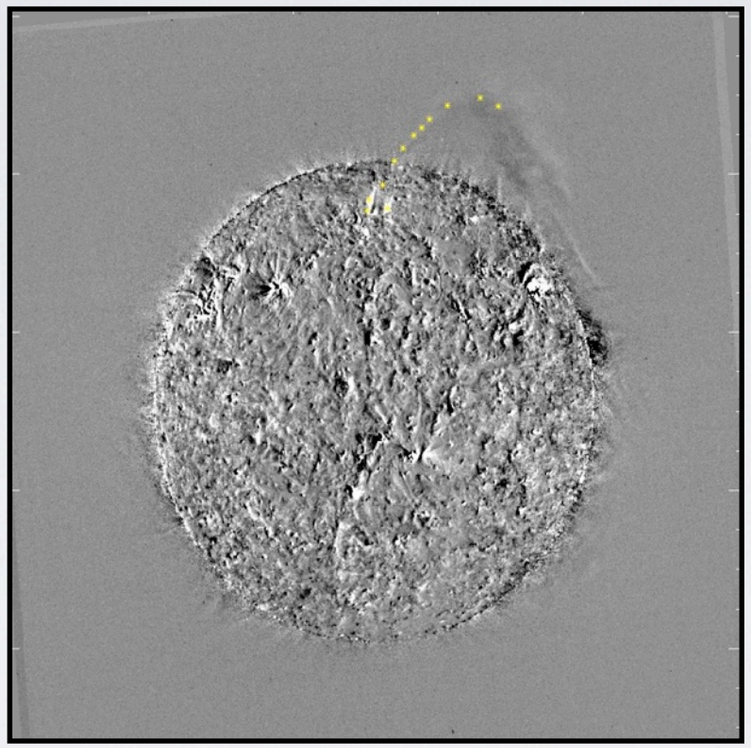




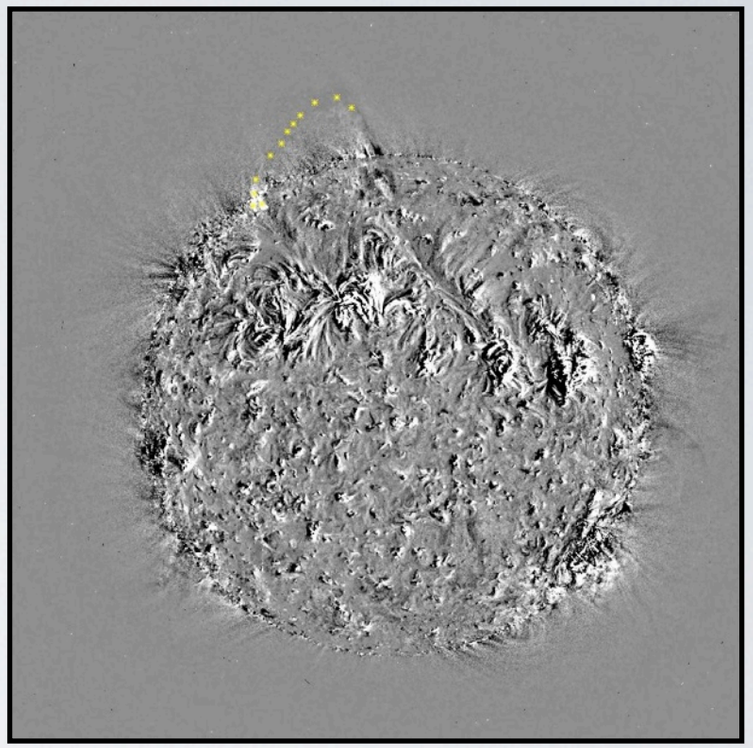
3D RECONSTRUCTIONS



STEREO B



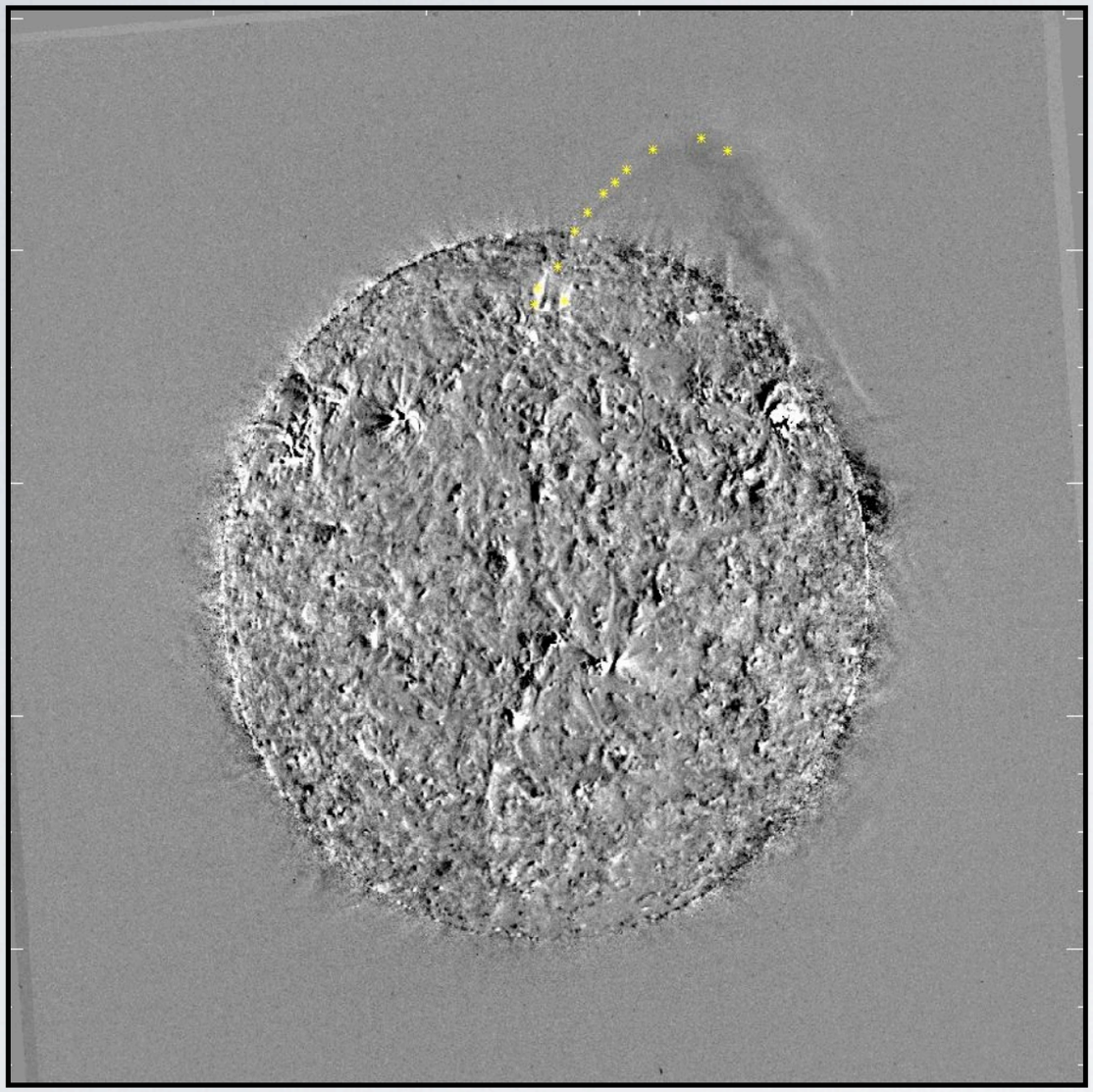
SWAP

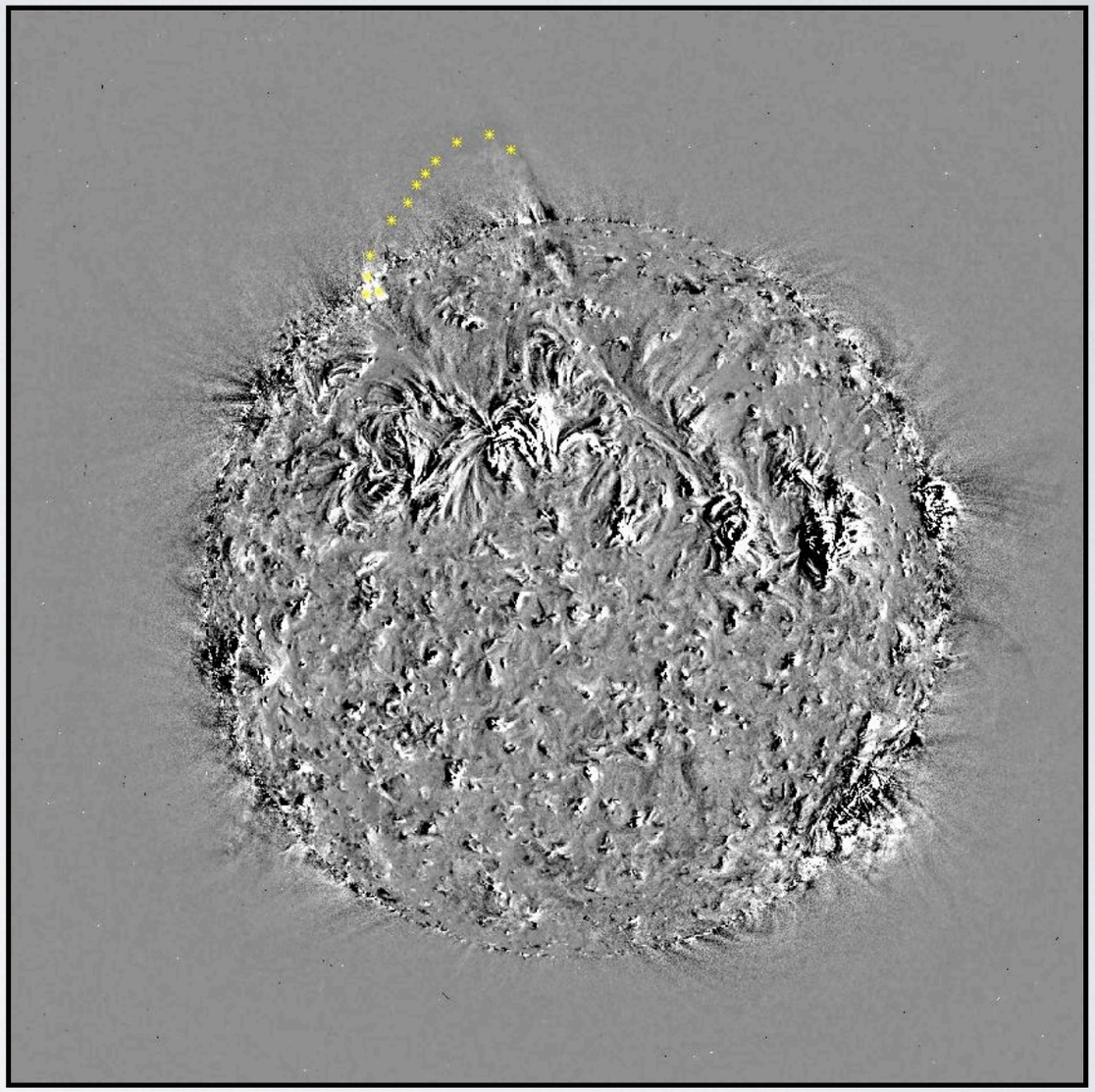


STEREO A

