Single pose camera calibration using a curved display screen

Raul Acuna[1], Robin Ziegler[1] and Volker Willert[1]

TU Darmstadt, Control Methods and Robotics Lab,
Landgraf-Georg Straße 4, 64283 Darmstadt

Abstract In this paper a method for single pose camera calibration is presented. Dense point correspondences are obtained by displaying structured light in a non-flat display screen (i.e. a curved display screen) which then are used as an input in common calibration algorithms. Experimental results demonstrate that the depth information present in a common commercial curved monitor with a radius of curvature of 1800 millimeters is sufficient in order to obtain calibration results comparable to the standard checkerboard method. In contrast with the commonly used checkerboard based calibration methods, the proposed method does not require to move the camera and it is a cheaper and easier to implement that other methods based on expensive 3D calibration rigs.

Keywords Camera Calibration, Structured Light, Display, Curved Screen

1 Introduction

Camera calibration is the first step for optical based 3D measurements [1], e.g. measure the size of an object in world units or determine the location of the camera in world coordinates. Camera calibration methods require the use of known control points in world coordinates which can then be correlated to image points in the camera. These control points are defined by using objects or shapes with known dimensions in world units, therefore either three dimensional or two dimensional calibration objects are used.

Three-dimensional calibration objects like the one used in [2] are usually complicated to manufacture (and measure) and it is generally hard to cover the whole image space with a good resolution of control points. To overcome these problems Zhang [1] proposed the nowadays most commonly used calibration technique which employs a checkerboard pattern on a flat surface. Since a single shot of a 2D calibration object can not provide enough constrains to calibrate the camera, the checkerboard must be captured by the camera in at least three different poses. The amount and size of squares on the pattern define the quantity of points that can be obtained in a single image and the range of the checkerboard detection, therefore in practice more than three captures (recommended more than 10) that cover the whole camera image space are required with different orientations and positions.

During the image capture process usually wrong measurements may be included due to the movement of the checkerboard (blurred images) or because an user with low experience in camera calibration may not understand how to properly place the checkerboard pattern to obtain good results. Additionally, the checkerboard is a fixed planar shape which has a limited optimal detection range, if the checkerboard is too close or too far away from the camera the measurements will have a worse quality than the ones taken at the optimal distance. Some calibration suites try to overcome this by using several calibration checkerboards with different sizes, which increases the complexity of the calibration process in favor of higher accuracy. Some camera calibration suites employ smart user interfaces that guide the user during the checkerboard image acquisition process and allow the removal of wrong captures.

Newer calibration methods use a LCD screen [3] that replaces the printed checkerboard. Using an LCD screen has several advantages, with the manufacturing capacity of nowadays LCD panels a resolution of over 1920×1080 pixels can be achieved with a constant pixel size less than 0.3 millimeters. This allows the precise presentation of any kind of calibration pattern, including a checkerboard, but also smaller patterns with more squares, phase-based shapes where more point correspondences can be obtained or other shapes robust to de-focusing [4]. However, using a flat screen requires still the capture of at least three different poses of the screen by the camera. Some single pose camera calibrations using flat screens involve the positioning of two or more

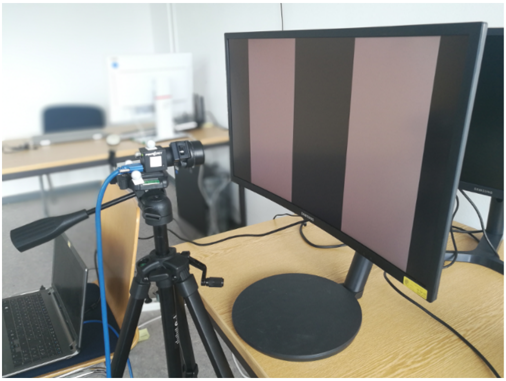


Figure 1.1: Example of our experimental setup for structured light based calibration of a camera using a curved display screen.

flat screens in order to create a 3D calibration rig [5], but the relative poses of the camera still have to be precisely calibrated incrementing the solution complexity.

In contrast, by using a curved screen, see Figure 1.1, the depth information may be added to every pixel without additional devices or additional motions. With a good point correspondence system established between pixels in the curved screen and the pixels in the camera image, the curved screen can be used as a 3-dimensional calibration rig with a high amount of control points. This could enable a calibration with only one camera pose reducing the amount of user interactions to a minimum. The question to answer is if the radius of curvature of a common commercial curved panel (18000 mm) is enough to provide the required depth information of the control points for a proper calibration and if the standard calibration algorithms can be used with this data. On the following section we propose and evaluate a calibration method which demonstrate that a it is in fact possible to perform camera calibration with a curved screen with comparable performance to the state of the art.

