Georgia Tech's Computational Photography Portfolio

Radu Balaban

http://radubalaban.com
radu.balaban@gatech.edu

https://cs6475.wordpress.com/spring-2017

Assignment #1: A Photograph is a Photograph

Strange Light

Baia Mare, Romania
October 31, 2016 16:26

iPhone 6 Camera
Exposure: 1/160 sec
Aperture: f/2.2
ISO 32

Summary

Sharing a photo and the details of how it was captured - a great way to get started!

Assignment #2: Image I/O

Black & white

Blending

Flipping

Summary

OpenCV and Numpy are libraries that allow loading and saving images, as well as performing different transformations. Here are some examples of some simple transformations, performed on grayscale images.

Paying attention to data types is important - otherwise overflow issues may occur.

Assignment #3: Epsilon Photography

Summary

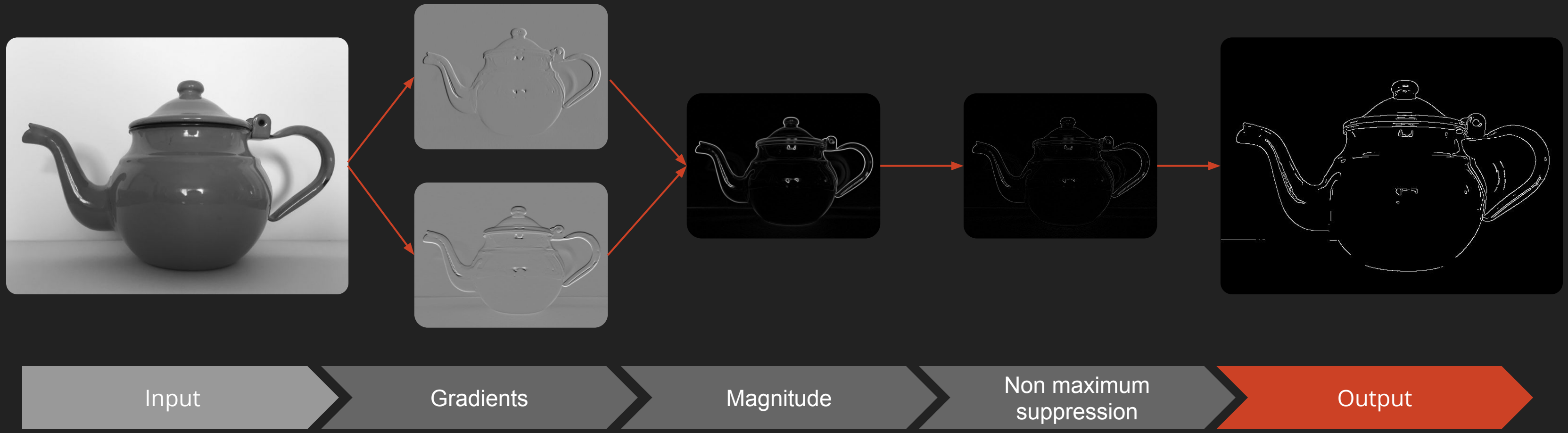
Epsilon photography consists of taking multiple photos in which a single parameter is slowly varied, and combining them computationally to create a new artifact.

In my example, four images (the ones at the bottom) are taken from slightly different positions with a small aperture camera (phone) and combined to simulate the bokeh effect of a camera with a larger aperture. This is called virtual aperture.

As the four input photos were too few to obtain a smooth blur, morphing was used to generate extra frames, simulating intermediary camera positions. The final result was obtained from the weighted average of all the photographs - and they lined up well because feature detection and homography transformations were used to bring the central doll at the same location, making it appear in focus.




Assignment #4: Gradients and Edges



Summary

Edge detection is a procedure that allows to identify edges in images, and one way to achieve this is by using the gradients of the image. The pipeline above is demonstrates my simplified implementation of the Canny edge detection algorithm.

Fun application - edge detection can be used to simulate drawing of sketches starting from photographs. I created an example and another one.

Assignment #5: Camera Obscura

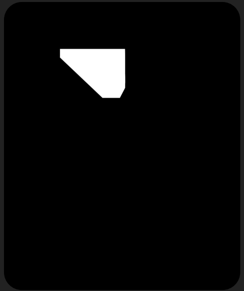
Summary

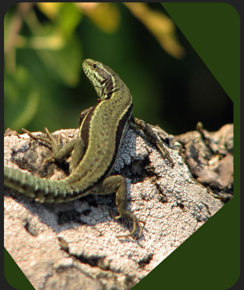
A camera obscura - dark room, literally - is one of the earliest photographic devices, known since antiquity.

Building a camera obscura is a surprising experience: it's really great to see the image outside the room (top) forming on the opposite wall (bottom) - simply because of having just a small hole through which the light can enter the room (center).

