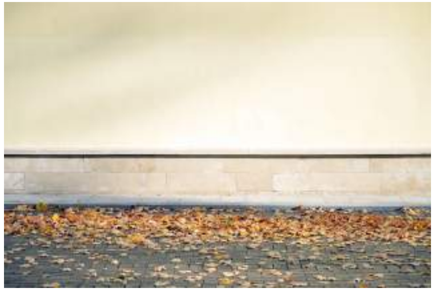
Computational Photography

Assignment #4: Blending[9]

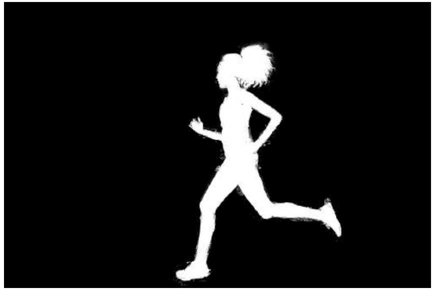
Nitish Sanghi Fall 2019

*Original work completed for 2019 Spring offering of computational photography

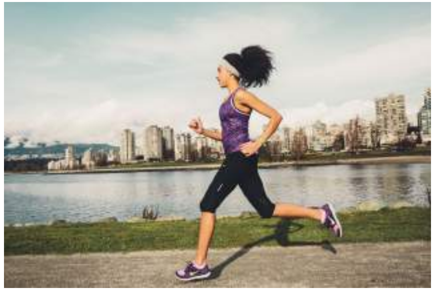
Input Images



Black [1]



Mask



White [2]

What is the subject of your images? Did you take your own pictures?

The subject of the images is a runner taking advantage of the onset of autumns and is running on paved road covered with dried leaves. The runner seems to be happy. In the black image only the paved road covered with leaves is imaged and in the white image the runner is running on a trail with water and a skyline in the background. No, I did not take these images. I found them on the internet. They have been cited in the resources.

Mask

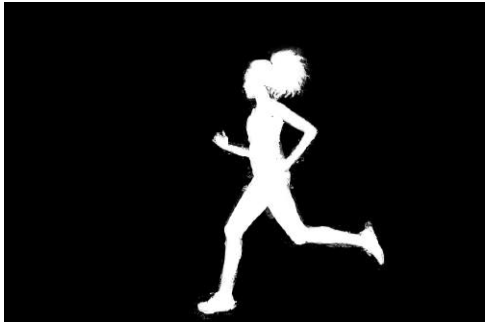
How did you create your mask? Discuss all tools you use.

The masks for this assignment were created using GIMP's "Foreground Selection Tool". Since this was the first time I used GIMP, I followed the instructions in the GIMP's help documentation. The mask was created using the following steps.

Step 1. Using the foreground select tool's lasso mouse icon the foreground in the white image was roughly selected. To improve the final result I attempted to select as little of the background as possible.

Step 2. Using the paint brush tool, connected lines were drawn on the foreground going over colors which were required to be kept for the extraction.

Step 3. The next step after separating the foreground and background into two layers was to create the binary mask. To create the binary mask, first the foreground was selected and using the paint brush the foreground was painted white. Since the background is a separate layer the paint brush does not paint on it.

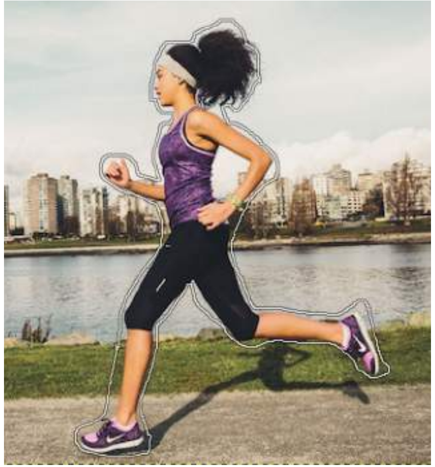


Mask made using GIMP

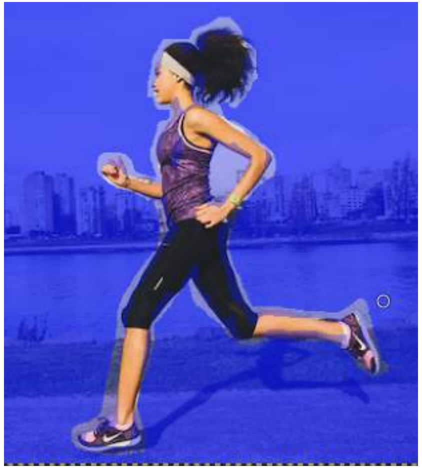
Mask

Step 4. The background layer was painted black to complete the mask.