2 Proposed calibration method

Our calibration work-flow is based on the flat screen approach but with some modifications to suite the special case of a curved screen, it is shown in a general from in Figure 1.2.

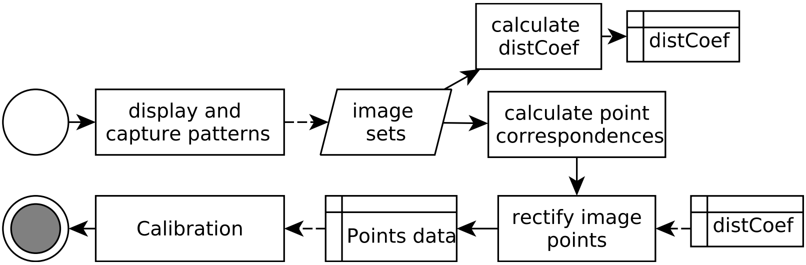


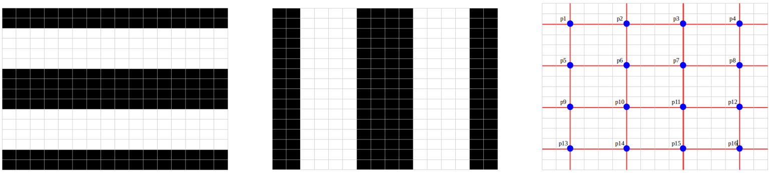
Figure 1.2: Proposed calibration method

2.1 Extraction of point correspondences

A time-multiplexing gray coded structured light technique was chosen due to its simple implementation and decoding. This method consists on the successive projecting of patterns on the display. First, horizontal and then vertical ones. The Gray code pattern generation was implemented using standard methods from the *OpenCV* structured light library [6]. Example images corresponding to the third horizontal and third vertical pattern are displayed in Figures 1.3(a) and 1.3(b).

This technique produces a mapping of several display pixels to a single pixel in the camera image due to the higher display screen resolution compared to the camera. The patterns can be detected only up to a specific resolution until the stripes can not be differentiated, see Figure 1.5(a). This problematic effect depends both on the relative camera-display resolutions and on the camera-to-display distance.

Screen pixel coordinates are denoted by $[U, V]$ and camera image pixel coordinates as $[u, v]$. The origin of the world coordinate system is defined in the center of the screen. The total amount of gray coded horizontal and vertical images necessary to completely codify a screen with a resolution of $[width, height]$ pixels are:



(a) (b) (c)

Figure 1.3: Process of display point decoding example. a) Third horizontal pattern, b) Third vertical pattern, c) Combined edge maps with marked intersection points.

$$n_u = log_2(width), n_v = log_2(height). \tag{1.1}$$

In order to overcome the resolution problem instead of displaying all the n_u and n_v images we show only the patterns up to some maximum horizontal and vertical density parameters D_u and D_v, each one defines the maximum value used in the gray coded patterns in the horizontal and vertical direction respectively. Additionally, the current displayed pattern number is denoted by d. For example, if we set $D_u = 3$ only the patterns $d = 1$, $d = 2$ and $d = 3$ are generated, displayed and captured in the horizontal direction.

For each horizontal or vertical pattern, the edge lines are extracted and for each one the gray code information of the previous pattern is used to match the line on the image to its correspondent on the display screen. The intersection of the pattern edges plus the gray code information are used to find the camera image to screen point correspondences. This approach is similar to the one proposed by [7].

The width of the black and white columns in a Gray pattern d will be half of the width of the previous Gray pattern $d - 1$, see Figure 1.4. As consequence the edge of a row/column in a pattern d will be located exactly on the middle of the row/column of the previous pattern $d - 1$. Hence, each edge on camera image for the current d can be tagged with the gray code of the previous pattern $d - 1$, the gray code of a given horizontal or vertical edge will be denoted by e_u or e_v. For example in Figure 1.4 the Gray code corresponding to the third edge from left to right is $e_u = 11_g$.

