

Gaussian Process for Lens Distortion Modeling

Pradeep Ranganathan and Edwin Olson

Abstract—When calibrating a camera, the radial component of lens distortion is the dominant source of image distortion. To model this lens distortion, camera models incorporate a radial distortion model that conforms to a certain parametric form. In practice however, multiple parametric forms can be used to model distortion for a given lens. Ideally, one would choose the best suited parametric form using a model selection procedure.

In this work, we propose the use of Gaussian Process regression to model lens distortion. With the use of a squared exponential covariance function, a Gaussian Process (GP) can describe the space of smooth distortion functions; kernel hyper-parameter selection in this space then analogous to performing explicit model selection between possible parametric models.

Our evaluation shows that this Gaussian Process formulation of lens distortion performs on par with parametric distortion models.

I. INTRODUCTION

A camera calibration procedure obtains a camera model that can be used to extract metric information such as object size or depth from 2D images. This procedure optimizes a parametric camera model to best explain a set of observed correspondences between world and image points.

For applications in computer vision and photogrammetry, it is common to assume a projective pinhole camera model. This simple pinhole model, however, does not account for the distortion caused by the camera lens. Therefore, camera models are augmented with a radial lens distortion model to compensate for lens effects. The calibration procedure for the augmented camera is then modified to estimate the parameters of the lens distortion, along with the other camera parameters.

Usually, the lens distortion is assumed to be well modeled by a parametric family of functions. Examples of parametric distortion models include polynomial, field-of-view, rational function and division models [1]. The particular choice of parametric family used is based on the type of lens being modeled. In reality, lenses may not conform to any of the proposed parametric distortion models. Manufacturing errors and tolerances can cause additional deviations from the intended distortion. In this situation, a rigorous approach for choosing the best parametric model is to perform model selection on multiple calibrated lens distortion models [2].

Also, traditional optimization based calibration approaches are designed to produce maximum likelihood estimates of the model parameters. However, these point estimates fail to capture the uncertainty in the model parameters, which arise because of inherent noise in input data: correspondences

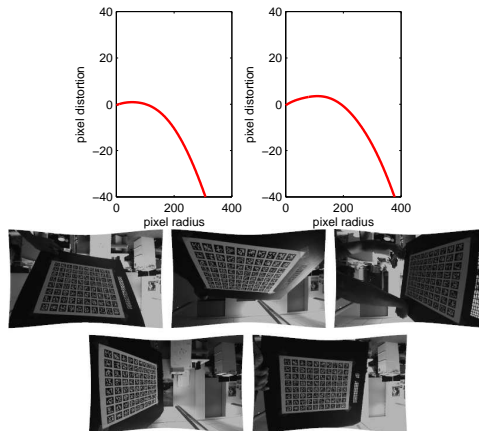


Fig. 1. A Gaussian process lens distortion model. In our work, we propose a technique for using a Gaussian process to learn a lens distortion model. Using the learn lens distortion model, distorted images from a camera can then be undistorted as shown.

between world and image points can be affected by pixel quantization, image feature localization accuracy and tolerances when printing the calibration target.

Prior research such as [3], [4], [5] investigate the propagation of input uncertainty through the calibrated camera model. However they only model the uncertainty of the pinhole camera parameters: much like the pinhole camera parameters, there is uncertainty in an estimate of the distortion function too.

In order to model this distortion function uncertainty, one would expect to capture a *distribution* over possible distortion functions. However, using a parametric model of the lens distortion makes this challenging since simple distributions over the distortion parameters can lead to complex and even unlikely distributions over distortion functions.

Using a GP to model lens distortion provides an elegant alternative to the problem of model selection. With a squared exponential kernel, a GP can model the space of smooth non-linear functions. Now, kernel hyper-parameter selection in this space of smooth functions is analogous to the process of explicit model comparison/selection between multiple parametric models.

Also, a simple GP prior over a function results in a simple GP posterior over that function. In other words a GP posterior can capture a distribution over possible smooth distortion functions. Thus a GP provides an elegant way of modeling the uncertainty in the distortion function estimates.

Traditionally, the camera calibration problem is formulated as a non-linear least-squares model fitting problem. This for-

The authors are with Department of Computer Science and Engineering, University of Michigan, Ann Arbor, USA {rpradeep,ebolson}@umich.edu

mulation can alternatively be viewed as maximum likelihood inference in a factor graph with Gaussian factor potentials (see [6]). Incorporating a parametric distortion model is achieved in a straight-forward manner (see section IV). Since GP models are a non-parametric regression technique, they have no explicit parameters and it is not immediately obvious as to how one can incorporate a GP into a framework that optimizes parametric models.

