Learning Depth From Stereo

Fabian H. Sınz[1], Joaquin Quiñonero Candela[2], Gökhan H. Bakır[1],
Carl E. Rasmussen[1], and Matthıas O. Franz[1]

[1] Max Planck Institute for Biological Cybernetics
Spemannstraße 38, 72076 Tübingen
{fabee;jqc;gb;carl;mof}@tuebingen.mpg.de
[2] Informatics and Mathematical Modelling, Technical University of Denmark,
Richard Petersens Plads, B321, 2800 Kongens Lyngby, Denmark
jqc@imm.dtu.dk

Abstract. We compare two approaches to the problem of estimating the depth of a point in space from observing its image position in two different cameras: 1. The classical photogrammetric approach explicitly models the two cameras and estimates their intrinsic and extrinsic parameters using a tedious calibration procedure; 2. A generic machine learning approach where the mapping from image to spatial coordinates is directly approximated by a *Gaussian Process* regression. Our results show that the generic learning approach, in addition to simplifying the procedure of calibration, can lead to higher depth accuracies than classical calibration although no specific domain knowledge is used.

1 Introduction

Inferring the three-dimensional structure of a scene from a pair of stereo images is one of the principal problems in computer vision. The position $\mathbf{X} = (X, Y, Z)$ of a point in space is related to its image at $\mathbf{x} = (x, y)$ by the equations of perspective projection

$$x = x_0 - s_{xy} c \cdot \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Xi_x(\mathbf{x}) \quad (1)$$

$$y = y_0 - c \cdot \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Xi_y(\mathbf{x}) \quad (2)$$

where $\mathbf{x}_0 = (x_0, y_0)$ denotes the image coordinates of the principal point of the camera, c the focal length, $\mathbf{X}_0 = (X_0, Y_0, Z_0)$ the 3D-position of the camera's optical center with respect to the reference frame, and r_{ij} the coefficients of a 3×3 rotation matrix R describing the orientation of the camera. The factor s_{xy} accounts for the difference in pixel width and height of the images, the 2-D-vector field $\Xi(\mathbf{x})$ for the lens distortions.

The classical approach to stereo vision requires a *calibration procedure* before the projection equations can be inverted to obtain spatial position, i.e., estimating the *extrinsic* ($\mathbf{X}_0$ and R) and *intrinsic* ($\mathbf{x}_0$, c, s_{xy} and Ξ) parameters of each

camera from a set of points with known spatial position and their corresponding image positions. This is normally done by repeatedly linearizing the projection equations and applying a standard least square estimator to obtain an iteratively refined estimate of the camera parameters [1]. This approach neglects the nonlinear nature of the problem, which causes that its convergence critically depends on the choice of the initial values for the parameters. Moreover, the right choice of the initial values and the proper setup of the models can be a tedious procedure.

The presence of observations and desired target values on the other hand, makes depth estimation suitable for the application of nonlinear supervised learning algorithms such as *Gaussian Process Regression*. This algorithm does not require any specific domain knowledge and provides a direct solution to nonlinear estimation problems. Here, we investigate whether such a machine learning approach can reach a comparable performance to classical camera calibration. This can lead to a considerable simplification in practical depth estimation problems as off-the-shelf algorithms can be used without specific adaptations to the setup of the stereo problem at hand.

2 Classical Camera Calibration

As described above, the image coordinates of a point are related to the cameras parameters and its spatial position by a nonlinear function $\mathbf{F}$ (see Eqs. 1 and 2)

$$\mathbf{x} = \mathbf{F}(\mathbf{x}_0, c, s_{xy}, R, \mathbf{X}_0, \Xi, \mathbf{X}) \tag{3}$$

The estimation of parameters is done by a procedure called *bundle adjustment* which consists of iteratively linearizing the camera model in parameter space and estimating an improvement for the parameter from the error on a set of m known pairs of image coordinates $\mathbf{x}_i = (x_i, y_i)$ and spatial coordinates $\mathbf{X}_i = (X_i, Y_i, Z_i)$. These can be obtained from an object with a distinct number of points whose coordinates with respect to some reference frame are known with high precision such as, for instance, a calibration rig.

