# Geometry measurements with image processing and forward projection


Steffen Reichel*, Karlheinz Blankenbach, Christian Link*, Michelle Winkle*

** Hochschule Pforzheim, Tiefenbronner Str. 65, D-75175 Pforzheim, Germany*

mailto:steffen.reichel@hs-pforzheim.de



We describe a method for determining the geometry (distance & lengths) of an object at arbitrary distance by just taking a single picture. Only once the distance and lengths of a reference object must be measured. After camera calibration and using forward projection followed by image processing an arbitrary object can be measured (distance & length) from a single picture at arbitrary distance.


## 1 Introduction

The automatic inspection of 2D or 3D geometries are used e. g. for component control in production processes. Often, such a visual inspection uses cameras and image processing to measure geometries. Geometry measurement methods use e. g. triangulation or stereo cameras.

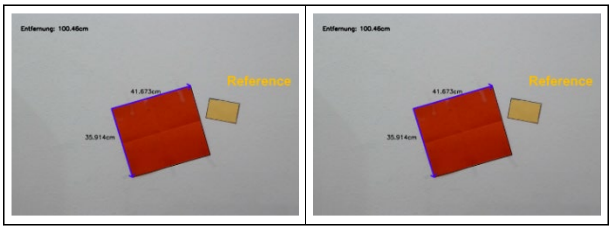


***Fig. 1*** *Illustration of length and distance measurement taking a single picture: at 100 cm distance (left) and at 150 cm distance (right, both measured). The numbers show the results of our calculation shown in this paper [1].*

We propose a new approach based on our presented camera calibration using re-projection as well as forward projection [3], [4]. After calibration, the 2D image coordinates of a 3D object are forward projected into the 3D world. Only once the distance and one dimension of a 3D object needs to be measured. For different distances the scaling value $s$ from our forward projection algorithm indicates the new distance and thus we are able to calculate the geometry of the object, see Fig. 1. Even if the object is translated or rotated, the forward projection can predict the geometry of the object.

## 2 Camera model and basic approach

We use a pinhole camera model including 5 distortion parameters implemented in OpenCV [2]. An image given in 2D image sensor coordinates $(u, \mathrm{v})^T$ corresponds to the 3D camera coordinates $(x_C, y_C, z_C)^T$ in a pinhole model according:

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_C \\ y_C \\ z_C \end{pmatrix} \qquad (1)$$

where $s$ is a scaling factor describing the loss of depth information, $f_x$ (and $f_y$) is the focal length in x-direction (y-direction, respectively) and $(c_x, c_y)^T$ is the principle point. We use a 5-parameter distortion model ($k_1$, $k_2$, $k_3, p_1, p_2$ - see [2]) since more parameters have no benefit as we have shown previously [4], [5]. Camera calibration is done using standard re-projection error and using forward projection error. The scaling factor $s$ is a side product by numerically solving the forward projection problem (see [3] for details).

Our basic approach is illustrated in Fig. 2: the distance of a reference object is measured only once together with its lateral dimensions (sizes).

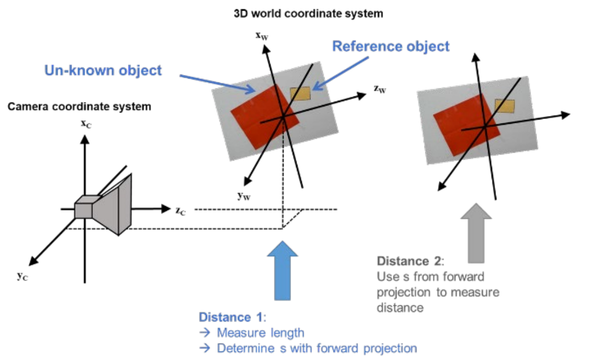


***Fig. 2*** *Basic approach for measuring lengths and distance of an object at arbitrary distance.*

From this single measurement we determine the parameter $s$ (Fig. 2 "Distance 1"). Different distances have different $s$ parameter obtained from forward projection (Fig. 2 "Distance 2") and thus the lengths and distance can be calculated: e.g. if parameter $s$ doubles than the real world distance doubles. A reference object is needed for lengths (not distance) calculations. Our proposed method has the following advantages:

- Rectification of device-under-test (DUT = un-known object) for accurate lengths measurements is done
- DUT can vary in distance
- Simple method

## 3 Experimental results

In order to test our proposed measurement method, we performed different experiments. We used a simple webcam (Logitech BRIO 4K STREAM, 4096 x 2160 / 30 frames per second 4K Ultra-HD in video mode, bought July 2022) and due to its simplicity, this webcam is ideal to test the idea if rectification and measurement. The camera calibration delivered the following parameters:

Camera matrix $\begin{pmatrix} 2780.80 & 0 & 1886.65 \\ 0 & 2779.73 & 1088.12 \\ 0 & 0 & 1 \end{pmatrix}$ and distortion coefficients $[k_1, k_2, k_3, p_1, p_2] =$ [0.16225384,-0.438159,-0.00098472,-0.002685, 0.23963348].