In this work, we present a method for incorporating a GP model into this factor graph inference framework. The GP distortion model is then learned in a joint optimization with other parameters. This method adds great flexibility to the factor graph optimization formulation, while preserving much of its computational efficiency.

Finally, we propose a rigorous evaluation strategy that evaluates the performance of camera models on a set of test images. The use of separate training and testing datasets might seem obvious to readers with a machine learning background. Nonetheless, this has not been a standard practice in camera modeling tasks¹.

In summary, the important contributions of our work are:

- We present a camera model that uses a GP to model lens distortion. We also show how a GP model can be incorporated into a factor graph inference framework optimized for Gaussian factor potentials.
- We propose a rigorous evaluation strategy that evaluates the performance of camera models on a set of test images.
- We show that our GP distortion model achieves accuracy comparable to the best parametric model.
- The proposed GP lens distortion model provides a basis for modeling the uncertainty in the estimate of the lens distortion.

In the following section we review prior work in this area. In the next section we give a brief summary of the background necessary to understand our work. Following that, in section IV, we describe how we model lens distortion using a GP and optimize the parameters of the GP. In section V we evaluate our method. Finally, we conclude this paper with a discussion on future work.

II. PREVIOUS WORK

Camera calibration techniques generally fall into two broad categories: *photogrammetric calibration* and *self-calibration*. Photogrammetric calibration uses a calibration target with a precisely known 3-D structure [8]. Alternatively, a planar target undergoing controlled motion can be used instead of a 3-D calibration target [9].

In contrast, self-calibration methods use multiple views of the same scene and exploit the rigidity of the scene to calibrate the camera parameters [10]. Photogrammetric and self-calibration methods use the same lens distortion models; only the calibration procedures are different. In this work however, we choose to implement photogrammetric

calibration, since it is the more mature technique. A classic reference for the reader interested in the history of camera calibration methods is the work by Clarke and Fryer [11].

Zhang [12] was the first to present a technique that used multiple views of a single planar calibration target to calibrate a camera. This technique is very popular because it requires only a planar calibration target that is easily manufactured. Because of its simplicity, many open-source calibration toolkits [13] are based on this technique.

Calibrating a lens distortion model in a photogrammetric or self-calibration setting falls under the technique of *metric lens calibration*. Alternatively, one can exploit the fact that straight lines in the scene must project to straight lines, in order to correct lens distortion in a *non-metric* fashion [14], [15]. In our work, we use metric lens calibration, since it is the natural choice for photogrammetric calibration methods.

Other non-parametric models have been proposed in the literature. Hartley and Kang [16] use a locally averaged estimate of observed distortion to build a non-parametric distortion model. Ricolfe-Viala and Sanchez-Salmeron [15] explore the use of model-free distortion estimation in a non-metric lens calibration setting. Our work differs from these approaches in that we use a GP as the non-parametric distortion model. This gives our method the advantage that the smoothness of the resulting distortion function is well determined. It also gives our model the ability to capture the uncertainty in the distortion function estimate.

III. BACKGROUND

A. Gaussian Process Regression

The use of Gaussian processes for regression is an extensive topic; we present a practical definition of the technique here and refer the reader to [17] for a thorough mathematical treatment.

GP regression can be understood as predicting new outputs using a locally weighted sum of the observed targets, where the local weighting is specified indirectly through a covariance function [18].

For GP regression on a set of observations t at a set of input locations x , we assume that the data was obtained from a GP with covariance function $k(x_n, x_m)$. Let β denote the precision (inverse variance) of input noise of the observed target values t . We then define C as the covariance matrix with elements $C(x_n, x_m) = k(x_n, x_m)$. When we require a prediction for a new input x_{N+1} , we first construct the covariance matrix C_{N+1} and partition it as follows:

$$C_{N+1} = \begin{bmatrix} C_N & k \\ k^T & c \end{bmatrix}$$

Then the mean and variance of the predicted value at a new location x_{N+1} are given by

$$m(x_{N+1}) = k^T C_N^{-1} t \quad (1)$$

$$\sigma^2(x_{N+1}) = c - k^T C_N^{-1} k \quad (2)$$

¹However, literature in the related field of optics and optical modeling report testing errors when evaluating models (see [7]).

