University of Ljubljana, Faculty of Electrical Engineering
Robot Vision (RV)

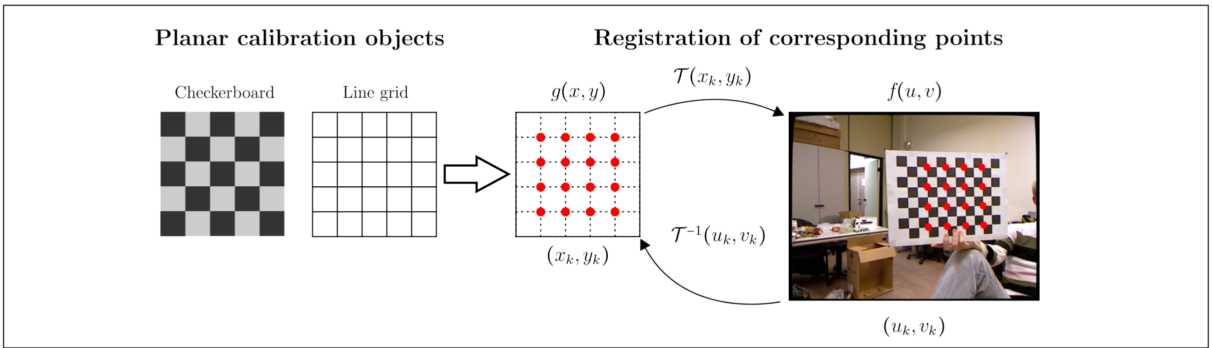
May 3/4, 2016

Exercise 7: Geometric Calibration of an Imaging System

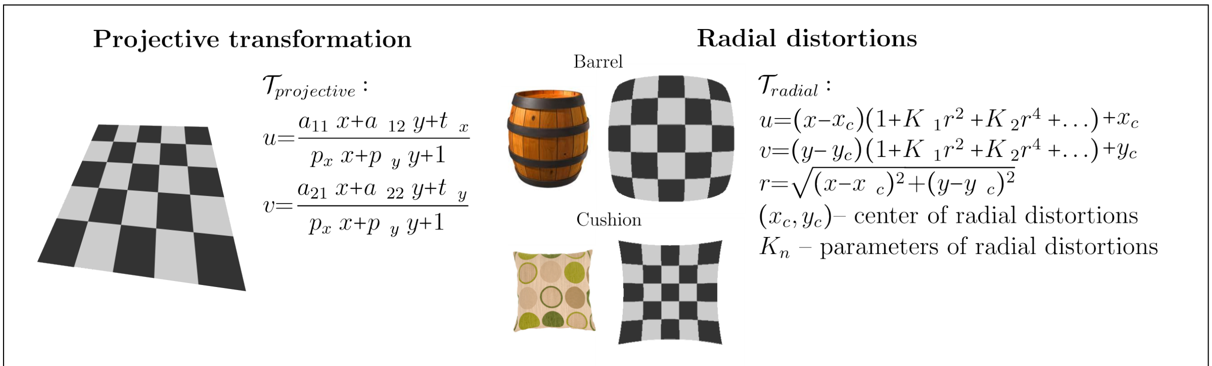
Created by: Žiga Špiclin | http://lit.fe.uni-lj.si/RV | Homework deadline: May 10/11, 2016

Instructions

Geometric calibration of an imaging system is used to define an absolute value of pixels in the image, which is otherwise dependent on the pose of objects in the image [1] and additionally distorted die to optical aberrations. Most common optical aberrations are radial distortions of type barrel or cushion. After the geometric calibration of an imaging system and a transformation of the acquired image into a metric space, one can perform dimensional measurements like sidelengths, area and volume of the imaged objects in absolute metric units (e.g. mm, mm^2, mm^3). For a geometric calibration of 2D imaging systems we may used **planar calibration objects** with simple, but high contrast patterns like **checkerboard, line grid or grid of small dots**. Geometric calibration involves a registration of pattern in the acquired calibration $f(u, v)$ and reference $g(x, y)$ images of calibration object. The most straightforward approach is to extract pairs of corresponding points on the calibration and reference images and use them to estimate the transformation parameters or registration, such that the locations of transformed calibration points are best matched with corresponding calibration points and vice versa.