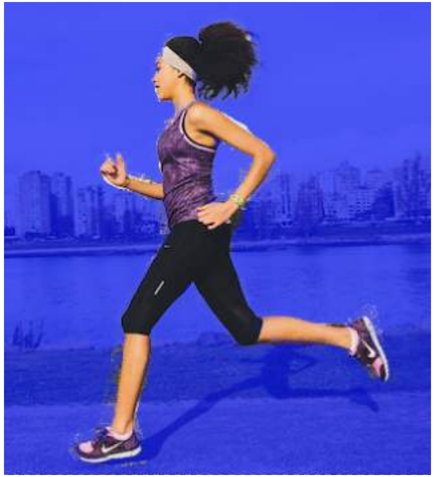
Step 5. Mask exported as .jpg to destination folder.



Step 1 Rough boundary of Foreground selected

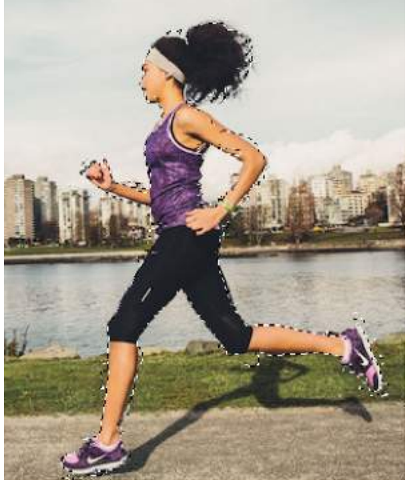


Step 2 Foreground Brushed to connect colors which will be captured



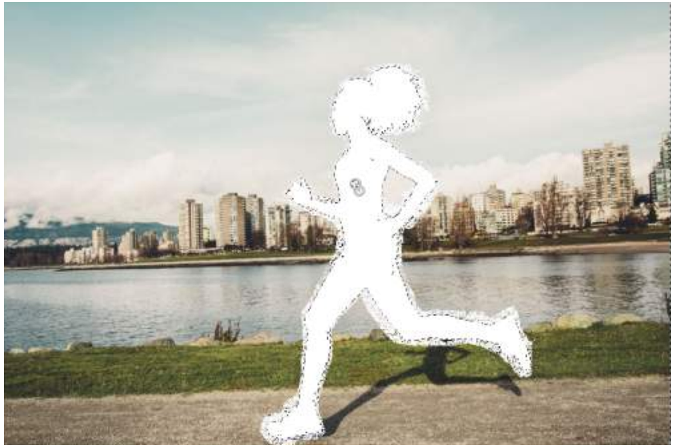
Step 2 Foreground and Background layers separate