A popular choice for the covariance function is the squared exponential (SE) kernel

$$k(x_m, x_n) = \theta_1^2 \exp \left\{ -0.5 \frac{(x_m - x_n)^2}{\theta_0^2} \right\} + \beta^{-1} \delta_{ij}$$

This covariance function captures the correlation between the outputs at different locations of the GP. Informally, one can think of the covariance function as controlling the smoothness of the resulting regression curve. The SE covariance function estimates infinitely differentiable curves and hence produces regression curves that are “very smooth”. It is reasonable to assume that distortion functions are smooth, infinitely differentiable functions and hence we use the SE covariance function in our formulation.

For effective regression, one has to determine appropriate kernel hyper-parameters $\theta_0, \theta_1, \beta$. A standard technique to find optimal values of the hyper-parameters, is to optimize the marginal likelihood of observed data given the hyper-parameters. In this work however, we use cross-validation on data folds to find appropriate values of the hyper-parameters, as explained in Section IV.

B. Camera Calibration

In this section, we present a basic overview of Zhang’s calibration method [12] and show how camera calibration can be formulated as a non-linear least-squares problem.

1) *Homography estimation:* For purposes of calibration, a camera is modeled as a projective pinhole camera. In this model, an object viewed through a camera undergoes a rigid-body transformation followed by a perspective projection as defined by the following equation:

$$\underbrace{\begin{bmatrix} u \\ v \\ w \end{bmatrix}}_{\mathbf{y} \text{ (image)}} = \underbrace{\begin{bmatrix} f_x & 0 & c_u & 0 \\ 0 & f_y & c_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Camera matrix}} \underbrace{\begin{bmatrix} r_{11} & r_{21} & r_{31} & t_x \\ r_{12} & r_{22} & r_{32} & t_y \\ r_{13} & r_{23} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Extrinsics}} \underbrace{\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}}_{\mathbf{x} \text{ (world)}}$$

The point (c_u, c_v) , called the *principal point* or *optical center* is the center of the coordinate space defined on the image plane. The homogeneous 3D point $\mathbf{x} = [x \ y \ z \ 1]^T$ is a point in the world coordinate frame. It is projected into a homogeneous 2D image coordinate space by first transforming it by a rigid-body transform $\mathbf{E} = [R \ t]$ and then projecting it into image coordinate space via the camera matrix \mathbf{K} containing the parameters f_u, f_v, c_u, c_v . The parameters f_u, f_v represent the focal length of the pinhole model in the u and v directions. This projection gives us the homogeneous 2D point $\mathbf{y} = [\frac{u}{w} \ \frac{v}{w} \ 1]^T$. The matrix \mathbf{K} contains the camera intrinsic parameters and \mathbf{E} is the matrix of camera extrinsic parameters.

The combined transformation $\mathbf{H} = \mathbf{K} \times \mathbf{E}$ defines a linear map from world coordinates to image coordinates, called a *homography*. The camera calibration problem is then mathematically equivalent to estimating a homography from a set of world-image correspondences. This estimation is done using the *Direct Linear Transform* (DLT) method [19].

In order to obtain a closed form solution, the DLT method estimates an \mathbf{H} that aligns the directions of the world and image point vectors. Mathematically, this translates to minimizing the following sum of errors:

$$e = \sum_i \mathbf{y}_i \times \mathbf{H}\mathbf{x}_i \quad (3)$$

There are two things to note about the DLT. First, one would expect to minimize the geometric reprojection error

$$e^* = \sum_i (\mathbf{y}_i - \mathbf{H}\mathbf{x}_i)^2 \quad (4)$$

Instead, it optimizes an algebraic error (3), that is only related to this geometric error. Second, given an estimate of the camera matrix \mathbf{K} , it is possible to recover the extrinsics \mathbf{E} from the estimated homography \mathbf{H} . Details of this procedure can be found in [12].

C. Parametric radial distortion

The camera model described previously ignores the distortion caused by the lens while estimating the homography. Also, it minimizes an algebraic error instead of the geometric re-projection error. Hence, the camera model is refined by performing iterative non-linear minimization on an error function that includes a lens distortion model.

