IMAGE FUSION

HDR TONEMAPPING

Theoretical background

Image fusion is intended to create an image with information from multiple source images, when each individual source image does not yield an individually good result. Usually, these algorithms work with 3 images, and some cases process up to 7. The given algorithm enables the user to flexibly choose how many source images to allocate, from 3 to 7.

For each pixel of the final image, the algorithm calculates the cost function for the corresponding pixel of the source images. This is done in the luminance track, using a 5x5 slider around each pixel (i, j) to find the amount of information in that area in the given source image.

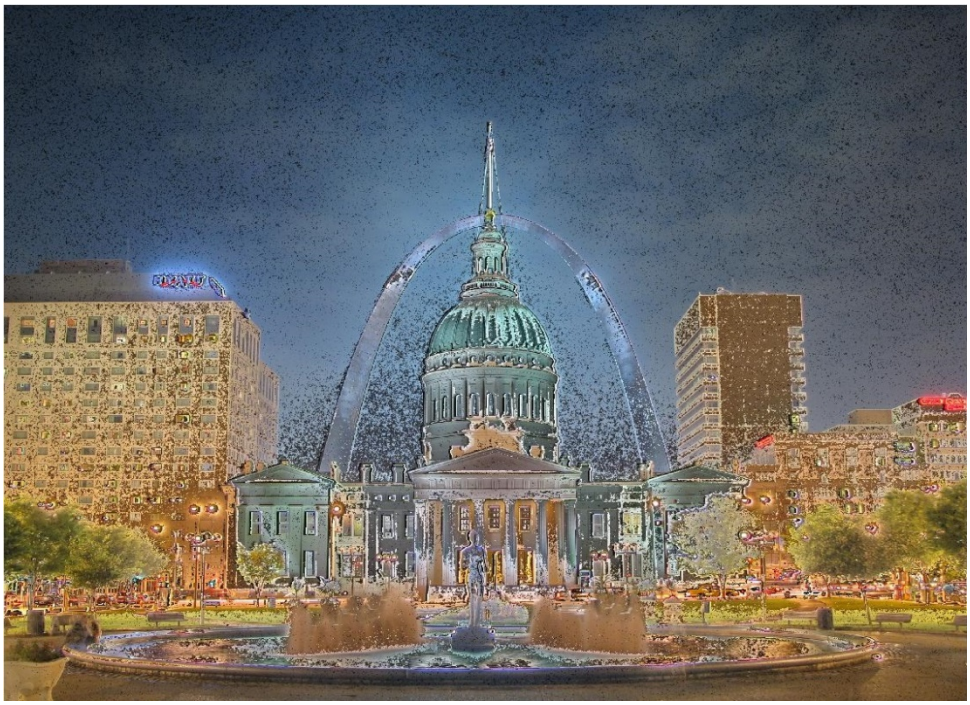
The fitness function is defined as:

$$\boldsymbol{fit}_{\boldsymbol{function}}(\boldsymbol{i},\boldsymbol{j}) = \frac{(\max(\boldsymbol{window}) - \min(\boldsymbol{window})) * \boldsymbol{std}(\boldsymbol{window})}{\left(128 - \boldsymbol{mean}(\boldsymbol{window})\right) * \left(128 - \boldsymbol{Image}(\boldsymbol{i},\boldsymbol{j})\right) * (\boldsymbol{Image}(\boldsymbol{i},\boldsymbol{j}) - \boldsymbol{mean}(\boldsymbol{window}))}$$

The source image displaying the highest value in the fitness function is selected, and the image number (1-7) is stored in the corresponding pixel in a new table. At the end of the process, therefore, we have a table as large as the final image, with integers ranging from 1 to 7 for each pixel. If fusion were done now, there would be a clear separation between the areas, resulting in a bad visual effect.

4	4	4	4
4	4	4	4
4	4	4	3
4	4	3	4
3	3	3	4
3	4	3	3

If we depicted the picture now, without any smoothness of transitions, the result is the following:



We notice that there are too many spots with different source tables, resulting in the appearance of salt and pepper noise. If we try to group source tables, we end up with color banding.

This is because although noise is excluded in each class, the classes remain crisp / clearly defined, so the source table contains singularities.

Therefore, for transitional smoothness between regions in the final image, we pass the table of appropriate sources, by Gaussian filter. This results in a smooth transition between regions. Each pixel of the final image is taken as a linear combination of the two pixels of the two source images at this position.

3.00404101949461	3.00232234572645	3.00056530188925	2.99874638128908
3.00285062987502	3.00111510584591	2.99934098769571	2.99750519986847
3.00160600940984	2.99985367362305	2.99806253092049	2.99620993537460
3.00036893010475	2.99860036982851	2.99679278194433	2.99492394803161
2.99915533486490	2.99737026104202	2.99554592297103	2.99366052836677

2.99795165897173 2.99614926303577 2.99430737023850 2.99240461307234

E.g. if the source table at one position had a value of 2.7, then the pixel of the final image at the same location would be 0.7 at the pixel value of source image 3, and 0.3 at the pixel value of source image 2. Like the result is a smooth transition between areas of brightness, just as the human eye perceives it.

For example, the picture for maximum information is as follows:

The final source-image table contains, as mentioned, values from 1 to 7, according to the fitness function. The resulting image is therefore the image with the maximum information. But that does not always mean that we will also have faith in human vision. Therefore, we calculate a second image, where we have normalized the source-image table, to have an average of the number of source-images, having approximately equal content of each image, being neither too bright nor too dark. . This shifts the values to all pixels.

It is important to note that all of this processing is done at source level rather than image (pixel brightness) level, so the quality of the final image is not affected.

The picture that is closest to how the human eye perceived the scene is as follows:

The two images above came from 4 source images, each with a very limited range.

In cases where source images are larger, and there is overlap in information, the result is much better. It is striking, however, that in this case the final picture is presented, from such limited sources.

Instructions for program execution

To run the program, we have the source photos in the format "# .jpg" where # is numbered from 1 to 7, the larger the number, the brighter the photo.

It first asks us to specify how many source images we have:

```
Specify number of source images. (3-7)
```

It then asks us if we want light post processing in the final photo, source image balancing in the second output photo is always done, regardless of our choice here.

```
Do you want postprocessing? (Y/N)
```

Finally, we can keep track of the processing, at any time, and have an idea of the total time.

```
evaluating 884 out of 2108
```

```
Elapsed time is 686.709808 seconds.
```

Once the program is finished, we are presented with two images, of maximum information and simulation of the human eye, and are stored for later use.

Examples