Step 2 Foreground and Background layers separate

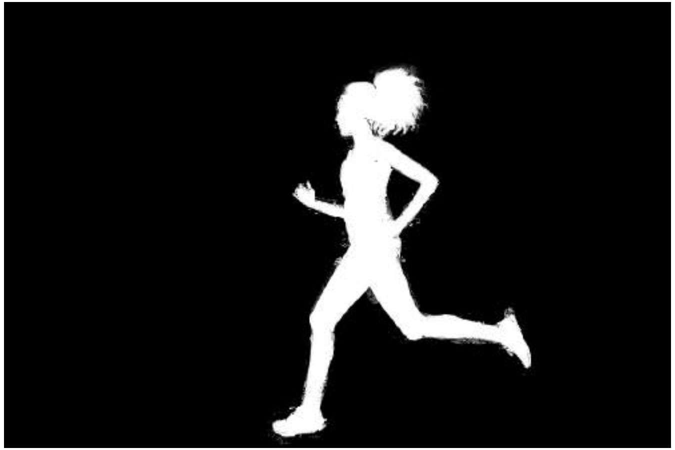


Step 2 Dashed line represents the outline of the foreground

Mask

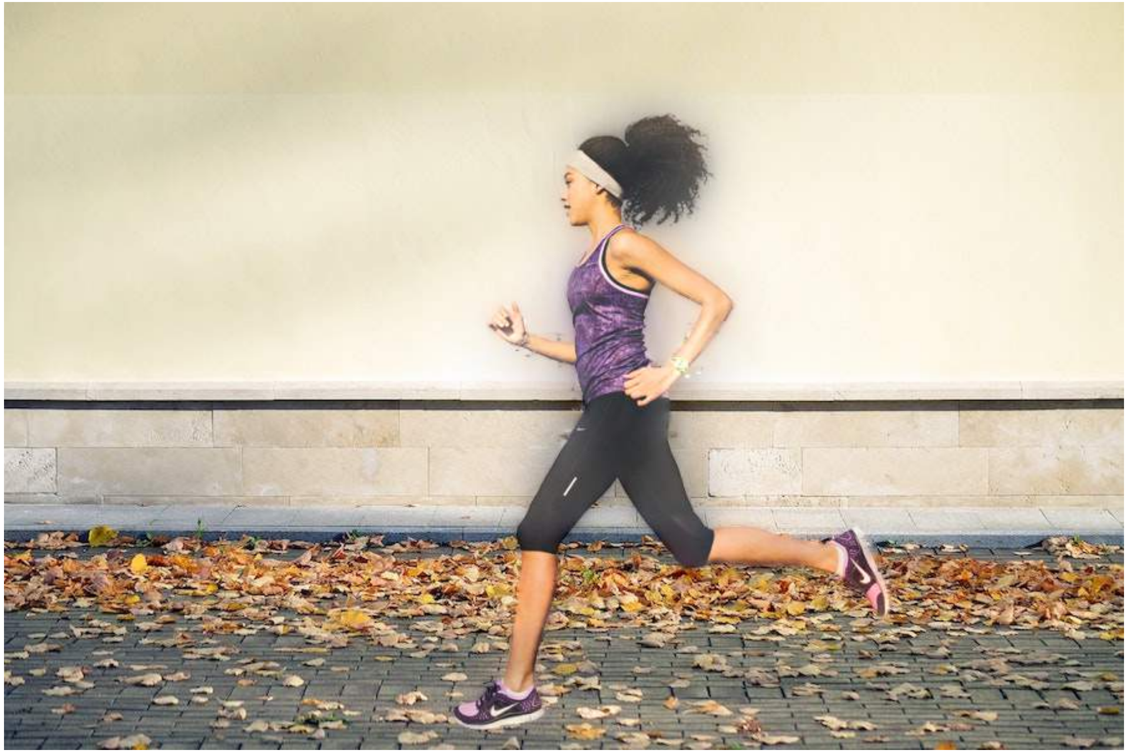


Step 3 Foreground painted white



Step 4 Background painted black

Final Blended Image



Une Course

Blend Discussion

How did you determine precisely where to make the blend?

The location of the blend was primarily driven by the images and the final target image. I attempted to "blend in" a **runner** from the white image to the black image. My intention was to blend in the runner so that is seems as if the runner was in the frame when the image was clicked. To make this happen I had to search through images to find the best two images which could potentially be blended with minimal secondary artifacts. While searching for the images I used variables like the perspective of the images, the lighting in the images (driven by time of day as the images were of the outside), relative size between the feature (runner) that waste be blended-in and the features of the receiving image (black image), location of blend on black image and color contrast with the surrounding features. The next step was to create a mask which would have minimal background from the white image to ensure white background features do not affect the blend. To do this the mask was created by carefully selecting and processing the foreground. With all these variables finalized, the precise location of the blend was easier to control by sticking to the contour of the runner.

How might you automate the location of the blending point?

The blending point can be automated by finding continuous seams in both images connecting pixels which have the "lowest difference of intensities" w.r.t. to each other. The "lowest difference of intensities" will be calculated by an appropriate cost function whose minimal value will result in seams of pixels which can be connected and blended together between the two images.

Blend Discussion

Do you think that blending or using a cut (as discussed in Lecture 4-04) is a better approach for your input images? Why?