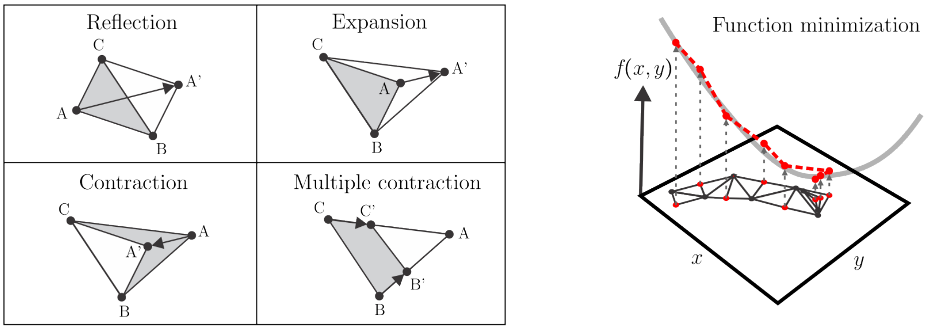


Most common transformations employed for a geometric calibration are **projective transformation**, which maps two arbitrary planes and is defined by 8 parameters (a_{11}, a_{12}, a_{21}, a_{22}, t_x, t_y, p_x, p_y) and can be written as a 3×3 matrix, and, to compensate for barrel or cushion distotions, a **Browns model of radial distortions**, defined by center (u_c, v_c) and coefficients K_i of radial functions. Geometric calibration is performed by transforming points from metric space into image space by projective transformation and then apply the model of radial distortions. Because Browns model of radial distortions is nonlinear an optimization method has to be employed to estimate its parameters. This is performed by minimizing a criterion function like mean square Euclidean distance $E^2 = \sum_{k=1}^{N} \|\mathcal{T}_{radial}\left(\mathcal{T}_{projective}(x_k, y_k)\right) - (u_k, v_k)\|^2$ between corresponding points in the image and metric space.



[1] e.g. due to the projection

Nelder-Mead simplex optimization



In order to obtain optimal parameters of the 2D projective transformation and model of radial distortions you will use **Nelder-Mead simplex optimization**. The simplex optimization is used to find a local optimum of an arbitrary function with respect to one or more parameters. Simplex is a primitive geometric structure used to sample the parametric space and, for example, is represented by a triangle for a function with two parameters $f(x, y)$. When searching for a local minimum of $f(x, y)$, the simplex optimization proceeds by moving those the simplex vertices, which have a high values of $f(x, y)$, in four different operations: 1) reflection, 2) expansion, 3) contraction and 4) multiple contraction. One operation is performed in one optimization step and the optimization process is performed for a predefined number of iterations or until the simplex shrinks below a certain minimal size. The location of local minimum is defined by the simplex's vertex corresponding to the minimal value of $f(x, y)$.

During this exercise you will write functions to correct for radial distortions and establish the 2D projective transformation. By annotating the locations of corners on the image of planar calibration object `calibration-object.jpg` and their transformation into the metric space through registration of corresponding points you will perform a geometric calibration of the imaging system. The image of a vernier caliper `test-object.jpg`, which was acquired by the same imaging setup as the image of calibration object, may then be transformed into metric space so as to measure distances on the caliper and thereby verify the calibration.

1. Write a function that transforms the input undistorted coordinates (u, v) in variables `iCoorU` and `iCoorV` using the Browns radial distortion model of arbitrary order n:

   ```
   def transRadial( iK, iUc, iVc, iCoorU, iCoorV ):
       return oCoorUd, oCoorVd
   ```

   where `iK` is a $1 \times n$ row-vector of parameters of the Browns model, while `iUc` and `iVc` represent the coordinates (u_c, v_c) of the center of radial distortions. Function should return the transformed *distorted* coordinates in variables `iCoorUd` and `iCoorVd`.

   Create a discrete grid of coordinate points using function `numpy.meshgrid()` and transform them with the Brown's model of radial distortions of order one ($n = 1$), using arbitrary parameter values. Display the input (u, v) and transformed (u_d, v_d) coordinates. Verify the impact of the center of radial distortions on the transformation of coordinate points. Which values of `iK` lead to barrel and which to cushion type of distortion? Which values of the parameters represent the identity transformation?

2. Write a function that transforms the input coordinates (x, y) in variables `iCoorX` and `iCoorY` with an arbitrary 2D projective transformation:

   ```
   def transProjective2D( iPar, iCoorX, iCoorY ):
       return oCoorU, oCoorV
   ```

   where `iPar` is a 1×8 row-vector consisting of parameters $(a_{11}, a_{12}, t_x, a_{21}, a_{22}, t_y, p_x, p_y)$. Function should return the transformed coordinates in variables `oCoorU` and `oCoorV`.

   Verify the function by transforming coordinates of the sampling grid of image `test-object.jpg` with some arbitrary parameters of projective transformation and then use the first order interpolation function `interpolate1Image2D()` to find corresponding intensity values in the transformed coordinates. Display the transformed image.

3. Test the simplex optimization by using Python function `scipy.optimize.fmin()` so as to find the local minimum of function $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$. As the initial parameter values use $x_0 = (1.2, 1)$. By setting additional parameters in the function call `fmin(func, x0, ...)` setup the display iteration information[2], the maximum number of iterations[3], function values tolerance[4] and parameter step tolerance[5]. Verify that the obtained minimum of $f(x, y)$ lies in $(1, 1)$ and observe the sequence of simplex operations.