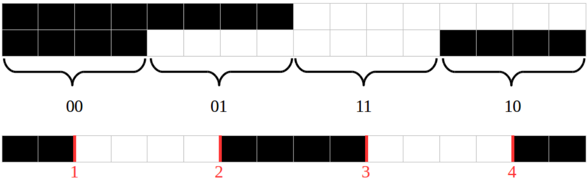


Figure 1.4: Column extraction process with $d = 3$ and a resolution of 16 pixels, *Gray Code*.

We can use this information to do the line correspondence between image captured edges and screen pattern edges and obtain its screen pixel coordinates by applying the following equations:

$$U = ((1 + G'(e_u)) - 0.5)w_u, \quad V = ((1 + G'(e_v)) - 0.5)w_v \tag{1.2}$$

Where the function $G(x), x_{dec} \mapsto x_{gray}$ represents the conversion from decimal values to Gray coded values and its inverse is denoted as $G'(x)$. For a given pattern number d, the width of a column is defined as w_u for horizontal patterns and the width of a row as w_v for vertical patterns. Both w_u and w_v are in screen pixel coordinates and calculated by the following equations:

$$w_u = \frac{2^{n_u}}{2^{d-1}}, \quad w_v = \frac{2^{n_v}}{2^{d-1}} \tag{1.3}$$

Finally, we obtain the world coordinates of the point by using the radius of curvature ρ and the pixel pitch of the screen in x and y directions ϕ_x, ϕ_y (values provided by the screen manufacturer):

$$\upsilon = (U - width/2)\phi_x, \tag{1.4}$$
$$\varphi = (V - height/2)\phi_y, \tag{1.5}$$
$$[X, Y, Z] = [sin(\upsilon/\rho) * \rho, \ -\varphi, \ cos(\upsilon/\rho) * \rho - \rho]. \tag{1.6}$$

The accuracy of the point extraction process depend mainly on three factors: 1) external light sources, 2) camera placement and 3) performance of the edge detector, see Figure 1.5. It is recommended to place the screen in a place where no reflections of other light sources can interfere with the camera capture and the ideal placement to cover the screen without degradation of the line detection is fronto-parallel, with the screen covering as much as possible area on the camera image.

To improve the performance of our edge detection an upscaling of the raw images using the Gaussian pyramid method and then the application of a Gaussian convolution kernel to estimate the values of the new missing pixels is performed. This results in a high resolution blurred image. A canny edge detector is then applied to the resulting image. Additionally, the display of a full black followed by a full white image on the screen is used to calibrate the thresholding of the patterns which allows an accurate recognition of the contour of the panel and the definition of a region of interest in the camera image for the processing.

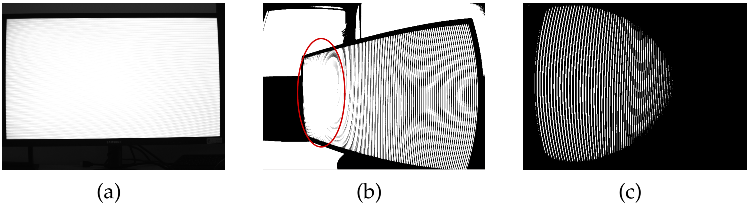


Figure 1.5: Potential problems in image acquisition. a) Gray Coded pattern with a resolution that can't be resolved by the camera, b) Influence of other light sources, c) Effect of an inclined camera placement.

2.2 Non metric image rectification preprocessing

The obtained points produced by the point correspondence system can be used as input for a classical camera calibration algorithm (e.g. the OpenCV calibrate camera function). Empirical results with different lenses showed that for points obtained by placing the camera directly in front of the curved monitor the calibration process produces wrong results. It is our guess that the calibration method can not differentiate between the lens distortion and the screen curvature. It is possible to obtain good calibration results by using a steeper camera angle relative to the screen, which increases the relative depth change. However, the more the camera is tilted, the less point correspondences are obtained in the far field. Hence, our proposed solution consists in a preprocessing step, in the form of a non metric image rectification.

A non metric rectification method exploits the constraint that straight lines in the scene must remain so in the projected space [8]. A least squares approach is used to find the distortion parameters, first the distortion error is defined as the sum of squares distances from the detected edge points to an approximated line distorted by some initial distortion parameters. This distortion parameters are then optimized iteratively by minimizing the error.