For the input images I used, blending is the better approach than using cut. Using cut the boundary line where the masked component and the black image meet would be sharp and very noticeable. This primarily because the color intensities are quite different for the two images and a mask which will capture a smooth transition between the white image component and the black image is not possible. By blending the images using the mask the boundary does not change suddenly but has a slight transition in pixel intensities which comes off as a smoothening effect.

reduce_layer() & expand_layer()

What is the significance of using a = 0.4 for the generating kernel?

Parameter “a” determines the shape of the weighting functions. If “a = 0.4” then the weighting functions resemble a gaussian probability density function and convolution with a gaussian has the effect of low-pass filtering on the image. For the algorithm being used in this assignment the fundamental idea is to create a gaussian pyramid consisting of low_pass filtered images, which in turn are used to generate the band-pass images which are used for the multi-resolution spline.

Why does the output of expand_layer have to be multiplied by 4?

The output of expand_layer is multiplied by 4 because when at each level a subsample_image is expanded alternating rows and columns of zeros are added. When the upsample image is convolved with the gaussian kernel the overall intensity falls due to the presence of zeros. To rectify this the expanded layer is multiplied by 4.

Successes and Improvements

Did your blended image come out as you expected? Explain.

The blended image mostly came out as expected but there were a few secondary artifacts present. The aspect ratios of the features i.e. the runner (white image) and the wall and paved road (black image) looks close. The lighting is mismatch looks to be minimal. The runner was running on an open trail lit by the sun and the black image has almost the same light but improvements can be made. The blending boundaries do not have white image background distortions.

What were you most happy about with the final result? Be specific.

The final result does not have a lot of distortions or secondary artifacts which is great. The image has a certain crispness. And the colors of the two images work well and do not look out of place. The algorithm code worked without a glitch and handled the manual mask well. There are a few improvement which are incremental but would improve the blending.

Successes and Improvements

What improvements would you make to improve your final blended image (from template slide 5)? Even if you have great results, what part of the process could be improved?

A few improvements can definitely be made. The transition between the boundaries is not a smooth as I was attempting to achieve. This can be potentially resolved by automating the blend positioning so that the blends has better seam/boundary lines. The perspective, the size of features, and lighting on the blended have some mismatch. Using better images which are taken while accounting for these effects would definitely improve the final result.

Resources

[1] https://www.pexels.com/photo/autumn-background-blank-color-265527/

[2] https://www.pledgesports.org/wp-content/uploads/2017/03/1142-hobi-olahraga-lari-cegah-bahaya-dehidrasi-dan-overhidrasi-2.jpg

[3] A multiresolution Soline with application to Image mosaics, Peter J. Burt and Edward Adelson

[4] Udacity notes

[5] https://digital-photography-school.com/exposure-fusion-what-is-it-how-does-it-compare-to-hdr-how-do-i-do-it/

[6] https://mericam.github.io/papers/exposure_fusion_reduced.pdf

[7] https://photo.stackexchange.com/questions/20896/how-does-exposure-fusion-work

[8] http://www.hdrsoft.com/examples2.html

[9] This original work completed as part of spring 2019 offering of Computational Photography

