A Screen-Based Method for Automated Camera Intrinsic Calibration on Production Lines

Wenliang Gao, Jiarong Lin, Fu Zhang, and Shaojie Shen

***Abstract*— For the manufacture of visual system product, it is necessary to calibrate a massive number of cameras in a limited time and space with a high consistency quality. Traditional calibration method with chessboard pattern is not suitable in the manufacturing industry since its requirement of motions leads to the problem of consistency, cost of space and time. In this work, we present a screen-based solution for automated camera intrinsic calibration on production lines. With screens clearly and easily displaying pixel points, the whole calibration pattern is formed with the dense and uniform points captured by the camera. The calibration accuracy is comparable with the traditional method with chessboard pattern. Unlike a variety of existing methods, our method needs little human interaction, as well as only a limited amount of space, making it easy to be deployed and operated in the industrial environments. With some experiments, we show the comparable performance of the system for perspective cameras and its potential in fisheye cameras with the developments of screens.**

I. Introduction

Visual systems are used in industrial applications in the last decade. With the developing of autonomous technologies, vision-based systems have been widely used in moving robots such as autonomous driving and unmanned aerial vehicles. For vision-aid productions, the camera calibration is one of the most essential techniques. It is necessary to develop a low-cost flexible calibration approach which works autonomously, robustly, and need no specially trained staff for manufacturing industrial.

Traditional camera intrinsic calibration are using some specially designed 2D patterns [1]–[9], or special objects [10]–[13]. There are some disadvantages existing: 1) It is difficult to sample on areas near the edges of the image. The sample points provided by the calibration object is sparse and sensitive for the partial occlusion. The object detection is also influenced by the larger distortion near the edges. Which cause nonuniformity of samples and the coverage is not enough. Samples distribute densely in the center and sparsely near the edges. 2) The image capture and relative movements of poses are controlled manually during calibration. The distribution of samples depends on the operator. Thus, calibration results are not consistent by lacking the standard of movements and influenced by the ability of operators; 3) Images are captured during relative motion sometimes that will cause blur. This effect deteriorates the accuracy of sample point detection. Especially for rolling shutter cameras, which are widely used in manufacture for their low cost.

W. Gao,and S. Shen are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR., China. wenliang.gao@connect.ust.hk,eeshaojie@ust.hk. J. Lin, F. Zhang are with the Department of Mechanical Engineering, Hong Kong University, Hong Kong SAR., China.jiarong.lin@hku.hk, fuzhang@hku.hk.

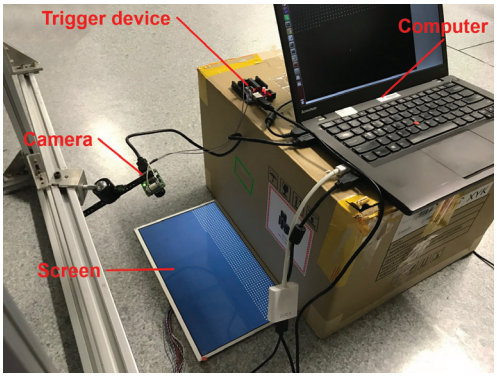


Fig. 1: Screen calibration system prototype.

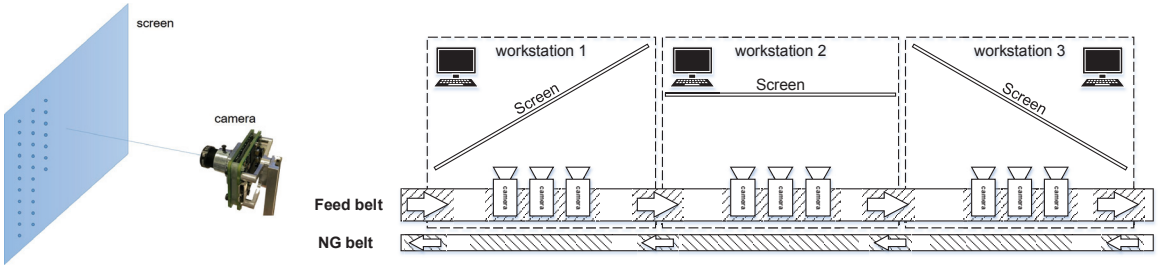
Using a robot arm to repeat the calibration operation is a reasonable solution which satisfies the demand of production capacity. Robots carry the camera or calibration pattern, execute specially designed action sequences, and collect images at some particular poses. This kind of approaches is to repeat the regularize trustworthy human operation automatically [14, 15]. However, it is inflexible and inconvenient because industrial robots require a long deployment cycle to redeploy for different products. A special space and expensive cost are demanded by industrial robots in factories.

Our proposed method is tailored for industrial automation: low-cost, limited amount of space, effective and easy to set up in industrial factories.

