Project Proposal

Stereoscopic Reconstruction on Android

1 Abstract

The problem of feature detection is well known and studied, but can be computationally expensive. In this project, we aim to build a robust feature detection system which will be more accessible, by utilizing a mobile phones built in hardware. Inspired by Android 9 (Pie) providing an API for multiple cameras [1] doing true stereo matching on mobile might become much easier. Currently, most stereo matching algorithms take two pictures sequentially and rely on the accelerometer and gyroscope to detect the positional disparity. Finally, we demonstrate the usability of our feature matching by applying a bokeh effect to one of the input images.

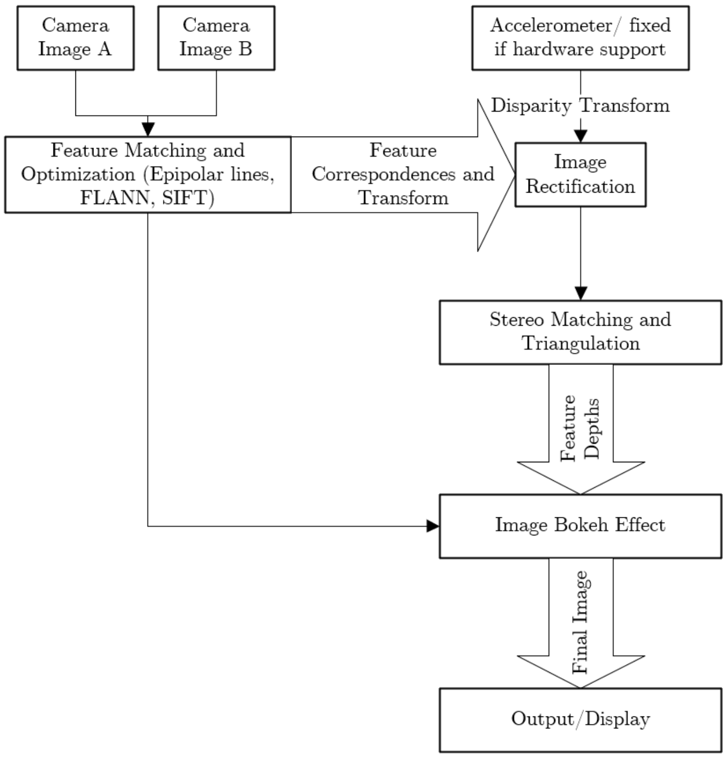

Figure 1: Our Proposed Processing Pipeline

Data Capture First, we capture two distinct RGB images from the phones camera sensor. At the same time, we also capture the orientation of the camera with the gyroscope sensor. Starting with the first image, we sample the accelerometer to obtain the inertia at discrete points between the two pictures.

Process Sensor Data The inertial data of the accelerometer combined is integrated twice to obtain the positional difference [7, 4]. Here, the numerical integration error accumulates over time. In contrast, the rotational difference between to RGB data sets is obtained by sampling the gyroscope sensor. We will test if the supplied commodity sensors are accurate enough to augment the image rectification process.

Feature Matching and Optimization After detecting and matching features we then use the methods from the lecture to optimize the transformation with a non-linear solver e.g. Ceres. This step is optional depending on the quality of the sensor data.

Image Rectification With the calculated translation and rotation, we can rectify the images [5, 2].

Stereo Matching Once rectified, we can apply a stereo matching algorithm e.g. FLANN, Block Matching, PatchMatch [6] or PM-Huber [3] to calculate the displacement map.

Depth Map The depth map is then calculated by triangulation as discussed in the lecture.

Application We then use the obtained depth map to create an artificial Bokeh effect, by applying a blur effect to all everything closer or farther than an arbitrary chosen depth.

With this approach, the accuracy of the depth estimation depends highly on the accuracy, numerical error and sampling frequency of the sensors.

The targeted platform is Android and the technical implementation will be done using java and C++ with the aid of the library OpenCV for various tasks, such as camera calibration or feature detection/matching.

2 Milestones

Week 24: Official project start: Project setup
Week 25: Sensor Data Processing and Image Rectification
Week 26: Stereo Matching
Week 27: Bokeh effect
Week 28: Feature Matching/Optimization
Week 29: Testing
Week 30: Create Poster

3 Requirements

Preferably, a multi-camera smartphone which offers an API for that would be good (LOGICAL_MULTI_CAMERA[1]) , but we are able to adapt if that is not possible. Alternatively, an USB camera which is supported by android[2] could be used to do traditional stereo vision. A Tango smartphone would be nice to benchmark our system against a full mobile depth camera.

4 Team

- Iulia - Otilia Mustea
- Manuel Schwonberg
- Jeremias Neth
- Michael Wechner

References

[1] Android 9 multicamera support. `https://source.android.com/devices/camera/multi-camera`.

[2] A. Fusiello, E. Trucco, and A. Verri. *A compact algorithm for rectification of stereo pairs*, pages 16–22.

[3] P. Heise, S. Klose, B. Jensen, and A. Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *2013 IEEE International Conference on Computer Vision*, pages 2360–2367, Dec 2013.

[4] Manon Kok, Jereon D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. In *Foundations and Trends in Signal Processing: Vol. 11 No. 1-2*, pages 1–153.

[5] Charles Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 125–131. IEEE, 1999.

[6] Christoph Rhemann Michael Bleyer and Carsten Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference*, pages 14.1–14.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.14.

[7] Kurt Seifert and Oscar Camacho. *Implementing Positioning Algorithms Using Accelerometers*, February 2007.

[1] Not all phones with Android 9 and stereo cameras actually support this: https://developer.android.com/reference/android/hardware/camera2/CameraCharacteristics

[2] https://source.android.com/devices/camera/external-usb-cameras