Before this can be done, we need to choose a low-dimensional parameterization of the lens distortion field Ξ because otherwise the equation system 3 for the points $1 \dots m$ would be underdetermined. Here, we model the x- and y-component of Ξ as a weighted sum over products of one-dimensional Chebychev polynomials T_i in x and y, where i indicates the degree of the polynomial

$$\Xi_x(\mathbf{x}) = \sum_{i,j=0}^{t} a_{ij} T_i(s_x x) T_j(s_y y),\ \Xi_y(\mathbf{x}) = \sum_{i,j=0}^{t} b_{ij} T_i(s_x x) T_j(s_y y), \tag{4}$$

The factors s_x, s_y scale the image coordinates to the Chebychev polynomials' domain $[-1, 1]$. In the following, we denote the vector of the complete set of camera parameters by $\theta = (\mathbf{x}_0, c, s_{xy}, R, \mathbf{X}_0, a_{11}, \dots, a_{tt}, b_{11}, \dots, b_{tt})$.

In the iterative bundle adjustment procedure, we assume we have a parameter estimate θ_{n-1} from the previous iteration. The residual $\mathbf{l}_i$ of point i for the

camera model from the previous iteration is then given by

$$\mathbf{l}_i = \mathbf{x}_i - \mathbf{F}(\theta_{n-1}, \mathbf{X}_i). \tag{5}$$

This equation system is linearized by computing the Jacobian $\mathcal{J}(\theta_{n-1})$ of $\mathbf{F}$ at θ_{n-1} such that we obtain

$$\mathbf{l} \approx \mathcal{J}(\theta_{n-1})\Delta\theta \tag{6}$$

where $\mathbf{l}$ is the concatenation of all $\mathbf{l}_i$ and $\Delta\theta$ is the estimation error in θ that causes the residuals. Usually, one assumes a prior covariance Σ_{ll} on $\mathbf{l}$ describing the inaccuracies in the image position measurements. $\Delta\theta$ is then obtained from a standard linear estimator [3]

$$\Delta\theta = (\mathcal{J}^\top \Sigma_{ll}^{-1} \mathcal{J})^{-1} \mathcal{J} \Sigma_{ll}^{-1} \mathbf{l}\,. \tag{7}$$

Finally, the new parameter estimate θ_n for iteration n is improved according to $\theta_n = \theta_{n-1} + \Delta\theta$. Bundle adjustment needs a good initial estimate θ_0 for the camera parameters in order to ensure that the iterations converge to the correct solution. There exists a great variety of procedures for obtaining initial estimates which have to be specifically chosen for the application (e.g. aerial or near-range photogrammetry).

The quality of the estimation can still be improved by modelling uncertainties in the spatial observations $\mathbf{X}_i$. This can be done by including all spatial observations in the parameter set and updating them in the same manner which requires the additional choice of the covariance Σ_{XX} of the measurements of spatial position [1]. Σ_{XX} regulates the tradeoff between the trust in the accuracy of the image observations on the one hand and the spatial observations on the other hand. For more detailed information on bundle adjustment please refer to [1].

Once the parameter sets $\theta^{(1)}$ and $\theta^{(2)}$ of the two camera models are known, the spatial position $\mathbf{X}^*$ of a newly observed image point ($\mathbf{x}_1^*$ in the first and $\mathbf{x}_2^*$ in the second camera) can be estimated using the same technique. Again, $\mathbf{F}$ describes the stereo camera's mapping from spatial to image coordinates according to Eqns. 1 and 2

$$\mathbf{x}_k^* = \mathbf{F}(\theta^{(k)}, \mathbf{X}^*), \quad k = 1, 2 \tag{8}$$

but this time the θ are kept fixed and the bundle adjustment is computed for estimates of $\mathbf{X}^*$ [1].

3 Gaussian Process Regression

The machine learning algorithm used in our study assumes that the data are generated by a Gaussian Process (GP). Let us call $f(\mathbf{x})$ the non-linear function that maps the D-dimensional input $\mathbf{x}$ to a 1-dimensional output. Given an arbitrary set of inputs $\{\mathbf{x}_i | i = 1, \ldots, m\}$, the joint prior distribution of the corresponding function evaluations $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)]^\top$ is jointly Gaussian:

$$p(\mathbf{f}|\mathbf{x}_1, \ldots, \mathbf{x}_m, \theta) \sim \mathcal{N}(0, K)\ , \tag{9}$$

with zero mean (a common and arbitrary choice) and covariance matrix K. The elements of K are computed from a parameterized covariance function, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \theta)$, where θ now represents the GP parameters. In Sect. 4 we present the two covariance functions we used in our experiments.