The overview of proposed automatic camera-calibration in production lines is constructed with calibration workstations, as shown in Fig. 2. A camera to be calibrated is conveyed into a workstation then fixed automatically and mechanically. The camera connects to the PC via the debugging port as Fig. 5 shows. Then capture images as introduced in Sect. VI-A. Once the sampling process finished, the connection between the PC and camera break and the data sample quality evaluations in Sect. V are executed. Intrinsic calibrate using the samples if the evaluation test passed. If the test failed, the camera would be regarded as “not good (NG)” and removed from the production line. One workstation handles one particular relative rotation of calibration. Since the calibration needs at least three relative rotation, three or four workstations are built in the system.

In this work, we proposed a quick method to evaluate the sample quality for calibration via the coverage rate and distribution uniformity in Sec. V. We present a camera

978-1-7281-0356-3/19/$31.00 ©2019 IEEE



(a) Screen calibration workstation (b) Calibration workflow in the manufacturing industry

Fig. 2: (a) The rotation and translation between the camera and screen are mechanical fixed, and initially estimated mechanically. The screen shows rendered images with blobs. (b) An illustration of calibration workflow in manufacturing industry. A sequence of calibration stations with different rotations form the whole calibration production line.

calibration station with screens, which is able to provide high-quality samples with uniform distribution and coverage of the whole image in Sec. VI. In Sec. VII, experiments certificate the calibration accuracy is comparable with the transitional chessboard method for perspective cameras. We also show the potential for fisheye cameras. Furthermore, our method provide a kind of dynamic sampling for some irregular cameras, such as event cameras [16, 17]. We identify our mainly contributions as follows:

- An evaluation method of sample quality for calibration via coverage rate and distribution uniformity.
- A high-quality sample method that guarantees uniformity and coverage.
- A hardware and software calibration system prototype tailored for manufacturing automation.

II. Related Work

Camera calibration is a key element of 3D computer vision. The accuracy of intrinsic parameters is required by all vision-based systems to calculate the extrinsic parameters and other geometry computer vision properties [1]. One of the wide-used methods is proposed by [2], which capture a sequence of images with a 2D planar pattern in different orientations, then determine the optimal intrinsic parameters by minimizing the reprojection error between observation points and the corresponding predictions. In [3], the author proposed a method to solve the partial occlusion problem of the calibration object by tracking the random dots markers on a plane. The control points are first localized and then used to estimate the camera parameters.

More methods are proposed to improve the calibration in the last decade. Instead of using a simple 2D chessboard, they use other kinds of objects. Circular patterns are used to improve the precision and reduce the preparation of calibration patterns [4]–[7]. The spherical object is flexible and easy to use since it is efficient to calibrate multiple cameras simultaneously [10]–[13]. The point (tiny circle) gets smallest error induced by the distortion for usual 2D calibration patterns methods, including centroid, conic, point, and edge [18]. Furthermore, to reduce the number of images needed, other properties besides geometry can also be used for calibration. For example, by using a particular object, whose observed color changes via the viewed angle, the researcher can calibrate their cameras by just capture using a single image [19].

The calibration procedure of well-known visual systems is still complex for the manufacturing industry. To guarantees an accurate calibration without any human intervention, while requiring only a limited amount of space, screens are used in calibration. Some methods directly show a rendered virtual 2D pattern by the screen [20] instead of moving a real chessboard. Show a blurred pattern in the screen and calibration by the defocus images is also a solution [21, 22]. A configuration with a display panel and a thin opaque sheet with a grid of holes are used for calibration in [23]. In theses methods, it is absolutely necessary to carry cameras shooting images for the calibration pattern from different points of views. The operations are always time-consuming, and choosing a set of views with suitable rotation is important. Moreover, it should be a trade-off carefully: the rotation angles between the camera and the calibration pattern are neither large nor small. For small rotation angles, the camera calibration will fail due to the degenerate configuration. For large rotation angles, the camera calibration accuracy cannot be guaranteed because of imprecise corner detection results.

However, the coverage of the sample points is hard to guarantee since the to detect the pattern and find its correspondence at the edge of the image is not easy. Our proposed approach does not have these problems with its properties: dense and uniform sampling, fixed relative pose during one operation, small space required, low-cost, and modular and easy to deploy.

III. Calibration in automatic production lines

Cameras calibration is one of the most important and complicated processes in manufacturing products with mounted cameras. It requires sophisticated mechanical structures (i.e., manipulators, 3-axis rotation platform, etc.) to take cameras to capture images in various point of views. What's more, to meet the massive demand of Units Per Hour (UPH), the procedure of camera calibration should be fast and easy to be parallel. Last but not least, compared to other scenarios such as hand-hold calibration, automatic production lines pose a higher requirement on both reliability and maintainability on the process of calibration.

