# Catadioptric Stereo

Irfan Bunjaku
ETH Zurich
irfan.bunjaku@inf.ethz.ch

Dominik Bornand
ETH Zurich
bornandd@student.ethz.ch

Felix Sarnthein
ETH Zurich
safelix@student.ethz.ch

Christian Gasser
ETH Zurich
chgasser@student.ethz.ch

## Abstract

*Nowadays electronic devices such as smartphones often come with multiple cameras to help with depth estimation. These can be used for 3D reconstruction and for foreground / background separation in images. However, a much simpler setup has been known to work well for a long time as was presented by Gluckman and Nayar [5]. A physical camera setup with a rigidly attached mirror enables us to capture stereo images with a single camera which we then aim to use for depth estimation. Our goal is to build the setup, implement the necessary algorithms for automatic calibration and then perform depth estimation.*

## 1. Introduction

Using mirrors to achieve multiple viewpoints has multiple benefits. First, all needed components are commercially available which makes the results inexpensive and our pipeline easy to use. Further, by using only a single camera we need to estimate the intrinsic parameters only for one camera. While this benefit also holds for other single-camera methods that for example use temporal stereo images, i.e stereo images acquired sequentially by a single camera, they have the drawback of having to recompute the extrinsics for every input image. Further there is some uncertainty about the depth estimation which is based on the uncertainty of the position of the images.

Our camera setup works as follows. We use a video sequence to detect the mirror in our scene with optical flow, in particular by using the Lucas-Kanade method [8]. In the next step we calibrate the intrinsic as well as extrinsic parameters for our physical and virtual camera. This has to be done only once in our setup and the found parameters can be reused for arbitrary many depth estimations. Finally we split the input image with respect to the detected mirror position, flip the mirrored side and apply rectification to the mirrored image. The rectified image is then used to compute the disparity map.

## 2. Related Work

The standard stereo setup to extract 3D information from 2D digital images usually requires two horizontally displaced cameras to obtain two different views from a scene. The camera intrinsics and extrinsics need to be calibrated only once during an initial calibration phase. Then, depth information is obtained by comparing rectified images and computing the disparity map. The disparity encodes the difference in the horizontal coordinates of corresponding image pixels. The values in the disparity map are inversely proportional to the distance of the object in the corresponding pixel. [1].

Single image depth estimation is highly underdetermined. Most current state of the art methods for single image depth estimation are machine learning based and usually use convolutional neural networks as thoroughly discussed by Mertan *et al*. [9].

To the best of our knowledge catadioptric stereo was first introduced by Gluckman and Nayar [5]. They discuss geometry and calibration of catadioptric stereo with two planar mirrors. They recover focal length with a single catadioptric image solely from a set of stereo correspondences.

Since then mirrors have been widely used in different computer vision tasks for use cases which otherwise would not be feasible with a single camera. Such as for example by Yi *et al*. [10]. They build an omnidirectional stereo imaging system which uses a concave lens and a convex mirror to produce a stereo pair of images on the sensor of a conventional camera.

Kawahare *et al*. [6] observed that images which capture direct and water-reflected real-world scenes offer an additional viewpoint to the direct sight which then form a stereo pair. Or in other words the water reflection can serve as a natural semi-translucent mirror. They remove the environmental illumination and the reflection from the bottom surface which prevents direct stereo matching. Finally, they estimate the displacement field in the reflected observation caused by the waves and show that this can be modeled as surface normal variations of the water surface and derive an iterative approach to simultaneously recover the shape of both the scene structure and the water surface.

# 3. Method

In this section we explain in detail our used method. We start out with the main objective of our method: the depth estimation. Initially we assume that the mirror position and the intrinsic as well as the extrinsic parameters of the scene are given. We then proceed by explaining how our pipeline calibrates the intrinsic and extrinsic parameters and detects the mirror. Finally, we describe our physical camera setup. We used blender to create a virtual setup with more controllable conditions to develop the individual steps of our pipeline.