We assume that the output observations y_i differ from the corresponding function evaluations $f(\mathbf{x}_i)$ by Gaussian additive i.i.d. noise of mean zero and variance σ^2. For simplicity in the notation, we absorb σ^2 in the set of parameters θ. Consider now that we have observed the targets $\mathbf{y} = [y_1, \ldots, y_m]$ associated to our arbitrary set of m inputs, and would like to infer the predictive distribution of the unknown target y_* associated to a new input $\mathbf{x}_*$. First we write the joint distribution of all targets considered, easily obtained from the definition of the prior and of the noise model:

$$p\left(\left[\begin{array}{c}\mathbf{y}\\ y_*\end{array}\right]\middle|\, \mathbf{x}_1, \ldots, \mathbf{x}_m, \theta\right) \sim \mathcal{N}\left(0, \left[\begin{array}{cc} K + \sigma^2\,\mathcal{I} & \mathbf{k}_* \\ \mathbf{k}_*^\top & k(\mathbf{x}_*, \mathbf{x}_*) + \sigma^2 \end{array}\right]\right) \ , \tag{10}$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_m)]^\top$ is the covariance between y_* and $\mathbf{y}$, and $\mathcal{I}$ is the identity matrix. The predictive distribution is then obtained by conditioning on the observed outputs $\mathbf{y}$. It is Gaussian:

$$p(y_*|\mathbf{y}, \mathbf{x}_1, \ldots, \mathbf{x}_m, \theta) \sim \mathcal{N}(m(\mathbf{x}_*), v(\mathbf{x}_*)) \ , \tag{11}$$

with mean and variance given respectively by:

$$\begin{aligned} m(\mathbf{x}_*) &= \mathbf{k}_*^\top [K + \sigma^2\,\mathcal{I}]^{-1}\mathbf{y} \ , \\ v(\mathbf{x}_*) &= \sigma^2 + k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top [K + \sigma^2\,\mathcal{I}]^{-1}\mathbf{k}_* \ . \end{aligned} \tag{12}$$

Given our assumptions about the noise, the mean of the predictive distribution of $f(\mathbf{x}_*)$ is also equal to $m(\mathbf{x}_*)$, and it is the optimal point estimate of $f(\mathbf{x}_*)$. It is interesting to notice that the prediction equation given by $m(\mathbf{x}_*)$ is identical to the one used in Kernel Ridge Regression (KRR) [2]. However, GPs differ from KRR in that they provide full predictive distributions.

One way of learning the parameters θ of the GP is by maximizing the evidence of the observed targets $\mathbf{y}$ (or marginal likelihood of the parameters θ). In practice, we equivalently minimize the negative log evidence, given by:

$$-\log p(\mathbf{y}|\mathbf{x}_1, \ldots, \mathbf{x}_1, \theta) = \frac{1}{2}\log|K + \sigma^2\,\mathcal{I}| + \frac{1}{2}\mathbf{y}^\top [K + \sigma^2\,\mathcal{I}]^{-1}\mathbf{y} \ . \tag{13}$$

Minimization is achieved by taking derivatives and using conjugate gradients. An alternative way of inferring θ is to use a Bayesian variant of the leave-one-out error (GPP, Geisser's surrogate predictive probability, [4]). In our study we will use both methods, choosing the most appropriate one for each of our two covariance functions. More details are provided in Sect. 4.

4 Experiments

Dataset. We used a robot manipulator holding a calibration target with a flattened LED to record the data items. The target was moved in planes of different

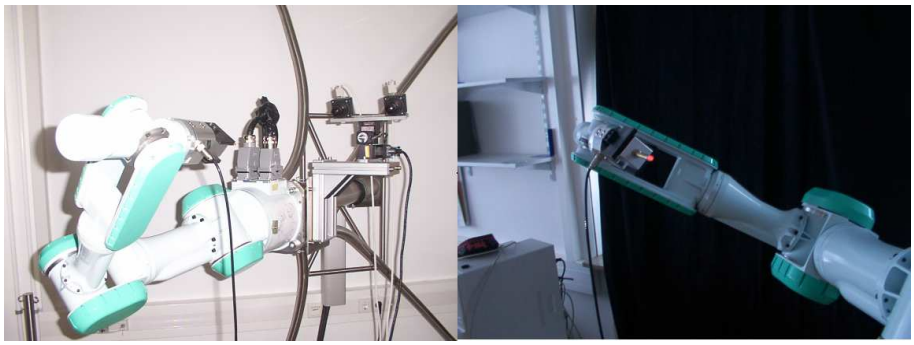

Fig. 1. Robot arm and calibration target, which were used to record the data items.