Above & Beyond: Exposure Fusion

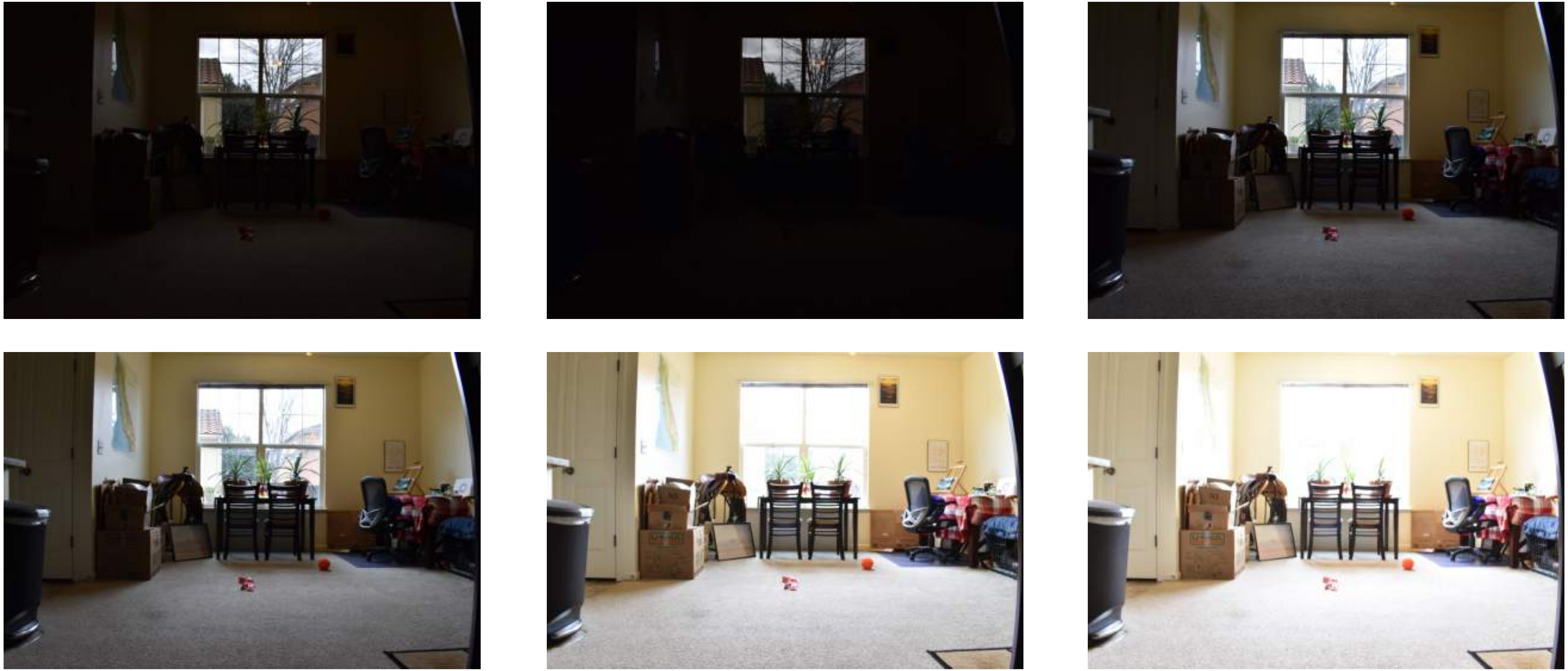
Researching different computational techniques I came across Exposure Fusion which sounded like HDR but upon further research [5][6][7][8] it turned out to be simpler and quick to implement. For the above and beyond this week, I further researched about Exposure Fusion and implemented it on a sequence of images I clicked at home.

Exposure Fusion simply put takes a sequence of images and evaluates pixels based on pixel characteristics and overall image quality, keeping the “best” pixels from the sequence and fusing together an image.