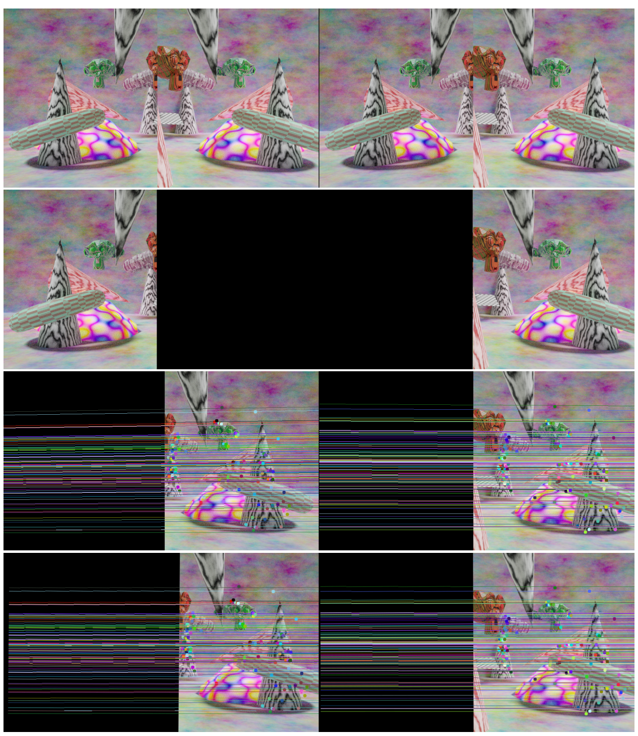

Figure 1. Pipeline for our catadioptric stereo setup: (i) Duplicate the input image. (ii) Split the images and mask out regions occluded by mirror. (iii) Flip the mirrored image and plot epipolar lines. (iv) Rectify images for disparity computation, epipolar lines are now parallel.

## 3.1. Depth estimation

Given a calibrated setup with known mirror position, intrinsics and extrinsics, we first spilt the input image into a direct view and a reflected view. We then flip the mirrored view and treat the two views as regular stereo image inputs. To preserve the original image dimensions, we mask out the regions of the scene which are occluded by the mirror as in Figure 1. To visualize the fundamental matrix with epipolar lines, we extract SIFT features [7] and compute the matching with a FLANN based method. We use the fundamental matrix to rectify our image input pair on which we finally compute the disparity map using the semi-global block matching algorithm [3].

## 3.2. Camera calibration

### 3.2.1 Intrinsic and extrinsic calibration

To calibrate the intrinsic and extrinsic parameters, we adapt the procedure described in [2]. We capture multiple images of a chessboard with different angles and distances. Then we extract the chessboard pattern from both the direct and the mirrored view as in Figure 2. Since the intrinsic parameters are the same for the direct and the mirrored view, we can combine them into a batch which is now twice the size of our original calibration data set. This batch is then used to create an initial estimate of the camera intrinsics.

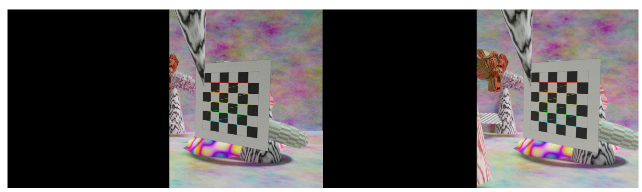

Figure 2. Detected chessboard in direct and mirrored view.

Using dual camera calibration, we can now refine the estimate of the camera intrinsics and compute the extrinsics. In particular, we compute the fundamental matrix which combines all information about the scene, i.e

$$F = K^{-T} \cdot E \cdot K^{-1} = K^{-T} \cdot [t]_\times R \cdot K^{-1}$$

We transform the rotation matrix to axis angle representation and consider the angular deviation to assess the quality of the computed fundamental matrix with respect to our blender setup.

### 3.2.2 Mirror detection

Given a moving scene with movement in horizontal direction we note that the depicted objects on the mirrored view move in the opposite direction of the direct view. We can use this to detect the mirror automatically. To this end we select a feature set on the input scene. Here we choose key points on a horizontal line. We then apply the Lucas-Kanade method [8] to track these features in our moving scene. We examine the optical flow along the line to determine where the direction of the movement changes and thus determine the location of the mirror.

### 3.3. Physical setup

For our physical Setup we use a *Canon PowerShot G16* and two square mirrors with a side length of 11.5 cm. We attach the mirrors on a piece of wood which we then screw on to a metal frame. On this frame we attach our camera as in Figure 3

Figure 3. Canon PowerShot G16 attached to our rigid construction

### 3.4. Recomputing extrinsics via SIFT

By following the standard stereo pipeline we assumed that our extrinsics are fixed after calibration. This assumption is violated since small movements of the mirror can cause quite drastic changes to the viewpoint of the reflected images. As an alternative solution we apply RANSAC on the previously computed SIFT features to estimate the extrinsics for every input image individually instead of once during an initial calibration step of the whole setup. In particular we use five-point RANSAC from OpenCV [4] and then add the intrinsics to obtain the fundamental matrix.