The conventional parametric approach is to augment the camera model with a radial distortion function $d(\cdot)$ to take into account the lens distortion. First, we define r as $\sqrt{\|\mathbf{H}\mathbf{x}_i - \mathbf{c}_{uv}\|}$, where $\mathbf{c}_{uv} = [c_u \ c_v]^T$, and then define d as a function of this radius r . A parametric form is assumed for the function d . The mathematical form of the augmented camera model is now

$$\mathbf{y}_i = f(\mathbf{H}\mathbf{x}_i)$$

The iterative non-linear optimization procedure then seeks to fit the model parameters (intrinsics, extrinsics and distortion) by minimizing the following error function

$$e = \sum_i (\mathbf{y}_i - f(\mathbf{H}\mathbf{x}_i))^2 \quad (5)$$

This is a non-linear least squares estimation problem. The optimization procedure works by linearizing the error function at the current estimate of the variables, and then solving the resulting linear least squares problem to obtain an updated estimate. This procedure is iterated until convergence. Estimates from the DLT algorithm are used to initialize this optimization procedure.

IV. METHOD AND IMPLEMENTATION

Zhang’s camera calibration method [12] operates by extracting world-image point correspondences from n different images of the calibration target. It then sets up a non-linear optimization to infer the values of the camera intrinsics, lens distortion parameters, and n different calibration extrinsics that best explain the observed world-image correspondences.

It is interesting to note the sparse variable dependence structure of this problem: each image point \mathbf{y}_i is dependent only on the corresponding input world point \mathbf{x}_i , the extrinsics

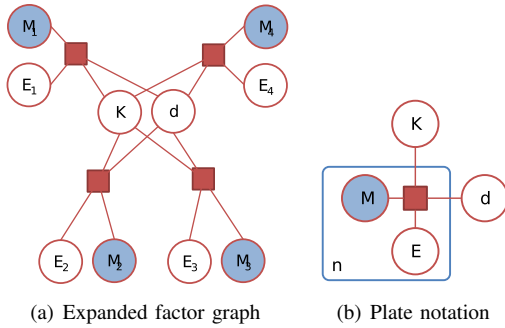


Fig. 2. Factor graph interpretation of the camera calibration problem. Four sets of world-image correspondences $M_1 \dots M_4$ have been observed from four different views $E_1 \dots E_4$ of the calibration target. A factor node connects each E_j, M_j pair with the camera intrinsics \mathbf{K} and distortion parameters \mathbf{d} , because each set of observed correspondences M_j are affected only by the nodes it is connected to (refer text for details). On the right is the plate notation representation of this factor graph.

of that particular calibration target pose E_j , camera intrinsics \mathbf{K} and distortion function parameters \mathbf{d} . This structure is captured graphically when the optimization problem is expressed in terms of inference in a factor graph, as shown in Fig. 2. This relation between non-linear least-squares optimization and inference in a factor graph has been previously studied in the SLAM community (see [6]).

Consider the factor graph in Fig. 2. A factor node connects nodes $\mathbf{K}, \mathbf{D}, E_j$ and the correspondences for that target pose M_j . The factor potential for this factor node is the non-linear geometric reprojection error (refer eq. 5):

$$q(\mathbf{K}, E_j, \mathbf{d}) = \sum_m \{y_{im} - f(\mathbf{K}E_j \mathbf{x}_{im})\}^2, \quad (6)$$

where the sum is over the m correspondences obtained for calibration target pose j . Inference on the factor graph then tries to obtain values for \mathbf{K}, E_j and \mathbf{d} that minimize each of the factor potentials. This factor graph can be represented compactly using plate notation as shown in Fig. 2(b).

The node \mathbf{d} in the factor graph contains the parameters of the distortion model. An important contribution of our work is to show how a GP model can be incorporated into this factor graph based inference framework, by replacing \mathbf{d} with a node that can model a GP.

A. Gaussian process factors in a factor graph framework

To explain incorporation of GP models in a factor graph inference framework, we begin from a basic property of a GP: by definition, a finite set of samples \mathbf{g} from a GP with zero mean prior and covariance function $k(x_m, x_n)$ has a multi-variate Gaussian distribution: $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, where the entries of the covariance matrix \mathbf{C} are given by the covariance function $k(x_m, x_n)$.

