Generating Depth Maps from Stereoscopic Images

DYLAN BRUSS, Clemson University, USA

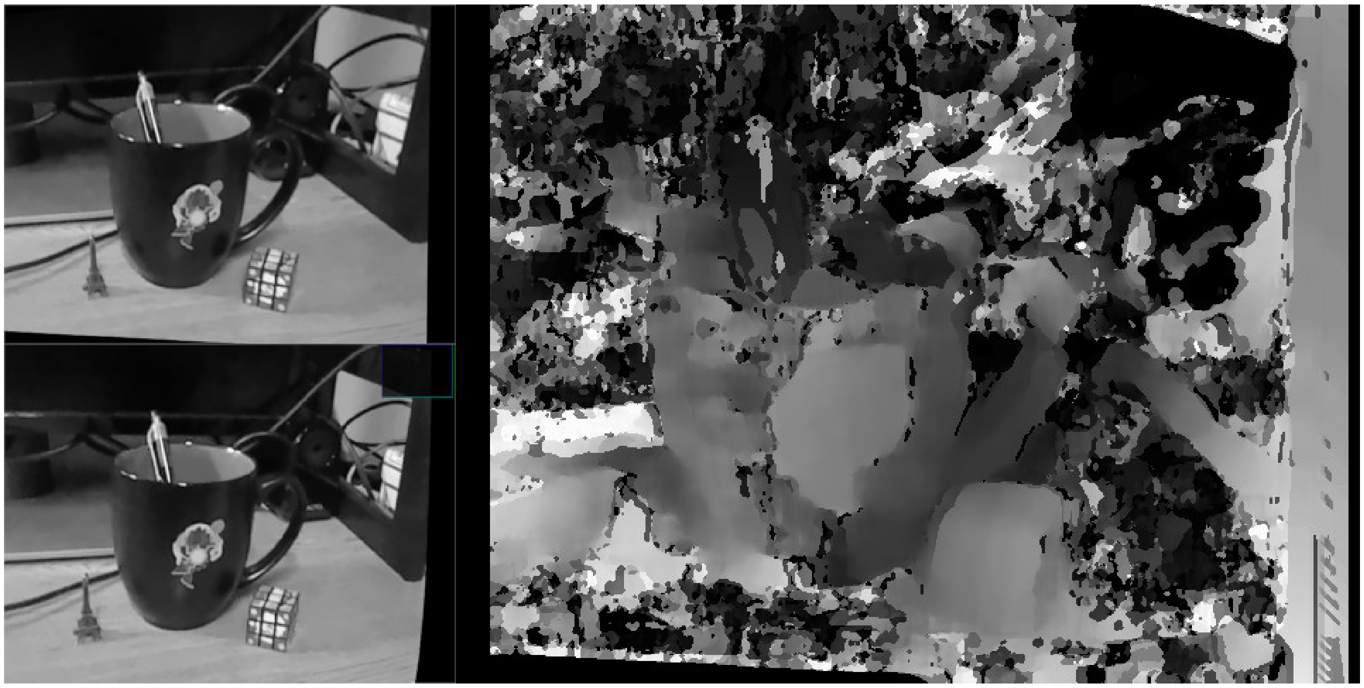

Fig. 1. An example pair of corrected stereoscopic images (left) and resulting depth map (right). Lighter colors in the depth map are closer to the camera, and darker colors are farther away.

This project aims to use advanced techniques to generate a depth map from a stereoscopic pair of low-quality images.

Additional Key Words and Phrases: stereoscopy, imaging, computer vision

ACM Reference Format:
Dylan Bruss. 2022. Generating Depth Maps from Stereoscopic Images. In . ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

Stereoscopy is an important part of computer vision, as it allows one to not only determine where an object is in a scene, but also how far away that object may be from the camera. This is used today in many applications, from autonomous vehicles to photogrammetry. This project goes over a technique for correcting and detecting depth from a pair of low-quality stereoscopic images.

2 METHOD

The methodology used to capture and process stereoscopic images in this project is described below.

2.1 Capture

Most images used in this process were captured with a Nintendo 3DS Model CTR-001 handheld system. This system features two 0.3 megapixel camera sensors at 640x480 resolution fixed 3.5cm apart from one another. Images were captured and compressed to stereoscopic .MPO image files on the device, and then separated into two complimentary JPEG compressed images before being run through the algorithm.

Additional, higher-resolution stereoscopic imagery was obtained from the Wikimedia Commons [1].

2.2 Calibration

Imagery captured with the Nintendo 3DS were calibrated and undistorted using a 6 by 9 checkerboard grid pattern. Nine images were used per camera lens in the calibration process. Four images for each camera were taken with the grid pattern in each corner of the frame. One image was taken with the pattern in the center of the frame. Finally, four additional images per camera were captured at various perspectives.

Nine images in the calibration process to generate the camera matrices and distortion coefficients for each lens. Next, these matrices and coefficients were used in the OpenCV stereoscopic calibration algorithm. The resulting values were then used to undistort each new stereoscopic image.