## 4. Results

### 4.1. Extrinsics calibration

The blender setup allows us to compare our computed fundamental matrix to the ground truth of the scene. To test the extrinsic calibration we created a scene where the mirror is tilted horizontally towards the camera by an angle of 2 degrees. This means that the rotation between the direct and reflected view is 4°. Since the underlying algorithms are not deterministic, we can get varying results. In roughly 70% of our test-runs the calibration algorithm converges to a correct solution as in Figure 1. However in some cases the solution converges to an angle of 4.26° which might seem like a very small error in absolute terms but is relatively big compared to the ground truth of 4°. Taking a closer look at the location of the epipolar points and the result of the rectification, we see that the resulting fundamental matrix is far from correct as you can see in Figure 4.

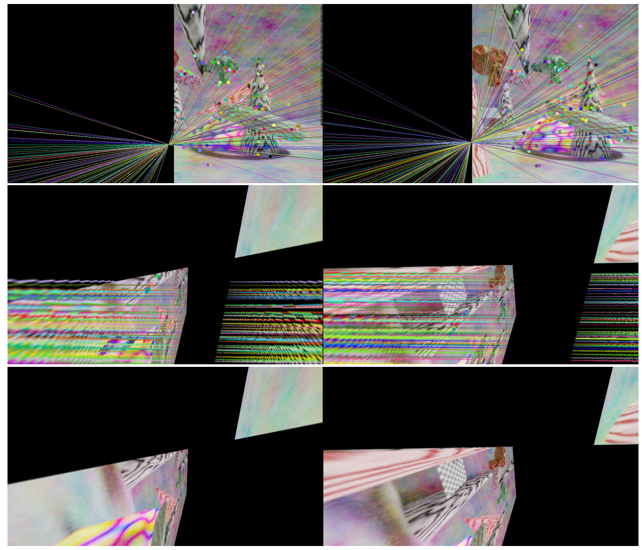

Figure 4. Extrinsics calibration converged to an angle of 4.26° instead of the 4° ground truth.

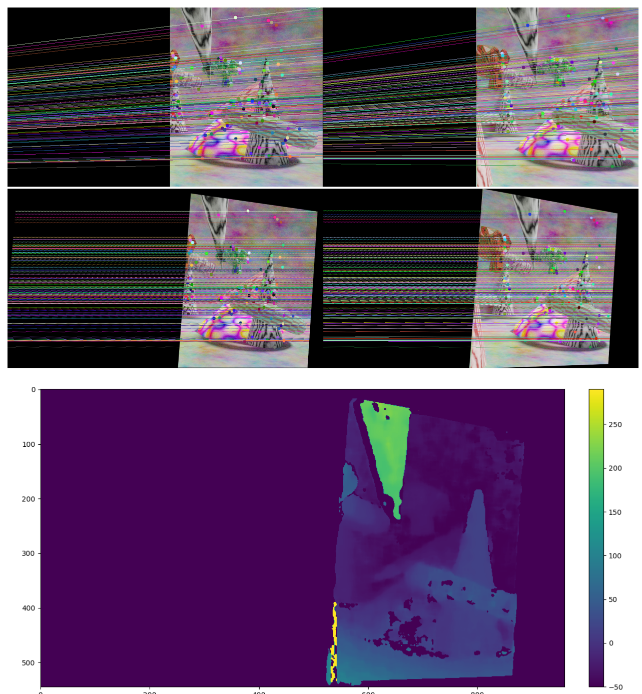


Figure 5. Extrinsics via SIFT do not fulfill catadioptric constraints.

### 4.2. Extrinsics via SIFT

Computing the extrinsics from SIFT features as described in Section 3.4 yields a more stable fundamental matrix. The resulting fundamental matrices, however, violate the constraints imposed by our catadioptric setup. For example in Figure 5, the epipolar point of the left image is detected on the wrong side. It should be on the right side of the image since the real and the virtual camera are facing each other. Instead, it usually occurs that the epipolar lines converge towards the same side.