A GP is an infinite-dimensional entity and we cannot model it directly. However, a finite subset of its samples has a Gaussian distribution with a specified covariance structure. Thus, we can indirectly model a GP using a finite subset of its samples. In a factor graph setting, we express this by creating a node containing GP samples. The factor node connected

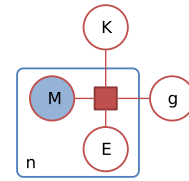


Fig. 3. Camera calibration factor graph with a GP lens distortion model. This factor graph has the same structure as the one in Fig. 2. However we replace the parametric distortion node \mathbf{d} with a node \mathbf{g} that contains distortion function samples. The factor node connected to \mathbf{g} then uses a distortion function estimated from these function samples.

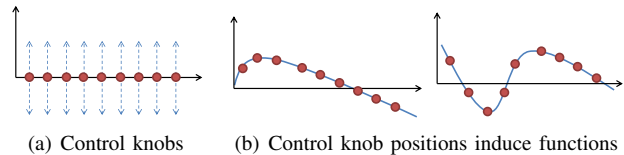


Fig. 4. Control knob analogy for GP models. The set of function samples in a GP factor node act as control knobs that induce a function based on their position. Inference on the GP node can then be visualized as positioning these knobs in order to induce the required function. This is analogous to the use of control points for manipulating Bezier and spline curves in graphic modeling software.

to these GP samples then uses the prediction equations (1) and (2) to interpolate the values of the distortion function for other pixel locations.

In other words, we try to infer the distortion function values at a set of pre-specified pixel radii. Intuitively, these distortion function values *induce* a mean function that is used to predict the values of distortion for new input pixel radii. One way of visualizing this technique is to see these distortion function values as “control-knobs” that control a smooth curve (see Fig. 4). Thus, in the case of a GP distortion model, the factor graph inference procedure infers a function indirectly, by inferring the position of these control-knobs (distortion function values).

The implementation of the factor node in the case of the factor graph in Fig. 3, uses a factor potential that is similar to (6). The only difference is the way in which the function f is evaluated: the mean prediction function (1), operating on the targets \mathbf{g} is used to evaluate $f(\cdot)$.

B. Gaussian process hyper-parameter optimization

In a fully Bayesian setting, a rigorous way of optimizing hyper-parameters is to optimize the marginal likelihood of the data given the hyper-parameters [17]. In our case, however, we only perform maximum-likelihood inference of the model parameters. Hence we are unable to apply Bayesian estimation techniques directly. Instead, we use cross-validation on a held-out validation set to find good GP hyper-parameter settings.

C. Implementation

We use the April Robotics Toolkit [20] to implement our camera calibration software. This toolkit provides a framework that allows us to express the calibration problem in terms of its underlying factor graph. We then solve the

resulting optimization problem using a Levenberg-Marquardt solver.

When integrating a GP model into the factor graph, one has to fix the number of function samples inferred within the GP node. A large number of samples makes the optimization process computationally expensive. However, a relatively small number of samples should suffice since the distortion function being modeled is smooth. In our implementation, we perform inference on 25 equally spaced function values.

For the calibration procedure, we used a training set of 14 images. An average of 40 world image correspondences were extracted from each image. In order to find hyper-parameter settings, we chose to further divide the training set into a development set consisting of 11 images and a validation set of 3 images. By random shuffling, 30 different development/validation partitions were created from the same training set.

We perform a grid search over the space of hyper-parameter settings and evaluate the validation error of each hyper-parameter setting on the 30 different training set partitions. We then choose the hyper-parameter settings that gave the lowest validation error as the best estimate of the hyper-parameters.

We evaluate prediction error on a testing/validation image by undistorting a test image of the calibration target using the lens distortion model that was estimated. World-image point correspondences are then obtained from this undistorted image. Then, an initial estimate of the calibration target extrinsics is obtained using the DLT method. This estimate is then refined using a non-linear optimization of the reprojection error, with the camera model and lens distortion held constant. The reprojection error after the non-linear optimization has converged is reported as the prediction error.

V. EXPERIMENTS

We evaluated the performance of our method on two wide field-of-view camera lenses: a Tamron lens with 2.2 mm focal length and another Tamron lens with 2.8 mm focal length. The 2.2 mm focal length lens produced more distortion than the 2.8 mm focal length lens. As mentioned previously, we report test errors as a rigorous assessment of model predictive performance.