3D RECONSTRUCTIONS

Combining SWAP & STEREO Views

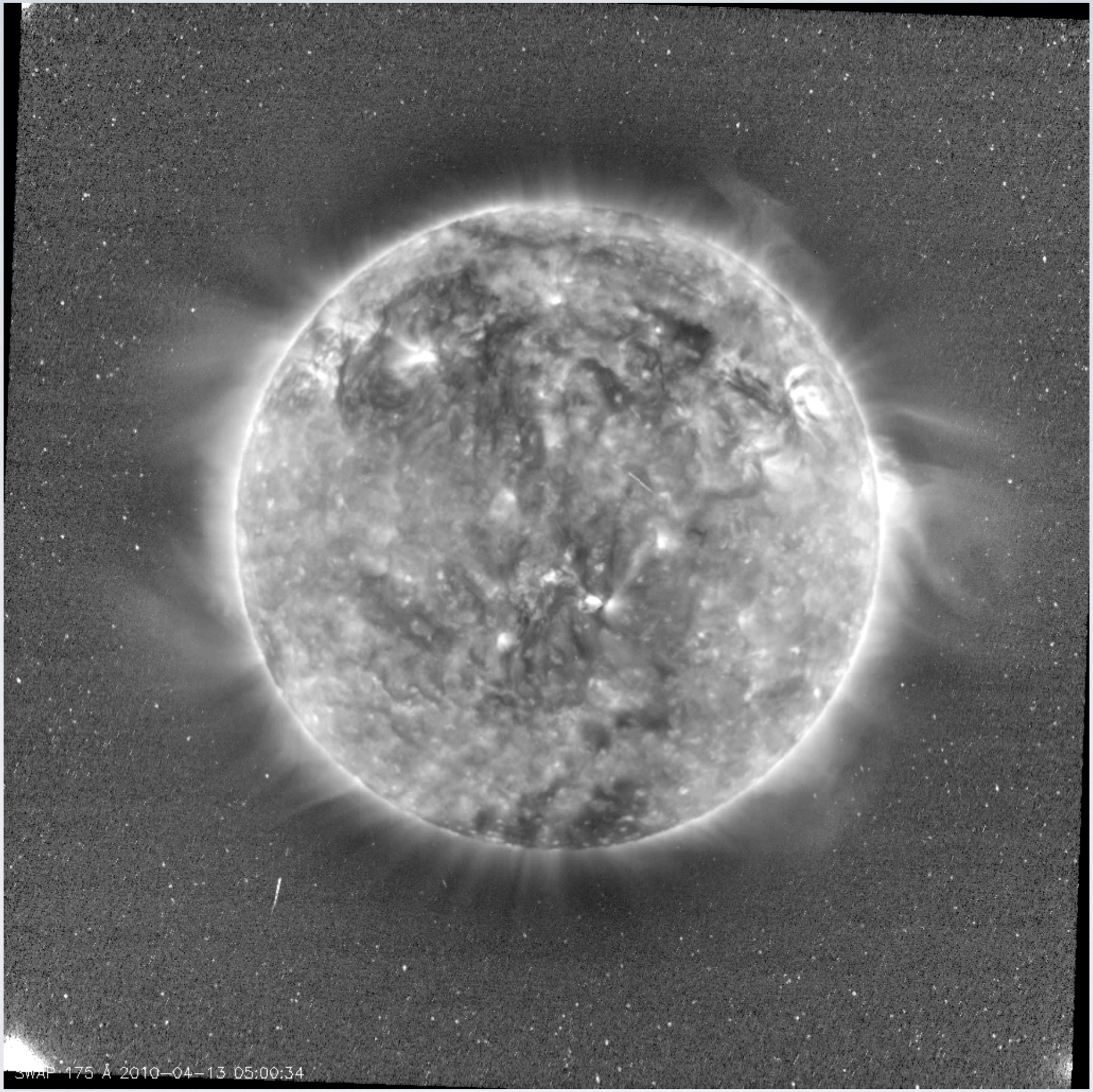




3D RECONSTRUCTIONS

Combining SWAP & STEREO Views

RADIAL FILTERING


SWAP 175 Å 2010-04-13 05:00:34

NOW IT'S YOUR TURN...

Task:

Using any of the techniques we've discussed produce a result that reveals something you can't see in the linearly scaled image.

```
IDL> sqrt_image = sqrt(image)
IDL> log_image = alog10(imge)
IDL> filtered_image = median(image, 3)
IDL> hifq_image = image - smooth(image, 10)
IDL> sharp_image = image + hifq_image
IDL> diff_image = image2 - image1
```

SUPER-ADVANCED DATA ANALYSIS & MANIPULATION

http://grian.phy.tcd.ie/solarmonitor/objects/swap/