Low light photography can be tricky achieve with a phone. Taking multiple shots and averaging them, and then performing dark frame subtraction allowed me to get the result above from a very noisy image.

Assignment #6: Blending





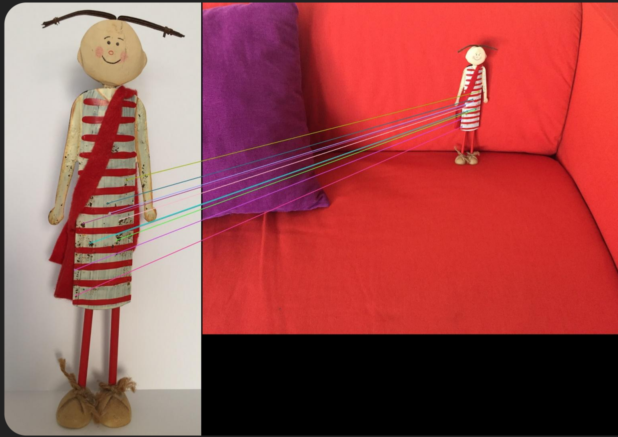
Summary

Image pyramids can be used to blend two images in a way that respects the Fourier frequencies: each pyramid layer blends the frequency band corresponding to its level with an appropriately blurred mask, creating a very natural result.

In my example, the two small lizard images have been blended with the mask in the middle - notice that even though the mask is not perfect, the output image on the left shows no blending artifacts.

Fun application - image pyramids can also be used to synthesize textures, you can see here (original textures on the left, synthesized texture on the right)

Assignment #7: Feature Detection



Summary

Testing a feature detection algorithm such as ORB in a variety of conditions provides valuable insight into how the algorithm works and how its parameters can be tuned to obtain better results.

Checking the correctness of the matches can be tedious - so I automated this part by pre-computing the homography between the main image and the test images, which allowed me to verify if the match occurs indeed in the same location. Using this automation I could really explore the parameter space and even obtain optimized parameters by running a genetic algorithm.

Assignment #8: Panoramas

Summary

Through a combination of feature detection, homography transformations and a blending algorithm, multiple related images can be stitched together to create a single, large, seamless panoramic image.

In my example, the images below were initially pre-processed through a cylindrical projection to match the movement of the camera, and they were blended using alpha feathering in the gradient domain. The resulting blended section was recovered in the intensity domain by solving the Poisson equation given by the gradient and the borders of the blended region.

Assignment #9: HDR

Summary

One limitation of ordinary cameras is their reduced ability to correctly capture the dynamic range of a scene.

Using multiple shots with different exposures (see below) allows recovering the irradiance map of the scene, which can then be transformed by tone mapping to better fit in the limited range of the displays.

In my example the irradiance map was recovered from 11 photos (a few of which are shown below), and the result has been tone-mapped through a custom histogram slicing algorithm.

Assignment #10: Pictures of Space (1)

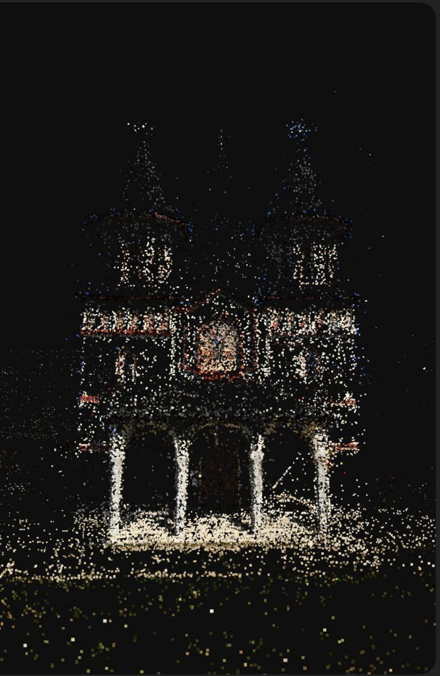
Summary

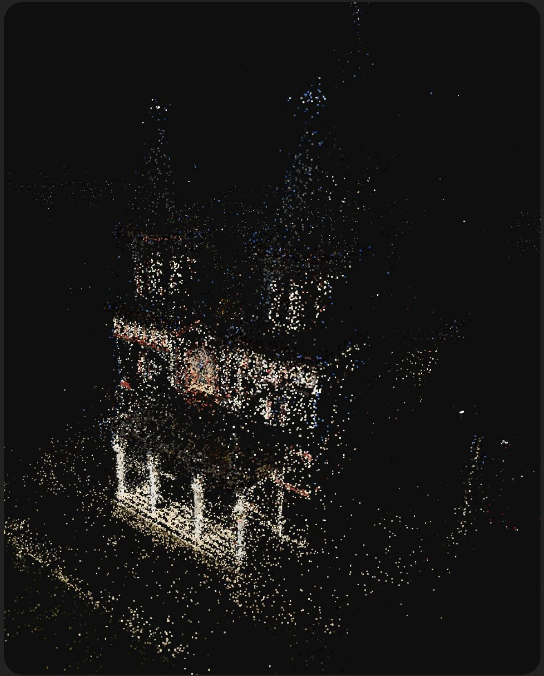
Specialized panorama software, such as Microsoft Image Composite Editor (ICE), makes it simple to produce a variety of panoramas.