To address the above problems, we propose the screen-based camera calibration approach, which can not only meet the requirements of mass production, but also have advantages on data sampling, mean the characteristics of higher reliability and maintainability. Compared the previous method, we have the following advantages.

- *Easy to setup and maintain*: without sophisticated mechanical structures, the pipeline of (shown in Fig. 2(b)) our method is much simpler than the previous methods, which calibrated using manipulator [14, 15]. What's more, the requirement of around illustration is much simpler compared to chessboard pattern since the screen is the light source itself.
- *Ready for parallels:* as the figure shown in Fig. 2(b), several cameras are calibrated at the same time, and the pipeline structure can effective decreases the stall time, therefore boost the productivity.
- *Data-efficient:* since we can control the screen's display and the trigger of camera capture, therefore our data sampling procedure in more efficient and reliable. We will expand more details in the following sections.

IV. Camera Model

In this section, we introduce the perspective camera model and the polynomial-based fisheye camera model considered in the calibration. The projection process of a point $\boldsymbol{\mathcal{P}}^w$ in the world frame w is formulated as:

$$\boldsymbol{\mathcal{P}}^c = \mathbf{R}_w^c \boldsymbol{\mathcal{P}}^w + \mathbf{T}_w^c, \tag{1}$$

$$\mathbf{u} = \pi(\boldsymbol{\mathcal{P}}^c), \tag{2}$$

where $\mathbf{R}_w^c$ and $\mathbf{T}_w^c$ is the transform between camera frame and world frame, $\boldsymbol{\mathcal{P}}^c = [x^c \ y^c \ z^c]^T$ is the point in camera frame. $\mathbf{u} = [u \ v]^T$ is the image in pixel coordinate. $\pi(\cdot)$ is the projection function of the camera.

A. Perspective Camera Model

As for perspective cameras, the projection is formulated in [24] as:

$$\mathbf{u}' = [u' \ v']^T = [x^c/z^c \quad y^c/z^c]^T, \tag{3}$$

$$\mathbf{u} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} (d_r \mathbf{u}' + d_t) + \begin{bmatrix} c_x \\ c_y \end{bmatrix}, \tag{4}$$

where the $r = u'^2 + v'^2$. f_x and f_y is the focal length in x and y axes. $[c_x \ c_y]^T$ is the principal point. d_r is the radial distortion and d_t is the tangential distortion with parameters k_1, k_2, p_1, and p_2:

$$d_r = 1 + k_1 r + k_2 r^2, \tag{5}$$

$$d_t = \begin{bmatrix} 2p_1 u'v' + p_2(r + 2u'^2) \\ p_1(r + 2v'^2) + 2p_2 u'v' \end{bmatrix}. \tag{6}$$

For perspective cameras, there are eight intrinsic parameters:

$$\kappa = [f_x \ f_y \ c_x \ c_y \ k_1 \ k_2 \ p_1 \ p_2]^T. \tag{7}$$

B. Fisheye Camera Model

As for fisheye cameras, the projection is combined with a polynomial-based radial distance $\sum_i \eta_i \theta^i$ in [25]:

$$\mathbf{u} = \sum_i \eta_i \theta^i \begin{bmatrix} f_x & \alpha \\ 0 & f_y \end{bmatrix} \begin{bmatrix} cos(\varphi) \\ sin(\varphi) \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}. \tag{8}$$

where f_x and f_y is the focal length in x and y axes and α is the affine parameter. θ and φ are the inclination and azimuth corresponding to $\boldsymbol{\mathcal{P}}^c$ [25]. Considering using seven order polynomial to formulate the distortion, there are 11 intrinsic parameters:

$$\kappa = [f_x \ \alpha \ f_y \ c_x \ c_y \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5 \ \eta_6 \ \eta_7]^T. \tag{9}$$

The tangential distortion of the fisheye cameras is ignored since the tangential distortion is much smaller than the radial distortion, as described in [26].

V. Sample Quality Evaluation

In the manufacturing industry, each operation step should be controllable and traceable. To reduce the positive true and negative false case, a simple and efficient evaluation of sample quality is necessary.

The camera intrinsic calibration is to refine the parameters of a camera model which describes the relationship between the 3D world points and the 2D pixel coordinates. The data for camera intrinsic calibration is a series of images capturing the manufactured calibration pattern. The 2D pixel coordinates are detected from the images and the 3D world points are known because of the artificially designed pattern. The domain of the image sensor is all pixels of the camera.