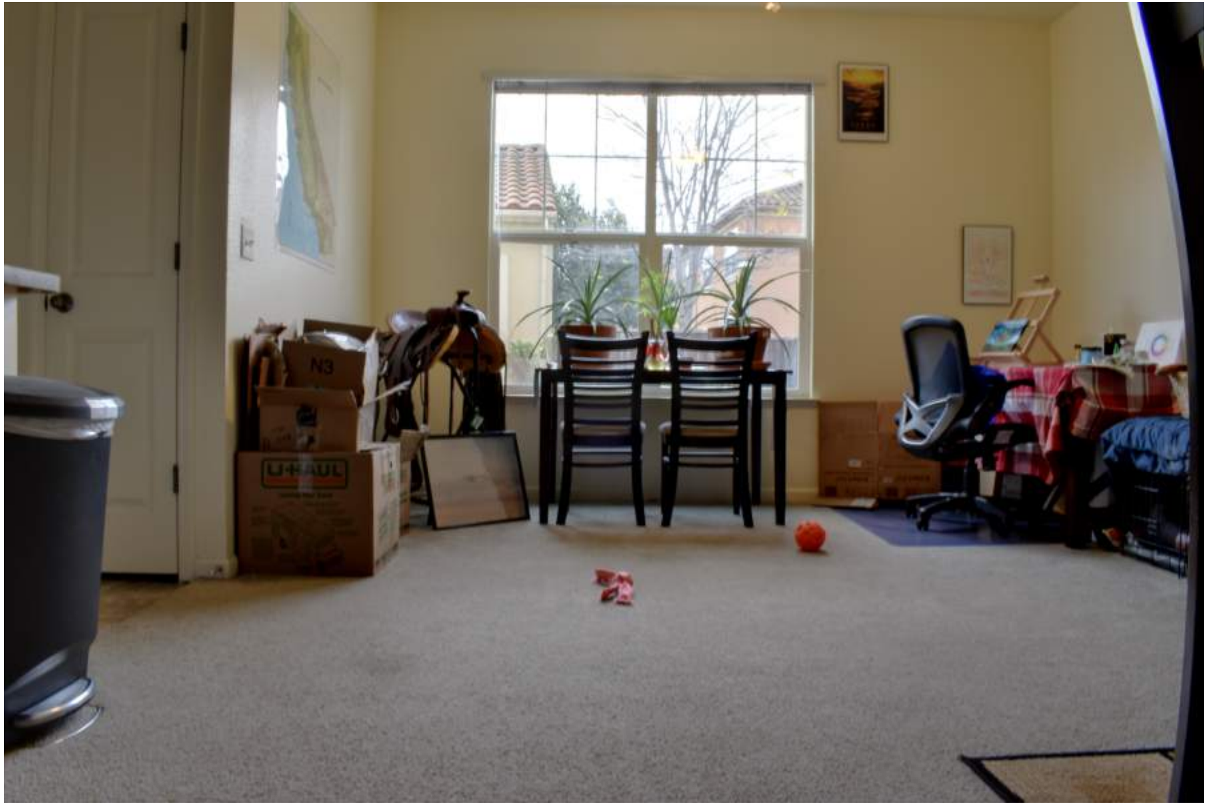
There are 3 prime characteristics [5][6] which are evaluated for each pixel across the sequence. They are:

- Luminosity: Pixels which are close to zeros or 255 are typically not used to find the pixel value. Pixels closer to middle intensity are preferred.
- Contrast: An image with high contrast is considered to be high quality, hence images in the sequence with high contrast are given higher weights.
- Saturation: Colors with high saturation are less washed out and so pixels with high saturation colors are given higher weights.

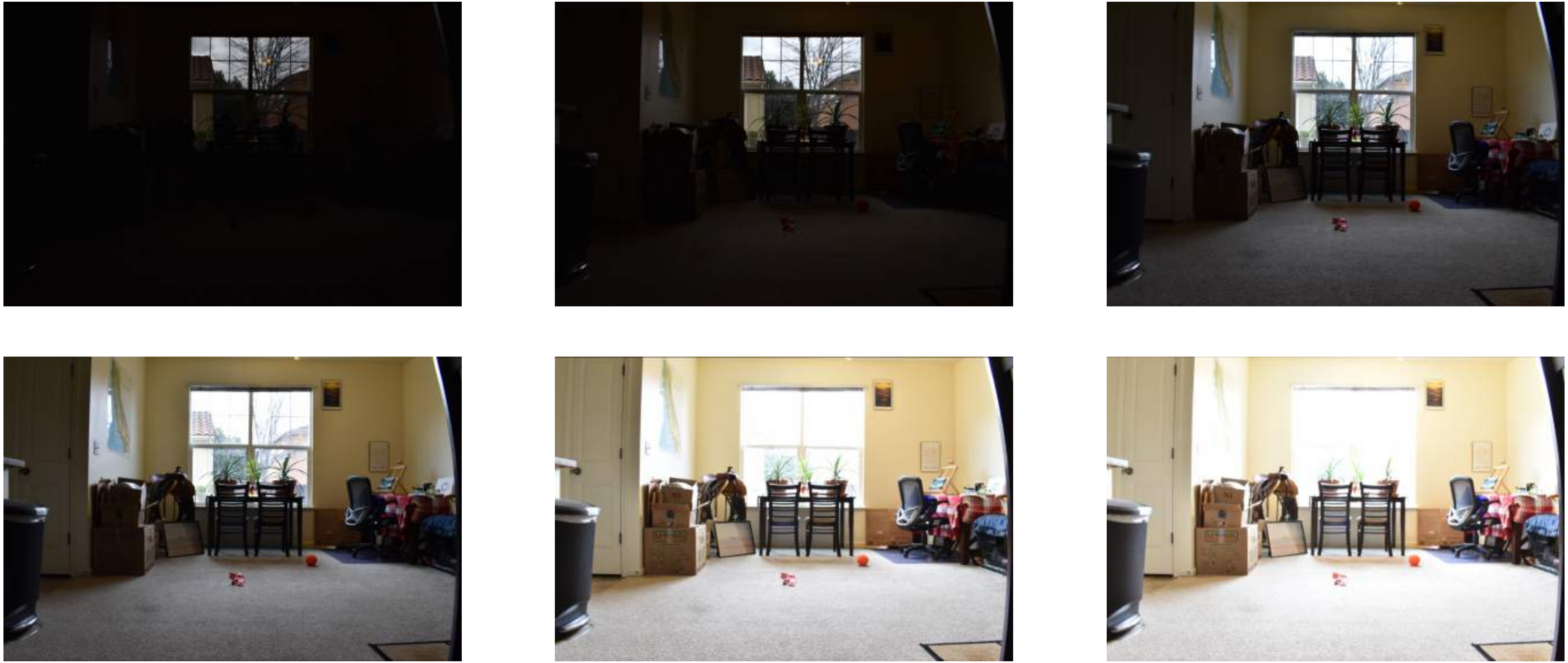
Exposure Fusion: Image Sequence Original