For both the lenses, we use an independent calibration sequence of 10 images as the test set. We present the results of this evaluation in Fig. 5. From our plots, we conclude that the GP distortion model is capable of capturing a suitable distortion function. Also, the performance of the GP distortion model is comparable to the performance of the best polynomial lens-distortion models.

The lens distortion model estimated for the lenses are shown in Fig. 6. The estimated models suggest a barrel type distortion for both the lenses. Visual inspection of the acquired camera images confirms the presence of predominant barrel distortion, and thus confirms the validity of the estimated model.

By convention, the number of training images used for calibration is six. This is due to a result presented in Zhang’s

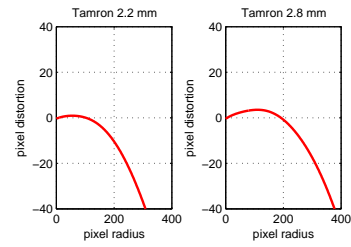


Fig. 6. Estimated distortion functions. This plots shows a pixel radius vs pixel distortion plot of the estimated GP distortion model. The estimated model has a large negative direction, suggesting a predominant barrel distortion.

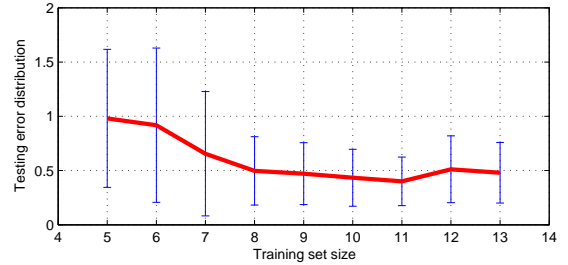


Fig. 7. Variation in test error as a function of training set size. This plot shows that the that variation in test error reduces with increasing training set size. However, the test error stabilizes around a training set size of 12. Hence, for our experiments, we choose a training (development when using validation) set size of 11 images.

original paper [12], which shows that the training error does not improve much with the use of more than 6 target images. We perform a similar study for our method and present our results in Fig. 7. Once again, we report improvement in testing error instead of training error since it is a rigorous way of evaluating model predictive performance. The results show that the improvement in testing error variance for training sets with more than 10 images is minimal. Based on this experiment, we choose a development set size of 11 images for this work.

VI. CONCLUSION AND FUTURE WORK

In this work, we proposed the use of a Gaussian Process model for lens distortion. A GP model of lens distortion provides an alternative approach for performing explicit model selection amongst multiple parametric models. Because a GP can represent a distribution over functions, it is also suitable for capturing the uncertainty in an estimate of the lens distortion.

We implemented our calibration procedure, by expressing it as a factor graph inference problem and then using a factor graph inference framework to perform maximum likelihood inference. A core contribution of our work is a technique for incorporating Gaussian Process nodes into this factor graph inference framework.

Finally, we presented results confirming the suitability of using this Gaussian process lens distortion model for modeling the distortion of two real world lenses. Our results also confirmed that the GP distortion model performed on par with polynomial lens distortion models.

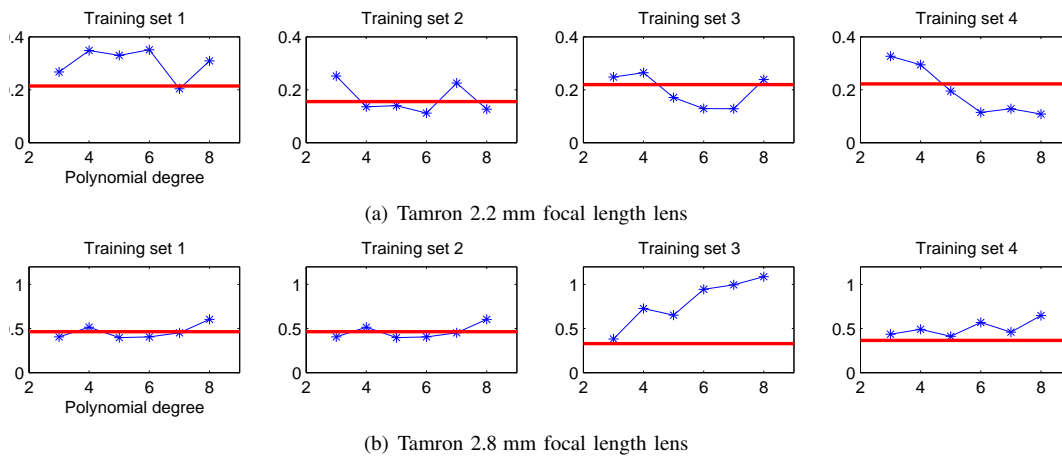


Fig. 5. Performance comparison of distortion models for two Tamron lenses. Here, we plot the test (pixel reprojection) errors obtained when using the GP model (red bar) and polynomial models of orders 3,4,...8 (blue segments). We observe that the GP model performs on par with the best polynomial distortion models.