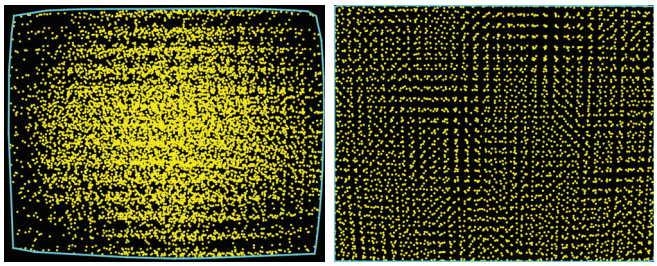
The essence of camera calibration is a nonlinear optimization problem. As for such kind of problem, the sample distribution influences the results by weight. The greater density of samples leads the greater weight. And obviously, a larger sample coverage of the domain leads to a better description of the model. The samples for calibration should cover the image pixel domain as much as possible and as more uniform as possible. We evaluate the sample quality using the sample coverage and sample uniformity.

a) Sample Coverage: We use sample coverage to evaluate the coverage rate in the pixel domain. The samples around the center of the camera can be collected easily for most calibration pattern. And for most of the cameras, the distortion happens at the edge of the image. To this end, the region around the edge of the image sample should also be sampled to ensure the camera intrinsic fits the whole camera.

We find a minimum convex hull to cover all of the samples [27] , to show if the sample can describe the edge of the image. The sample coverage r_{cov} is defined as:

$$r_{cov} = \frac{S_{cov}}{S_{image}} \times 100\% \tag{10}$$

where S_{image} is the area of the image and S_{cov} is the area of the convex hull of the whole sample points in pixel coordinates.



(a) Chessboaard calibration. (b) Screen calibration.

Fig. 3: The sample points distribution. Yellow dots show the sample points for calibration. The blue polygon shows the convex hull of the samples. The sample coverage r_{cov} of chessboard calibration (a) and screen calibration (b) is 92.6% and 99.4% respectively.

b) Sample Uniformity: We use the standard deviation σ_ρ of density image $\boldsymbol{\rho}$ to evaluate the uniformity of sample distribution.

We calculate the sample density image $\boldsymbol{\rho}$ which has the same size of pixels, to describe the sample density distribution. The elements of each pixel coordinate is

$$\rho(u,v) = \frac{N_{sw}}{S_{sw}}, \tag{11}$$

where N_{sw} is the number of sample points in the sliding window around pixel $[u\ v]^T$ and S_{sw} is the area of slide window.

The sample uniformity is defined with the standard deviation σ_ρ of the the sample density:

$$\sigma_\rho = \sqrt{\frac{\sum_{\rho(u,v)\in\boldsymbol{\rho}} (\rho(u,v) - \bar{\rho})}{N}} \tag{12}$$

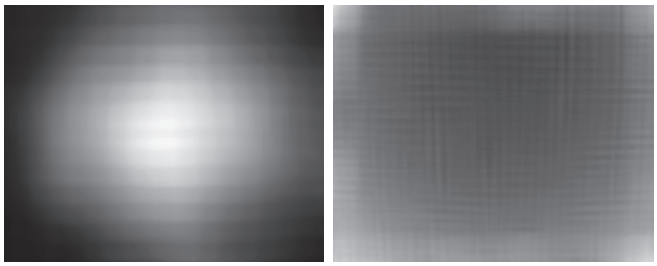
where N is the area of image and $\bar{\rho}$ is the mean value of $\boldsymbol{\rho}$. A smaller σ_ρ means a more uniform of data distribution, which is better for camera calibration.

VI. Camera Calibration

A. Sample from Screen

In the manufacturing industry, the calibration process is limited with time and space in the factory. According to Fig. 3(a), it is challenging to get a uniform samples distribution from a traditional chessboard pattern method. By control the screen, thousands of samples with quality of coverage and uniformity are used for calibration. The distributed image is shown in Fig. 3(b).

The screen is a controllable planer pattern whose marker and texture can be rendered. We can show a particular marker in a particular time timestamp, which evidences the controllability in both time and space. Using a screen as a calibration pattern, we light a few pixels of the screen, form a small marker and detect the light pixel from the image. It is reasonable to generate a uniform data distribution: the sample density is controllable. A screen can provide a maximum 2 million pixel-wise point marker with a 1920×1080 resolution. As for a 12×8 chessboard, the number is 96.



(a) Chessboaard calibration. (b) Screen calibration.

Fig. 4: The illustration of sample density. The intensity shows the sample density of the image: the lighter, the denser. The σ_ρ of chessboard calibration (a) and screen calibration (b) is 5.3×10^{-3} and 8.2×10^{-4} respectively.

Different from the laboratory environments, cameras have similar properties in a particular production line. The hardware debugging ports of the products are reserved for factory by hardware designers.

As shown in Fig. 1, the screen calibration system includes a screen, an embedded trigger device, and the camera. They all connect to a computer. The rotation and the translation between the camera and the screen are mechanical fixed.

As shown in Fig. 5, the computer renders an image with a light blob with a known coordinate. Then shows the image on the screen. Then a trigger signal is sent to the embedded trigger device. The embedded trigger device triggers the camera to capture an image with the shutter trigger. Finally, grabbed images are collected by the computer.