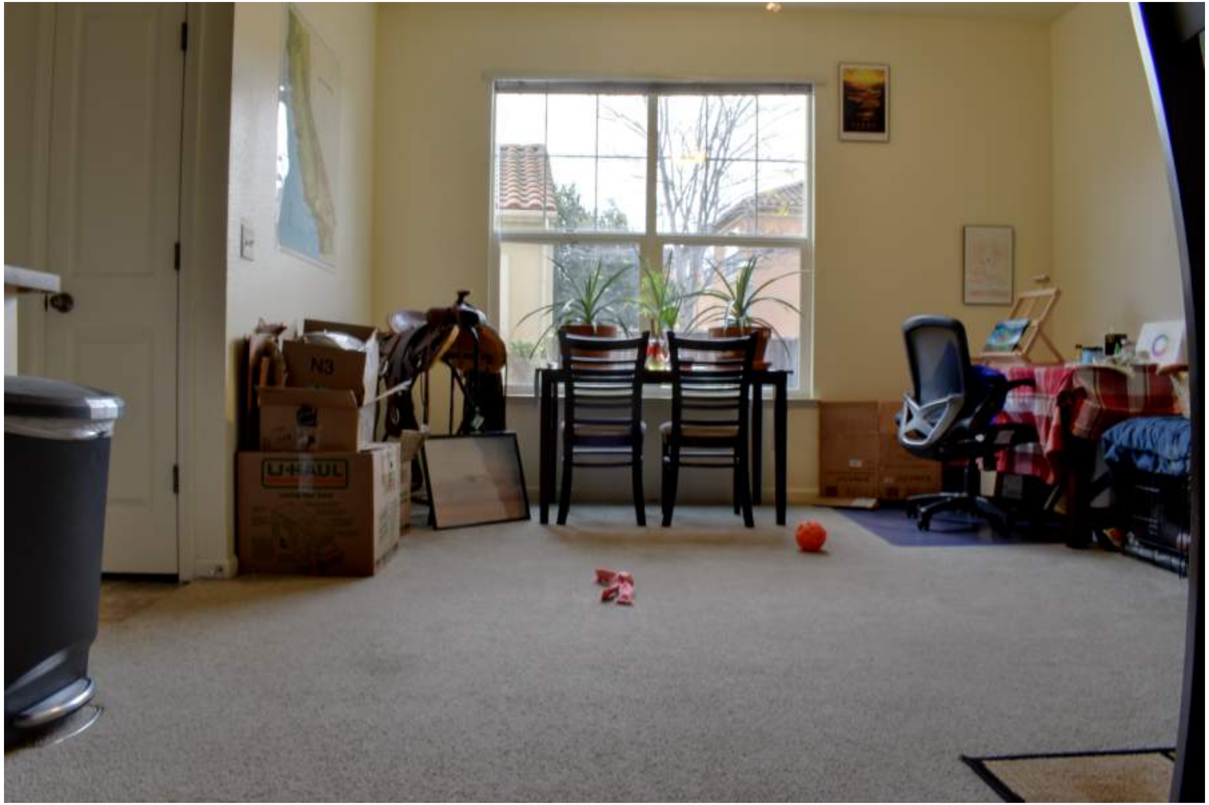
Exposure Fusion Final Image from Original Sequence