We used this approach in combination with the one parameter division model proposed by Fitzgibbon [9], where the distortion is described in camera coordinates as:

$$x_d = x\frac{1}{1+k_1r^2}, \quad y_d = y\frac{1}{1+k_2r^2}, \tag{1.7}$$

where x_d, y_d are the distorted camera coordinates, x, y the undistorted ones, $r = \sqrt{x^2+y^2}$ and k_1, k_2 are the distortion parameters to be found. Since we only have vertical straight lines (horizontals are curved) we assume that $k = k1 = k2$. In order to not confuse this distortion parameter with the produced by OpenCV or Matlab calibration suites when performing the calibration we are going to denote k as γ. And then in camera pixel coordinates equation becomes:

$$u_d = (u-u_0)\frac{1}{1+\gamma_1r^2} - u_0, \quad v_d = (v-v_0)\frac{1}{1+\gamma_1r^2} - v_0, \tag{1.8}$$

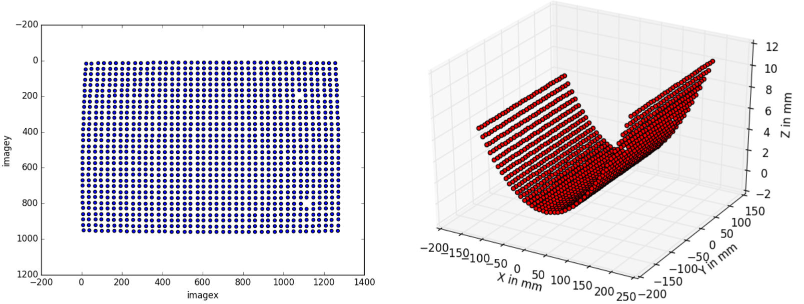
with $u_0 = width/2$ and $v_0 = height/2$. For simplification reasons, the center of distortion is assumed to be the center of the image which is a good approximation.

Once γ is found the edge intersections in camera coordinates are rectified and together with the world coordinates are used as inputs for the camera calibration functions of Matlab and OpenCV.

3 Results and discussion

To test the performance of our calibration method a curved screen with a radius of curvature of 1800 millimeters was used. The cameras used was a *FLIR Blackfly BFLY-U3-13S2C-CS* from Point Grey with a resolution of 1288×964 pixels. Three different focal lengths were selected. For each focal length the camera was placed in front of the screen so the whole display area was within the camera image and the point correspondences, non metric rectification and calibration with both Matlab and OpenCV was performed. As a comparison we also calibrated the camera using the traditional checkerboard method and with a planar LCD screen (3 views of the planar screen).

An example of the point correspondences obtained with our method when using a density parameter $d = 7$ is shown in Figure 1.6. Notice the density of the acquired points and their regularity.



(a) Intersection points in camera image.

(b) Corresponding points in display coordinates.

Figure 1.6: Point correspondences obtained from a curved screen with a camera perpendicular to the screen ($d = 7$).

The camera intrinsics results obtained for the 4mm focal length for all the compared methods are shown in Table 1.1. The error is the reprojection error, calculated using the following equation:

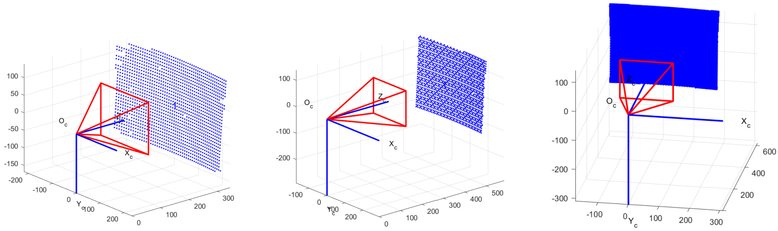
$$error = \sqrt{\frac{\sum_{i=1}^{N}(x_i - x_i')^2 + (y_i - y_i')^2}{N}}, \tag{1.9}$$

where x_i', y_i' are the reprojected image point coordinates and N the total amount of points, this is default error on OpenCV and was calculated as well from the data of the Matlab Calibration Toolbox (which in contrast uses by default the standard deviation of the pixel error).