The frequency of the screen is around 60-144 Hz. During the render images reflashing, it is reasonable that the k^{th} captured image is mixed with the k^{th} image and the $k-1^{th}$ image because of the trigger delay, exposure and screen reflashing. To avoid this, we render the images with incremental light blobs: there is one more light blob shown in the k^{th} rendered image than the last image. The sample of raw images are shown in Fig. 6, the left image is $k-1^{th}$ captured image $\mathbf{M}_{k-1}$ and the right image is the k^{th} captured image $\mathbf{M}_k$. The difference image $\mathbf{M}_k^d$ is:

$$\mathbf{M}_k^d = \mathbf{M}_k - \mathbf{M}_{k-1}. \tag{13}$$

Pre-process the difference image $\mathbf{M}_k^d$ by adjust the brightness and contrast, we get image $\mathbf{M}_k^p$:

$$\mathbf{M}_k^p = \alpha \mathbf{M}_k^d + \beta, \tag{14}$$

where α is the image intensity gain and β is the image intensity bias. k^{th} light blob in the image $\mathbf{u}_k$ can be detected in image $\mathbf{M}_k^p$.

Thousands of images are grabbed at the highest frequency, it cost around 10s seconds. In the factory, the hardware reliability test is required for the camera module. The can be set up in the calibration at the same time.

B. Light Point Detect

In the calibration system, the camera is mounted close to the screen ($< 1m$), as shown in Fig. 1. Since the camera is

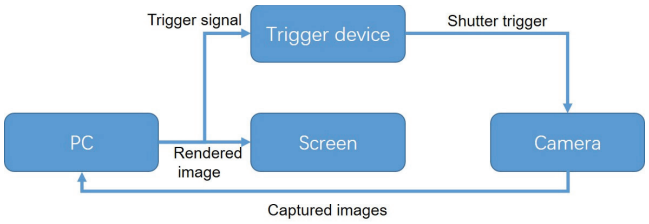
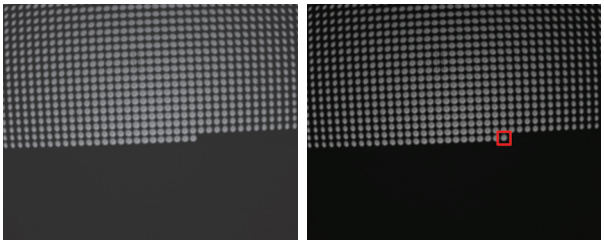


Fig. 5: The pipeline of screen calibration.



(a) Captured at time $k-1$. (b) Captured at time k.

Fig. 6: Two raw images of screen calibration. The red rectangle marked the k^{th} light blob.

focused at a farther distance ($> 1m$), the projection of the light pixel blob will amount to circular disks on the sensor instead of a sharp point, known as circles of confusion (COC) [28], as shown in Fig. 7 shown. The size and shape of the COC mainly depend on the distances from the light blob to the sensor and its focal plane.

Consider the shape of COC is influenced via the distortion, the projected location of a point is estimated by fitting a 2D Gaussian function to the image blob. Fig 8 shows a sample of a fitted Gaussian function. Since the screen pixel object can be assumed to have a small size, as shown in Fig. 7, this method does not suffer from foreshortening effects known to pose challenges for location estimation using spatial fiducials such as circular feature points [29].

The intensity of the COC is described as a Gaussian distribution:

$$\begin{aligned} \mathbf{G}[u|\sigma_u] &= \frac{1}{\sigma_u\sqrt{2\pi}} e^{-u^2/2\sigma_u^2}, \\ \mathbf{G}[v|\sigma_v] &= \frac{1}{\sigma_v\sqrt{2\pi}} e^{-v^2/2\sigma_v^2}, \end{aligned} \tag{15}$$

where the u and v is $\mathbf{u} = [u\ v]^T$, the sub-pixel image coordinate of the light point, σ_u and σ_v is the standard

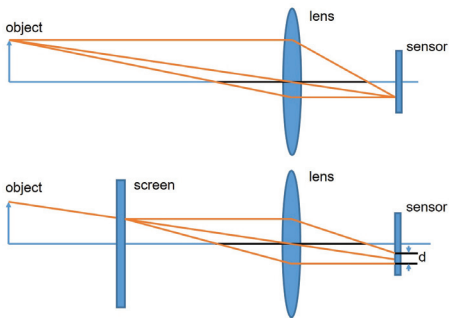
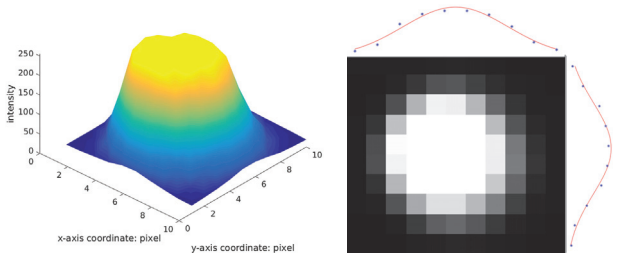


Fig. 7: The illustration of lens projection.