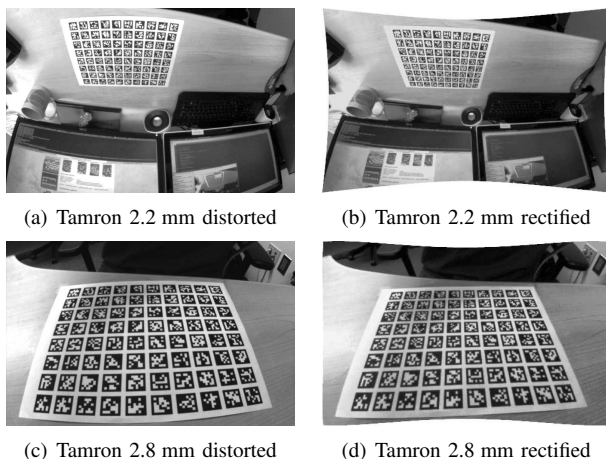


Fig. 8. Rectified images. The estimated distorted models are used to rectify the image. The lenses have a predominant barrel type distortion (straight-lines tend to bend inwards with increasing distance from the center). The 2.2 mm focal length lens has a wider field-of-view and hence produces more distortion than the 2.8 mm lens.

In our current implementation, the GP model optimization is an order of magnitude slower than the polynomial model optimization, because of the multiple validation runs used to determine the optimal hyper-parameter settings. As future work, we intend to pursue more efficient gradient based hyper-parameter optimization strategies.

REFERENCES

- [1] D. Claus and A. Fitzgibbon, "A rational function lens distortion model for general cameras," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, june 2005, pp. 213 – 219 vol. 1.
- [2] M. El-Melegy and A. Farag, "Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 554–559.
- [3] R. Haralick, "Propagating covariance in computer vision," *Series in Machine Perception and Artificial Intelligence*, vol. 25, pp. 171–182, 1996.
- [4] R. Sundaeswara and P. R. Schrater, "Bayesian modelling of camera calibration and reconstruction," in *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 394–401.
- [5] G. Di Leo and A. Paolillo, "Uncertainty evaluation of camera model parameters," in *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*, may 2011, pp. 1–6.
- [6] F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.
- [7] C. Ricolfe-Viala and A.-J. Sanchez-Salmeron, "Using the camera pin-hole model restrictions to calibrate the lens distortion model," *Optics and Laser Technology*, vol. 43, no. 6, pp. 996 – 1005, 2011.
- [8] J. Heikkila, "Geometric camera calibration using circular control points," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 10, pp. 1066 – 1077, oct 2000.
- [9] J.-M. Frahm and R. Koch, "Camera calibration with known rotation," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, oct. 2003, pp. 1418 –1425 vol.2.
- [10] S. J. Maybank and O. D. Faugeras, "A theory of self-calibration of a moving camera," *International Journal of Computer Vision*, vol. 8, pp. 123–151, 1992.
- [11] T. A. Clarke and J. G. Fryer, "The development of camera calibration methods and models," *The Photogrammetric Record*, vol. 16, no. 91, pp. 51–66, 1998.
- [12] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1330–1334, November 2000.
- [13] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [14] F. Devernay and O. Faugeras, "Straight lines have to be straight," in *In SPIE, volume 2567*, 2001.
- [15] C. Ricolfe-Viala and A.-J. Snchez-Salmern, "Correcting non-linear lens distortion in cameras without using a model," *Optics and Laser Technology*, vol. 42, no. 4, pp. 628 – 639, 2010.
- [16] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1309 –1321, aug. 2007.
- [17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, Nov. 2005.
- [18] P. Sollich and C. K. I. Williams, "Understanding gaussian process regression using the equivalent kernel," in *Deterministic and Statistical Methods in Machine Learning*, 2004, pp. 211–228.
- [19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press.
- [20] E. Olson, "The APRIL robotics toolkit," 2010.