4. Write a function to compute the mean square Euclidean distance between coordinates (u_k, v_k) in the acquired caliber image, which are transformed with Brown's model of radial distortions and corresponding coordinates (x_k, y_k) in metric space, which are transformed with the 2D projective transformation:

$$E^2 = \frac{1}{N} \sum_{k=1}^{N} \| \mathcal{T}_{radial} \left(\mathcal{T}_{projective}(x_k, y_k) \right) - (u_k, v_k) \|^2 .$$

   The first input parameters shoudl be a row-vector `iPar`, which represents the parameters of 2D projective transformation and the model of radial distortions as (a_{11}, a_{12}, t_x, a_{21}, a_{22}, t_y, p_x, p_y, u_c, u_v, K_1, K_2, ..., K_n).

   Other function's parameters should be the N coordinate pairs (u_k, v_k) $\rightarrow$ `iCoorU`, `iCoorV`, and the N coordinate pairs (x_k, y_k) $\rightarrow$ `iCoorX`, `iCoorY`:

```
def geomCalibErr( iPar, iCoorU, iCoorV, iCoorX, iCoorY ):
    return oErr2
```

   Function should compute the error E^2 between the transformed coordinates and return the value in a scalar variable `oErr2`. Employ the affine approximative registration `mapAffineApprox2D()` (Exercise 6) to obtain initial estimate of parameters (a_{11}, a_{12}, t_x, a_{21}, a_{22}, t_y).

   For the purpose of geometric calibration display the image `calibration-object.jpg` and manually annotate at least 8 corners on the calibration object using Python function `matplotlib.pyplot.ginput()`. Then create a reference grid of points that correspond to an ideal location of the corners on the planar checkerboard. The reference points should lie in a metric space, therefore you need to take into account that the sidelength of a square on the checkerboard is 20 mm. For instance, the corner in the upper left part of the checkerboard has coordinates $(20, 20)$ in the metric space. Find the optimal values of parameters `iPar` by minimizing the error E^2 through teh simplex optimization. Display the reference coordinates and the corner coordinates on the image before and after the transfromation with optimal parameters.

Homework Assignments

Homework report in the form of a Python script entitled `NameSurname_Exercise7.py` should execute the requested computations and function calls and display requested figures and/or graphs. It is your responsibility to load library packages and provide supporting scripts such that the script is fully functional and that your results are reproducible. The code should execute in a block-wise manner (e.g. `#%% Assignment 1`), one block per each assignment, while the answers to questions should be written in the corresponding block in the form of a comment (e.g. `# Answer: ...`).

1. Write a function that uses the obtained optimal parameters of geometric calibration `iPar` to transform the acquired `iImage` into metric space:

```
def geomCalibImage( iPar, iImage, iCoorX, iCoorY ):
    return oImage
```

   where variables `iCoorX` and `iCoorY` represent coordinates (x, y) in the **metric space**, which need to be transformed into image space in order to compute the corresponding grayscale values. Function should return the calibrated image `oImage`, the dimensions of which are given by the dimensions of the coordinates (x, y).

[2] `fmin(..., disp = 1, callback = fname)`, where `fname(X)` is a function to format the display
[3] `fmin(..., maxiter = 100)`
[4] `fmin(..., ftol = 1e-6)`
[5] `fmin(..., xtol = 1e-6)`

Verify the function by transforming the coordinates (x, y) using the optimal values of 2D projective transformation and Brown's model of radial distortions. Note that the sidelength of the checkerboard square is 20 mm. Create the sampling coordinates (x, y) in such that the output normalized or calibrated image `oImage` will have a pixel size of 1/5 mm.

Normalize the image `calibration-object.jpg` using first order interpolation `interpolate1Image2D()`. Display the transformed image and verify that the checkerboard pattern are aligned to image axes.

2. Repeat the geometric calibration three times, each time manually annotating the corners in the image `calibration-object.jpg`. Create a reference geometric calibration based on averaging the positions of the manually annotated corners. Use the reference geometric calibration to estimate the precision (in mm) of the geometric calibration based on manually annotated corners. What impacts the precision of the geometric calibration?

3. Use the color image `test-object.jpg`, which was acquired using the same imaging setup as the image of calibration object, to perform a calibration based on reference geometric calibration such that the pixel size will be 1/3 mm. Display the calibrated color image and measure the distance of the gap on the vernier caliper. You can do this by defining a line segment that spans across the gap. Verify that the obtained distance is similar to the distance marked on the vernier caliper.

4. Develop an automatic method for geometric calibration of the imaging system. Employ the Harris corner detector (Exercise 5) to localize the corners on the image of calibration object `calibration-object.jpg`. Create the ideal grid of reference corners and employ the ICP method (Exercise 6) to establish correspondences between the corners found on the image of calibration object and the grid of reference corners. Estimate the accuracy of automatic calibration method (in mm) with respect to the reference geometric calibration obtained in the first homework assignment.