The one on the top has been taken on March 22, 2017, at the historical center in Baia Mare, Romania and uses 23 photos arranged in a cylindrical projection. Full resolution panorama and images can be found here.

The panorama on the right has been taken on the same date, at St. Andrew's Church in Baia Mare. It uses a stereographic projection to create the 'little planet' effect. Full resolution panorama and images are at this location.

Assignment #10: Pictures of Space (2)





Summary

The 3D structure of a scene can be computed using a reasonably large number of photographs taken at that scene from different angles and locations. The images are matched using feature detection, the camera locations and parameters are recovered as well through calibration, and then the features can be mapped into 3D space using the structure from motion algorithm.

My example uses 107 photos (like the ones at the bottom) taken on March 22, at St. Andrew's Church in Baia Mare, Romania, and processed with the Open SFM software.

The computed cloud data and source images are available here.

Assignment #11: Video Textures

Summary

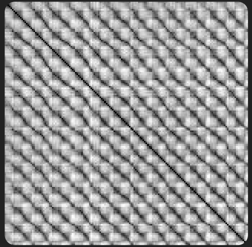
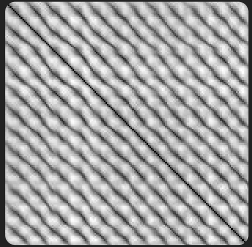
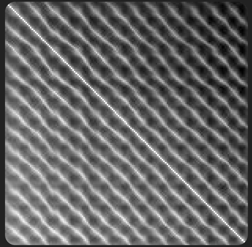
A texture is a repetitive pattern we can observe in space, while a video texture is one that can be observed in time.

Starting from a normal video, a loopable one can be created by analyzing the similarity of the frames with a similarity matrix such as the ones below, where each row and column stand for a frame in the video, and the corresponding pixel depends on the frame similarity (black - similar, white - dissimilar).

Similar frames can be used as jump locations, allowing the video to be played back infinitely in a variety of ways.

My example shows a seamlessly looping video - the parameters of which were tuned using a binary space search algorithm.

The animated video texture can be seen here, and the source frames can be found at this location.

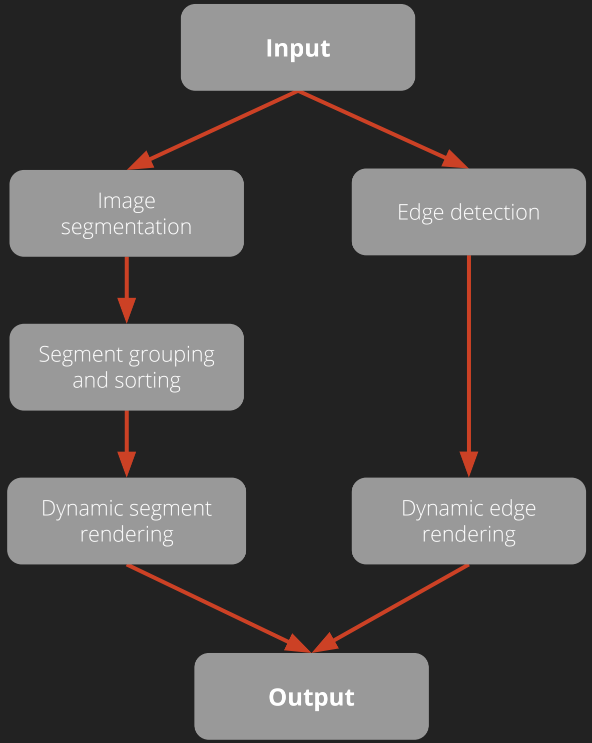
Final Project

Summary

My final project generates a video animation showing a timelapse of how a painting might come to life starting from a photograph.

This means drawing an initial sketch of the scene on a canvas, applying brushes of paint in a manner that imitates a human painter, and finally end up with an image that looks close to a real painting of the same scene.

Pipeline



Final Project (2)

Here are three sample videos generated starting from each of the three photographs.

Watch video

Watch video

Watch video

Credits & Thanks

Cover background image: https://www.flickr.com/photos/danielygo
Presentation theme: http://freegoogleslidestemplates.com

Thanks to Mark Carder for his great oil painting tutorials, to Professor Essa for the course, and to the awesome TA's and colleagues for making this such an enjoyable experience!