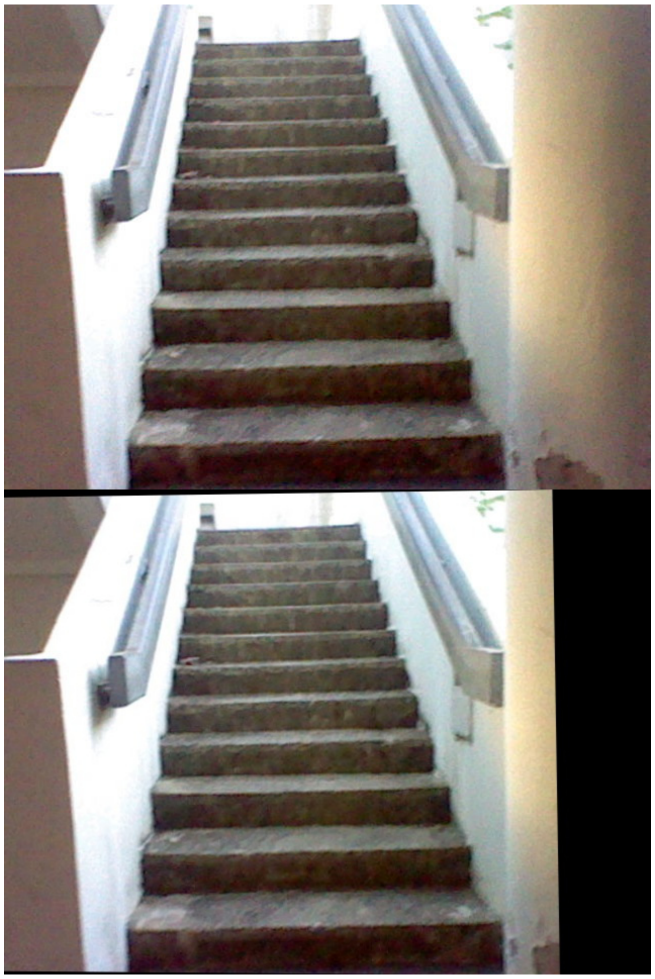

Fig. 2. The left-eye stereoscopic image (bottom) is aligned to the right-eye image (top) using feature detection

2.3 Alignment

The imagery produced by the Nintendo 3DS was often aligned poorly, which resulted in poor performance in the block matching algorithm. As a result, a feature-based alignment method was employed [2].

This method detects a number of well-matched features with the ORB (Oriented FAST, Rotated BRIEF) method of feature detection. from this, a homography matrix is calculated. The homography matrix is then used to translate the left-lens stereoscopic image closer to the right-lens image, as seen in figure 2.

2.4 Image Filtering

Once the images are undistorted and aligned, they are run through a few simple algorithms to aid in the stereo matching process. First, each image is run through an optimized Non-local means de-noising algorithm with a h-value of 3, a template window size of 7 and a search window size of 21. This helps remove the excess noise in the image, which can provide false features that the algorithm may recognize as details in the scene, creating a false depth map. Next, both images are run through a 2-dimensional sharpening filter, to emphasize the remaining details.

2.5 Stereo Disparity Algorithm: Semi-Global Block Matching

Once the images have been pre-processed, the Semi-Global Block Matching algorithm is used to find the disparity between the two images. This was found to perform much better on the given data than the default Block-Matching algorithm in OpenCV. The algorithm is run set to detect 32 disparity levels starting at -8, with a block size of 21 and a P2 value of 1. The disparity is then computed twice, once normally, and once with the left and right images reversed. The disparity from the reversed calculation is inverted, retaining zero values as zero, and is averaged with the normal disparity values to produce the final depth map. This is done because the inverted disparity has the ability to pick up some details not visible in the normal pass, and vice versa. Combining the two images allows for a better final product

3 RESULTS

The results of this process can be seen in figure 1. While it contains many artifacts, especially in areas of poor detail, like the monitor and the background wall, high-detailed areas, such as the mug, Rubik's cube, and frame on the side are well represented in the final depth map. Some smaller details, such as the pen inside the cup, and the Eiffel tower miniature, are present in the depth map as well, though slightly distorted.

4 CONCLUSIONS

Overall, the scene is not perfect, but the algorithm performed admirably for low-fidelity 640x480 images. Many of the issues I believe have to do with the capture method, and not the stereo reconstruction method. Many of the details of the scene, which are very important for reconstruction, are lost to noise and lack of resolution. Had I been in possession of a better fixed stereoscopic camera, the algorithm could have performed much better. It may be possible to use a monocular camera at two different points in time, but it would make it much more difficult to properly calibrate the images if they are not taken from a fixed, verifiable length from each other.

Future work may include using better photography equipment, or using machine learning to infer additional detail. Additionally, machine learning algorithms or other post-processing techniques can be applied to the final image to improve the continuity and remove artifacts.

REFERENCES

[1] Brad Regier. Stereo (45432795025).jpg, 2018.
[2] Satya Mallick. Feature based image alignment using opencv (c++/python). *LearnOpenCV*, Nov 2021.