Parameter	Curved screen *(OpenCV)*	Curved screen *(MATLAB)*	Flat screen *(Matlab)*	Checkerboard *(OpenCV)*
fx	1071.912	1084.156	1065.406	1074.786
fy	1072.538	1084.777	1054.065	1074.958
cx	632.110	632.444	628.077	627.898
cy	508.721	508.295	495.022	493.513
k1	0.0312	0.02234	-0,35959	-0.39147
k2	-0.0655	-0.03323	0,13475	0.24112
k3	0.0051	0.00522	0	-0.63883
p1	-0.0045	-0.00452	-0,00014	-0.62106
p2	0.0316	0	0,00013	-0.10542
γ	-3.607e-07	-3,607e-07	0	0
Error	0.548282	0.5524	0.51886	0.33515

Table 1.1: Camera parameters obtained using the curved screen compared to other calibration methods.

A comparison of the calibration results for different focal lengths in terms of the reprojection error are show in Table 1.2 and the graphical calibration results of the *Matlab Calibration Toolbox* are shown in Figure 1.7.



(a) 4mm.

(b) 7mm.

(c) 8mm.

Figure 1.7: Point correspondences obtained from a curved screen with a camera perpendicular to the screen ($d = 7$).

The curved screen approach achieves sub-pixel reprojection errors, with values comparable to the 3-views flat screen approach and higher

Focal length	Curved Opencv	Curved Matlab	checkerboard
4mm	0.5482	0,5524	0,33515
7,5mm	0,4670	0.4671	0,33411
8,4mm	0.4808	0.4812	0,32209

Table 1.2: Re-projection error in px units for different focal lengths using the curved screen method.

than the checkerboard method. Which is remarkable considering that only one camera pose is needed. Comparing the camera intrinsics obtained with the curved screen to those of the checkerboard we can notice that the parameters corresponding to the x axis of the camera present better results than the ones on the y axis. This could be a product of the lower amount of depth present in that axis in the curved screen, more test are required with different display orientations in order to test this dependency. It is possible to observe that the $k1$ parameter is reduced considerably in the curved screen results, this is consequence of performing the one parameter division model rectification preprocessing step.

We notice in Table 1.2 that our method works better for larger focal lengths, we believe that it is due to the low radius of curvature (1800 mm) of our screen. It is possible that a screen with a higher radius of curvature will provide better results. In terms of speed our method is the fastest. The whole process of placing the camera in front of the screen, displaying and acquiring the images and calibrating takes less than one minute with minimal user interaction which compared to the checkerboard method is a remarkable advantage.

4 Conclusion

A single pose automatic camera calibration method was presented. The novelty of the approach lies on the use of a curved display screen which allows the capture of dense point correspondences without moving the camera. Since traditional methods will fail when directly using this data due to the screen curvature our pipeline introduces a preprocessing step in the form of a non-metric rectification using a the single parameter division model. Experimental results confirm, that our method achieves sub-pixel reprojection errors and has comparable performance to the standard checkerboard method with the added advantage of not having to move the camera. The whole process is made in an automatic way by the software with a minimum of user interaction. However, there is still great margin for improvement. First, the refraction of the screen has not been considered which has been tackled in flat screens based methods with an increase in accuracy. Phase based patterns may be used instead of the gray code approach which may give denser and more accurate points and finally, a more complex distortion model can be employed for the preprocessing rectification step.

References

1. Z. Zhang, "A Flexible New Technique for Camera Calibration (Technical Report)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2002.
2. J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066–1077, 2000.
3. F. Forster, "Camera calibration: active versus passive targets," *Optical Engineering*, vol. 50, no. 11, p. 113601, 2011.
4. B. Cai, Y. Wang, K. Wang, M. Ma, and X. Chen, "Camera calibration robust to defocus using phase-shifting patterns," *Sensors (Switzerland)*, vol. 17, no. 10, 2017.
5. E. Grossmann, J. Woodfill, and G. Gordon, "Display screen for camera calibration," *US Patent*, 2011.
6. Q. Gu, K. Herakleous, and C. Poullis, "3DUNDERWORLD-SLS: An Open-Source Structured-Light Scanning System for Rapid Geometry Acquisition," *arXiv*, no. May 2015, 2014.
7. R. Sagawa, M. Takatsuji, T. Echigo, and Y. Yagi, "Calibration of lens distortion by structured-light scanning," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, no. Lcd, pp. 1349–1354, 2005.
8. El-Melegy and Farag, "Nonmetric lens distortion calibration: closed-form solutions, robust estimation and model selection," in *Proceedings Ninth IEEE International Conference on Computer Vision*, no. Iccv. IEEE, 2003, pp. 554–559 vol.1.
9. A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–125–I–132, 2001.