(a) (b)

Fig. 8: The image of the projected point.

deviation of the Gaussian function. For each light point, we estimate the $\mathbf{u}_c$ by fitting a Gaussian function of the blurred intensity profile.

C. Calibration

The back-end of the screen calibration is to fit the model with 3D points and the detected 2D points, similar with the process with chessboard methods.

The calibration is to minimize the sum of the Mahalanobis norm of the re-projection error of all the collected points:

$$\min_{\kappa,\mathbf{R}_i,\mathbf{T}_i} \sum_i \sum_j \left\| \mathbf{u}_{ij} - \pi(\mathbf{R}_i \boldsymbol{\mathcal{P}}_{ij}^w + \mathbf{T}_i) \right\|^2, \tag{16}$$

where κ is the intrinsic in equation (7) and (9). $\boldsymbol{\mathcal{P}}_{ij}^w$ is j^{th} point in the screen (world) frame in i^{th} calibration workstation, and $\mathbf{u}_{ij}$ is the corresponding pixel coordinate. $\mathbf{R}_i$ and $\mathbf{T}_i$ are the relative orientation and translation of the camera respectively, initially estimated mechanically.

Since the calibration pattern should cover all of the sensing regions, for a camera with wide filed-of-view such as fisheye cameras, more screens are required, as shown in Fig. 10(a).

VII. Experiments

In this section, we show the calibration result of some cameras. The result is comparable for the perspective cameras with traditional chessboard methods. And we also show the potential of the fisheye calibration.

A. Implementation Details

The system is combined with a fixed screen, a joint to connect the camera, and an embedded trigger device, as shown in Fig. 1. A normal Arduino Uno[1] embedded device is enough for our system to trigger the shutter of the camera. The PointGrey Chameleon3[2] cameras equipped with the different lens are used as the cameras. With the 1280×1024 resolution and hardware synchronization, the camera can be driven by grabbing images at an up-to-$100Hz$ frequency.

The software system is based on the Robot operation system(ROS)[3]. And the core of the optimization is based on Ceres Solver[4].

[1] https://www.arduino.cc
[2] https://www.ptgrey.com
[3] http://wiki.ros.org
[4] http://ceres-solver.org

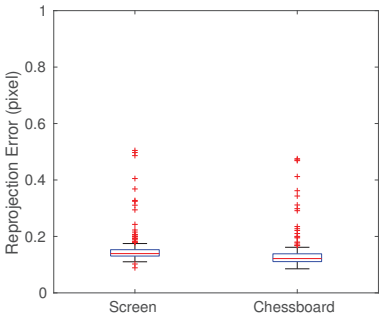


Fig. 9: The reprojection error of the chessboard.

TABLE I: Intrinsic Parameters of Same Camera (Cam3)

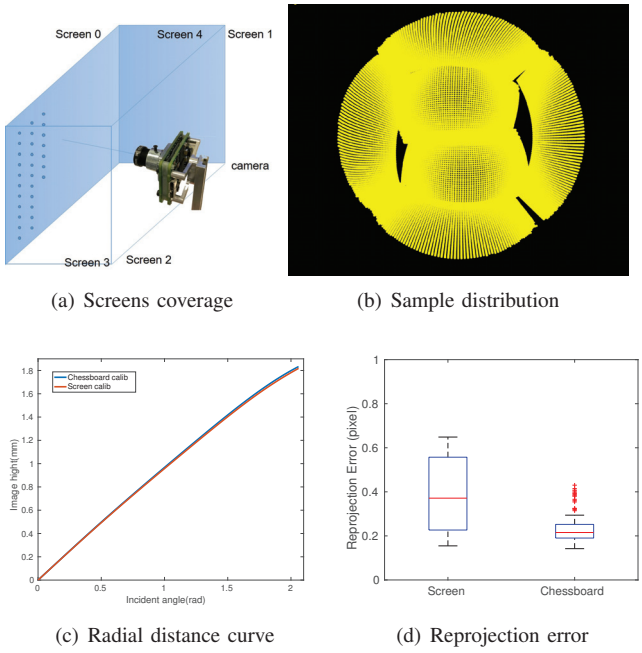
Point Dis.($pixel$)	15	20	25	30	35	Chessboard
Sample Num.	6827	3831	2437	1719	1244	14400
f_x	2561.58	2560.82	2564.62	2561.73	2559.93	2562.89
f_y	2555.30	2554.63	2558.22	2555.46	2553.9	2559.51
c_x	642.90	642.84	640.12	644.01	642.10	649.62
c_y	475.16	474.35	476.99	474.87	473.87	480.20
k_1	-0.38815	-0.38778	-0.45302	-0.38910	-0.38769	-0.40720
k_2	-0.71124	-0.71159	-0.08740	-0.70275	-0.71397	-0.55060
p_1	0.004030	0.004091	0.003980	0.004122	0.004227	0.000110
p_2	0.000827	0.000851	0.001029	0.000833	0.000919	0.000325
Error (pixel)	0.257	0.261	0.382	0.269	0.262	0.236