In a catadioptric setup the fundamental matrix has only three degrees of freedom. Therefore, we allow for too much flexibility by using five-point RANSAC and consequently obtain solutions which are outside of the valid solution space. Because RANSAC finds a solution which correctly relates most SIFT features, the rectification maps matching SIFT features to the same vertical location. This means that the block matching algorithm is still able produce good result, as can be seen in Figure 5.

### 4.3. Mirror detection

Our optical flow segmentation works to some extent on the blender scene as you can see in Figure 6. The determined segmentation split is not exact but close enough for further processing. Note that the intrinsics and extrinsics computations do not depend on a accurate image split, the field of view just diminishes.

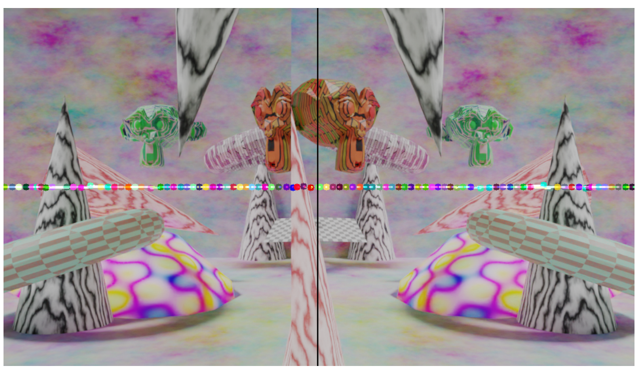

Figure 6. Optical flow image segmentation on blender input.

However, our proposed method struggles with real world scenes, like in Figure 7. This is due to the fact that the noise incurred by moving a hand-held camera produces unsteady optical flow. This is further amplified by the slackness of the mirror which causes the segmentation split to change over time. Therefore, the assumption that the optical flow is strictly separable into left and right motion is violated and we do not obtain an automatic segmentation on the physical setup.

### 4.4. Real world images

Using SIFT features to compute the extrinsics and by manually segmenting the image, we are able to produce disparity computations on real world images as in Figure 8. To demonstrate this we build a scene with depth variations and well track-able SIFT features at multiple depths. We try to align the camera with the mirror such that the real and mirrored views are not titled towards each other. This means that the camera needs to be close to the mirror, otherwise we would have no overlap of the real and virtual view.

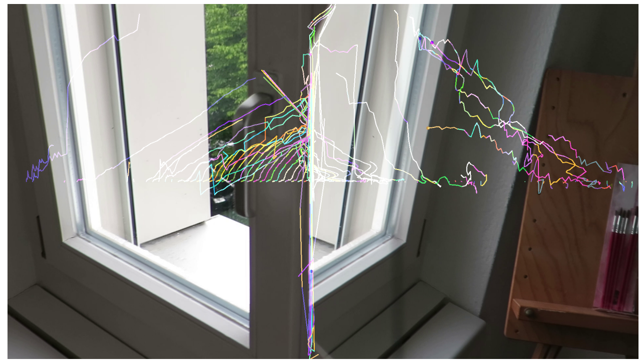

Figure 7. Optical flow image segmentation on real input.

First we observe that there are significant internal reflections in the mirror due to the small distance which then lead to blurring artefacts on the right partition. Secondly, we note that the epipolar lines are close to parallel and thus the fundamental matrix contains indeed very little rotation. Finally, we are able to compute a reasonable disparity map from the rectified images. Note that we also rectify the mask to crop the disparity map to the valid regions.

## 5. Conclusion

A mirror attached to a camera gives an additional viewpoint of the scene. We created a pipeline to extract depth from such a catadipotric system and build a physical setup to investigate the method in practice.

There are two means to obtain the camera extrinsics. Dual camera calibration allows to compute the extrinsics together with the camera intrinsics during an initial calibration phase. Another approach is to use RANSAC to compute the extrinsics from SIFT features for every input frame individually. The first method suffers from instabilities, while the second method consistently produces results which are close to the correct solution but outside of the valid solution space. It might be interesting to adapt a three-point RANSAC algorithm to impose a catadipotric prior on the extrinsics computation.

We use Lukas-Kanade optical flow to automatically detect the mirror segmentation from a video sequence. This method works on a virtual blender scene. On real data, it is not flexible enough and needs to be adjusted. One could for example compute the dense optical flow over many frames and than use an SVM with slackness to enable this required flexibility.