For every object the rotation and translation vectors and forward projection was calculated giving the parameter $s$ (and distance). Then, image processing (program flow in Fig. 3) was performed.

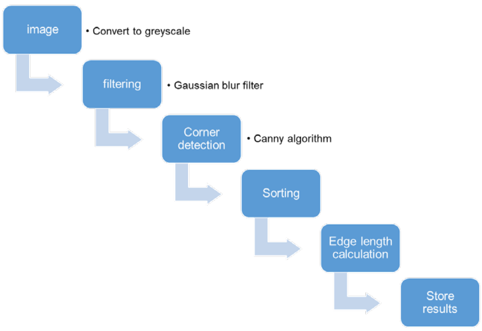


***Fig. 3*** *Program flow chart of our image processing with Python for geometry detection and edge length calculation.*

In a first test a triangle was measured, see Fig. 4 (left, indicated by arrow).

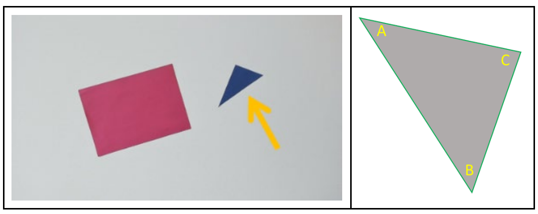


***Fig. 4*** *Triangle to be measured together with reference object (reddish quadrangle), left. Right: corner definition A, B, C of the triangle.*

First, we tested if the scaling factor $s$ can be used as distance measurement. The results show Fig. 5.

Factor: 1.555

| Measured distance | 900 mm | 1000 mm | 1200 mm | 1400 mm |
|---|---|---|---|---|
| Scaling factor s | 889.73 | 999.65 | 1184.41 | 1380.95 |

*s* from this distance

Factor: 1.552

***Fig. 5*** *Results for a triangle: reference distance is 900 mm ($s = 889.73$).*

For 900 mm (measured) the scaling factor $s$ was used as reference. At 1400 mm a $s = 1380{,}95$ was calculated which corresponds to a factor of 1.552 to the 900 mm reference. The factor from 900 mm to 1400 mm is 1.555, which is nearly identical to the factor in s. This demonstrates that or method works.

Next, we tested the lengths (edges) calculation for the triangle at 1490 mm distance (measured). The results from measurement and calculation shows Tab. 1.

| | Edge AB | Edge BC | Edge CD | Distance |
|---|---|---|---|---|
| calculated | 24.38 mm | 20.05 mm | 24.26 mm | 148.34 mm |
| measured | 24.00 mm | 20.00 mm | 24.00 mm | 149.00 mm |
| deviation | 1.6 % | 0.3 % | 1.1 % | 0.5 % |

***Tab. 1*** *Comparison of results for a triangle at 1490 mm distance, compare with Fig. 4 (right).*

As shown in Tab. 1 the maximum deviation is 1.6 % with this simple webcam. For lengths calculation we tested also different geometries and distances. All experiments showed that our proposed method is also suitable for lengths calculation.

## 4 Summary and conclusion

As shown, our basic idea works: a scaling factor $s$ can be used for distance calculation. The edge length calculation works as well also at different distances with a measurement uncertainty of less than 1.6 % using a simple webcam. In future, we will repeat our experiments, but with a better camera (lower distortion). We expect lower measurement uncertainties.

## References


[1] M. Caycioglu, A. Serzysko, M. Özkan: "Geometrievermessung mit einer Kamera unter Python und OpenCV", project work, Pforzheim University, Jun. 2022

[2] OpenCV Documentation on Camera Calibration and 3D Reconstruction, available online: *https://docs.opencv.org/4.x/d9/db7/tutorial_py_table_of_contents_calib3d.html* (accessed on 19 July 2024)

[3] A. Pak, S. Reichel, J. Burke, "Machine-Learning-Inspired Workflow for Camera Calibration", Sensors 2022, 22, 6804. *https://doi.org/10.3390/s22186804* (2022)

[4] S. Reichel, J. Burke, A. Pak, T. Rentschler, "Camera calibration as machine learning problem using dense phase shifting pattern, checkerboards, and different cameras", Proc. SPIE 12438, *https://doi.org/10.1117/12.2648304*, (2023)

[5] S. Reichel, P. Frank, K. Blankenbach, „Reliable and good camera calibration based on machine learning inspired workflow: parameter study and experimental results", Proc. SPIE 12903, *https://doi.org/10.1117/12.3000846*, (2024)