TABLE II: Intrinsic Parameters of Cameras

Component Component	Cam1 Proposed	Cam1 Chessboard	Cam2 Proposed	Cam2 Chessboard
f_x	886.24	868.08	1680.94	1682.14
f_y	886.47	867.85	1681.25	1683.37
c_x	645.43	649.90	630.42	629.99
c_y	489.81	490.59	516.93	519.32
k_1	-0.1378	-0.1282	-0.1047	-0.1056
k_2	0.1043	0.0868	0.0966	0.0967
p_1	-0.00031	0.00016	0.000048	0.00015
p_2	0.00018	-0.000021	0.00027	0.000091
Error(pixel)	0.1564	0.1392	0.3534	0.4143

TABLE III: Intrinsic Parameters of Fisheye Camera

Component	Chessboard	Proposed
f_x	256.47	259.31
f_y	256.16	258.82
α	0.2025	0.4973
c_x	645.68	645.93
c_y	496.69	497.37
η_2	0.070162	-0.006879
η_3	-0.235970	0.007569
η_4	0.170539	-0.193370
η_5	-0.024252	0.251363
η_6	-0.017415	-0.120197
η_7	0.004437	0.019464
Error(pixel)	0.2361	0.3964



(a) Screens coverage (b) Sample distribution

(c) Radial distance curve (d) Reprojection error

Fig. 10: (a) More screens are required for fisheye cameras. (b) The distribution is awful for the huge distortion and the narrow viewing angle of screens. (c) The comparison of the radial distance. (d) The comparison of the reprojection error.

B. *Screen Calibration Results*

1) Calibrate Perspective Camera: We test the perspective cameras with limit field-of-view. The calibration sample distribution of the perspective camera is shown in Fig. 3(b).

We test the calibration on an 85-degree field-of-view (FOV) wide-angle lens (Cam1), a 50-degree FOV medium focal length lens (Cam2), and a 35-degree FOV telephoto lens (Cam3). The results are shown in in Table I and Table II. Compare with the calibration results by chessboard images. We use chessboard images as the test set to evaluate the accuracy of results. We collect some chessboard images, detect the chessboard points and calculate the reprojection error of the detected points via our screen calibration intrinsic parameters. For the perspective camera, the average reprojection error is 0.1564 pixel by 171 chessboard images, as shown in Fig. 9. The accuracy is comparable with the traditional chessboard methods.

To test the consistency of calibration, we do multiple tests with the parameters changing. We calibrate a camera with different density of the screen pattern, to evaluate a uniform sample will obtain consistent result. The distance between neighbor points is changed from 15 pixels to 35 pixels, the number of the samples decrease from about 6800 to about 1200. The parameter values almost is consistent with the changing of pattern density, as Table I shown. In the factory, the parameters of calibration station need little adjust to suitable for different cameras.

The calibration results of the perspective cameras is enough for computer vision. And the accuracy is comparable with transitional chessboard methods.

2) Calibrate Fisheye Camera: We test the performance of fisheye camera calibration on a 235-degree FOV camera. For ultra-wide FOV cameras, the viewing angle of screens

and the huge distortion influences the detection of COC. The calibration sample distribution is shown in Fig. 10(b), and the intrinsic parameters are shown in Table III. The average reprojection error is 0.4165 pixel by 124 chessboard images, as shown in Fig. 10(d). And Fig. 10(c) compares the radial distance curve of the results, which indicates the distortion. We show the potential of fisheye calibration, the performance would improve with the developing of screens. Using more screens can not only cover the full FOV of the camera but also reducing the viewing angle of samples.

VIII. Conclusion and Discussion

In this paper, we discussed the requirements of calibration in the manufacturing industry. We proposed a quick method to evaluate the calibration sample quality via the coverage and uniformity. We show a camera calibration station prototype with screens is able to provide high-quality samples with uniform distribution and coverage of the whole image. We presented a screen-based solution for automated camera intrinsic calibration on production lines. The accuracy is comparable with the transitional method for perspective cameras. And we mention the potential for fisheye cameras.

Our proposed calibration needs little manual interaction operation, a limited amount of operation space and cost, unlike a variety of existing methods. It is able to handle a massive number of cameras in a limited time and space with a high consistency quality and easy to deploy in industrial environments. We believe a flexible camera calibration system will be widely used in flexible manufacturing in next decades.