We show that the concept works on a virtual blender scene. In practice, the leeway of the mirror poses additional problems. The segmentation of the mirror might vary and

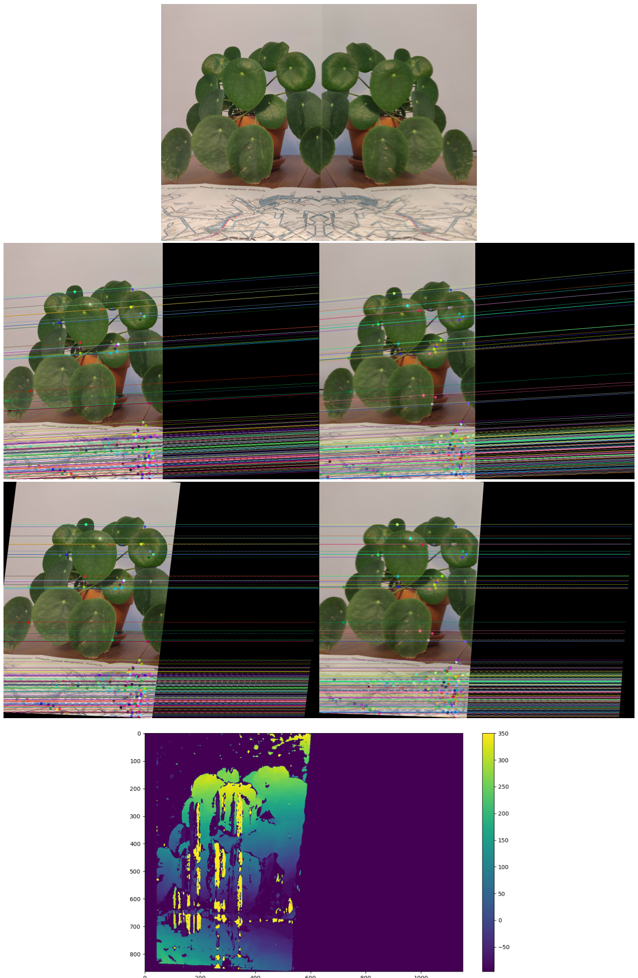

Figure 8. Running example on catadioptric image of a plant: (i) Input image. (ii) Flip the mirrored image and plot epipolar lines. (iii) Rectify images for disparity computation, epipolar lines are now parallel. (iv) Disparity map.

needs to be adjusted on the fly. Furthermore, the extrinsics cannot be calibrated but need to be recomputed in every frame using SIFT features and RANSAC. This raises the question if it would not be more efficient to follow a structure from motion approach where more advanced methods like bundle adjustment are available.

## References


[1] Computer stereo vision - wikipedia. https://en.wikipedia.org/wiki/Computer_stereo_vision. (Accessed on 05/18/2021). 1

[2] Opencv stereo camera calibration sample. https://github.com/opencv/opencv/blob/master/samples/cpp/stereo_calib.cpp. (Accessed on 06/13/2021). 2

[3] Opencv stereosgbm. https://docs.opencv.org/3.4/d2/d85/classcv_1_1StereoSGBM.html. (Accessed on 06/14/2021). 2

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 3

[5] J. Gluckman and S. K. Nayar. Catadioptric stereo using planar mirrors. *International Journal of Computer Vision*, 44(1):65–79, 2001. 1

[6] R. Kawahara, S. Nobuhara, and K. Nishino. Shape from water reflection. *CoRR*, abs/1906.10284, 2019. 1

[7] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. 2

[8] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). volume 81, 04 1981. 1, 2

[9] A. Mertan, D. J. Duff, and G. Unal. Single image depth estimation: An overview. *CoRR*, abs/2104.06456, 2021. 1

[10] S. Yi and N. Ahuja. An omnidirectional stereo vision system using a single camera. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 861–865, 2006. 1

## A. Work distribution among team-mates

| Task | Irfan B. | Dominik B. | Felix S. | Christian G. |
|---|---|---|---|---|
| Physical Setup | | | | ✓ |
| Intrinsic Calibration | | | ✓ | ✓ |
| Optical Flow | | ✓ | | |
| Parameter Tuning | ✓ | ✓ | ✓ | |
| Blender Setup | | ✓ | | |
| Code Cleanup | ✓ | ✓ | ✓ | |
| Documentation | ✓ | | ✓ | ✓ |
| Rapport | ✓ | | ✓ | |