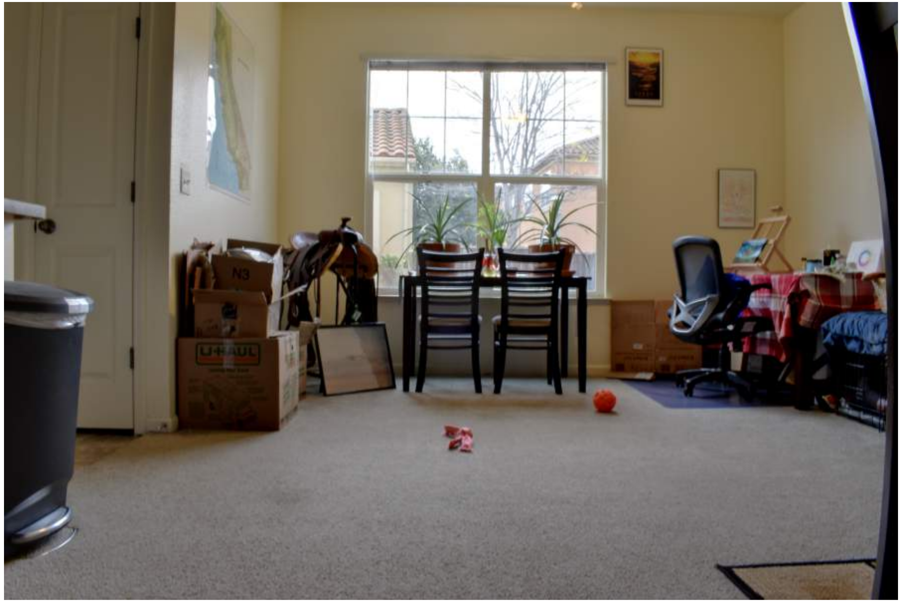
Exposure Fusion: Image Sequence Transformed to Remove Jitter



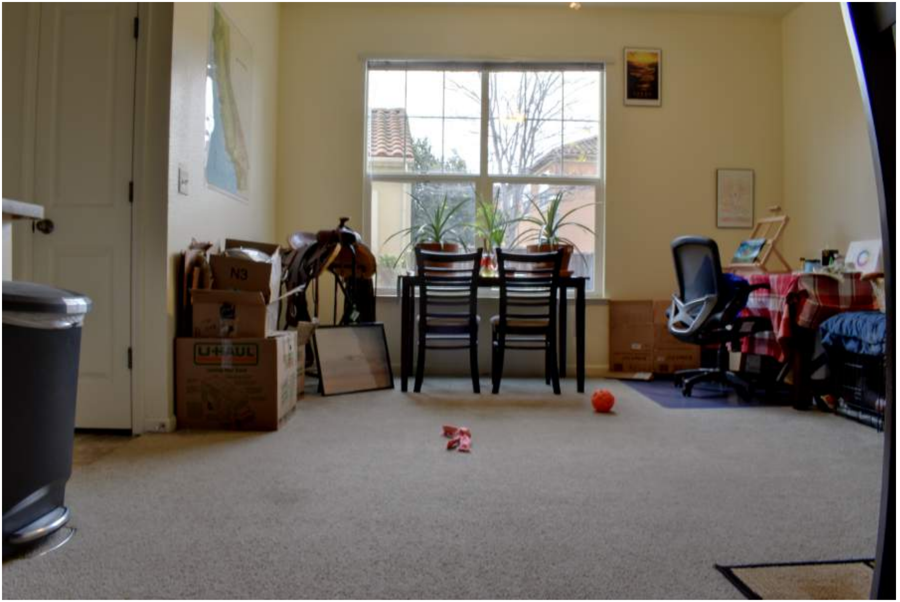
Exposure Fusion Final from Transformed Sequence



Exposure Fusion: Comparison



Final From Original Sequence



Final For Transformed for Jitter sequence