References

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[3] Y. Oyamada, P. Fallavollita, and N. Navab, "Single camera calibration using partially visible calibration objects based on random dots marker tracking algorithm," in *The IEEE and ACM International Symposium on Mixed and Augmented Reality, Workshop on Tracking Methods and Applications (TMA), Atlanta, USA*, 2012.

[4] Q. Chen, H. Wu, and T. Wada, "Camera calibration with two arbitrary coplanar circles," in *Proc. of the Euro. Conf. on Computer Vision*. Springer, 2004, pp. 521–532.

[5] F. Bergamasco, L. Cosmo, A. Albarelli, and A. Torsello, "Camera calibration from coplanar circles," in *Proc. of the IEEE Intl. Conf. on Pattern Recognition*. IEEE, 2014, pp. 2137–2142.

[6] X. Meng and Z. Hu, "A new easy camera calibration technique based on circular points," *Pattern Recognition*, vol. 36, no. 5, pp. 1155–1164, 2003.

[7] P. Gurdjos, A. Crouzil, and R. Payrissat, "Another way of looking at plane-based calibration: the centre circle constraint," in *European conference on computer vision*. Springer, 2002, pp. 252–266.

[8] J.-Y. Bouguet, "Camera calibration toolbox for matlab," *http://www. vision. caltech. edu/bouguetj/calib_doc/index. html*, 2004.

[9] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[10] M. Agrawal and L. S. Davis, "Camera calibration using spheres: A semi-definite programming approach," in *null*. IEEE, 2003, p. 782.

[11] K.-Y. K. Wong, G. Zhang, and Z. Chen, "A stratified approach for camera calibration using spheres," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 305–316, 2011.

[12] M. Agrawal and L. Davis, "Complete camera calibration using spheres: Dual space approach," *IEEE*, vol. 206, pp. 782–789, 2003.

[13] H. Zhang, K. W. Kwan-yee, and G. Zhang, "Camera calibration from images of spheres," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 499–502, 2007.

[14] H. Zhuang, K. Wang, and Z. S. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 5, pp. 649–660, 1995.

[15] J. Miseikis, K. Glette, O. J. Elle, and J. Torresen, "Automatic calibration of a robot manipulator and multi 3d camera system," in *System Integration (SII), 2016 IEEE/SICE International Symposium on*. IEEE, 2016, pp. 735–741.

[16] C. Reinbacher, G. Munda, and T. Pock, "Real-time panoramic tracking for event cameras," *arXiv preprint arXiv:1703.05161*, 2017.

[17] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[18] J. Mallon and P. F. Whelan, "Which pattern? biasing aspects of planar calibration patterns and detection methods," *Pattern recognition letters*, vol. 28, no. 8, pp. 921–930, 2007.

[19] I. Schillebeeckx and R. Pless, "Single image camera calibration with lenticular arrays for augmented reality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3290–3298.

[20] L. Tan, Y. Wang, H. Yu, and J. Zhu, "Automatic camera calibration using active displays of a virtual pattern," *Sensors*, vol. 17, no. 4, p. 685, 2017.

[21] X. Hu, G. Wang, J. Wang, P. Sun, J. Fan, F. Chen, and Y. Xie, "A robust and accurate calibration method for out-of-focus camera," *Electronic Imaging*, vol. 2018, no. 2, pp. 1–5, 2018.

[22] H. Ha, Y. Bok, K. Joo, J. Jung, and I. So Kweon, "Accurate camera calibration robust to defocus using a smartphone," in *Proceedings of the IEEE International conference on computer vision*, 2015, pp. 828–836.

[23] M. A. Tehrani, T. Beeler, and A. Grundhöfer, "A practical method for fully automatic intrinsic camera calibration using directionally encoded light," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 125–133.

[24] G. Bradski and A. Kaehler, "Opencv," *Dr. Dobbs journal of software tools*, vol. 3, 2000.

[25] W. Gao and S. Shen, "Dual-fisheye omnidirectional stereo," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 6715–6722.

[26] K. Kanatani, "Calibration of ultrawide fisheye lens cameras by eigenvalue minimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 813–822, 2013.

[27] D.-T. Lee, "On finding the convex hull of a simple polygon," *International journal of computer & information sciences*, vol. 12, no. 2, pp. 87–98, 1983.

[28] R. Jacobson, S. Ray, G. G. Attridge, and N. Axford, *Manual of Photography*. Taylor & Francis, 2000.

[29] K. Kanatani, Y. Sugaya, and Y. Kanazawa, "Ellipse fitting for computer vision: implementation and applications," *Synthesis Lectures on Computer Vision*, vol. 6, no. 1, pp. 1–141, 2016.