depths, perpendicular to the axis of the stereo setup. The spatial position of the LED was determined from the position encoders of the robot arm with a nominal positioning accuracy of $0.01mm$. The center of the LED was detected using several image processing steps. First, a threshold operation using the upper 0.01 percentile of the image's gray-scale values predetected the LED. Then a two-dimensional spline was fitted through a window around the image of the LED with an approximate size of $20px$. A *Sobel* operator was used as edge detector on the spline and a *Zhou* operator located the LED center with high accuracy (see [1]). We recorded 992 pairs of spatial and image positions, 200 of which were randomly selected as training set. The remaining 792 were used as test set.

Classical calibration. During bundle adjustment, several camera parameters were highly correlated with others. Small variations of these parameters produced nearly the same variation of the function values of $\mathbf{F}$, which lead to a linear dependency of the columns of $\mathcal{J}$ and thus to a rank deficiency of $\mathcal{J}^\top \Sigma_{ll}^{-1} \mathcal{J}$. Therefore, the parameters of a correlating pair could not be determined properly. The usual way to deal with this problem is to exclude one of the correlating parameters from the estimation. As both the principal point $\mathbf{x}_0$ and the coefficients a_{00}, b_{00} highly correlated with camera yaw and pitch, we assumed them to be zero and excluded them from estimation. Furthermore a_{10}, b_{01}, a_{12} and b_{21} were excluded because of their correlation with s_{xy} and c. As the combination $a_{01} = -b_{10}$ showed a high correlation with the roll angle of the camera, the parameter a_{01} was not estimated and its value was set to b_{01}. The correlations of a_{20} and a_{02} with camera yaw resp. b_{20} and b_{02} with camera pitch were removed by setting a_{20} to b_{02} and a_{02} to b_{20}. Higher degree polynomials in the parameterization of the lens distortion field induce vector fields too complex to correlate with other parameters of the camera such that none had to be switched off due to correlations (see [6] for more detailed information on the parameterization of the camera model). We used a ten-fold crossvalidation scheme to determine whether the corresponding coefficients should be included in the model or not. The error in the image coordinates was assumed to be conditionally independent with $\sigma^2 = 0.25px$, so the covariance matrix Σ_{ll} became diagonal with $\Sigma_{ll} = 0.25 \cdot \mathcal{I}$.

The same assumption was made for Σ_{XX}, though the value of the diagonal elements was chosen by a ten fold cross validation.

Gaussian Process Regression. For the machine learning approach we used both the *inhomogeneous polynomial kernel*

$$k(x, x') = \sigma_\nu^2 \langle x, x' + 1 \rangle^g \tag{14}$$

of degree g and the *squared exponential kernel*

$$k(x, x') = \sigma_\nu^2 \exp\left(-\frac{1}{2} \sum_{d=1}^{D} \frac{1}{\lambda_d^2} (x_d - x'_d)^2 \right) . \tag{15}$$

with automatic relevance determination (ARD). Indeed, the lengthscales λ_d can grow to eliminate the contribution of any irrelevant input dimension.

The parameters σ_ν^2, σ^2 and g of the polynomial covariance function were estimated by maximizing the GPP criterion [4]. The parameters σ_ν^2, σ^2 and the λ_d of the squared exponential kernel were estimated by maximizing their marginal log likelihood [5]. In both cases, we used the conjugate gradient algorithm as optimization method.

We used two different types of preprocessing in the experiments: 1. Scaling each dimension of the input data to the interval $[-1, 1]$; 2. Transforming the input data according to

$$(x_1, y_1, x_2, y_2) \mapsto \left(\frac{1}{2}(x_1 - x_2), \frac{1}{2}(x_1 + x_2), \frac{1}{2}(y_1 - y_2), \frac{1}{2}(y_1 + y_2) \right) . \tag{16}$$

The output data was centered for training.

5 Results

The cross validation for the camera model yielded $\sigma_X = 2mm$ as best a priori estimation for the standard deviation of the spatial coordinates. In the same way, a maximal degree of $t = 3$ for the Chebychev polynomials was found to be optimal for the estimation of the lens distortion. Table 1 shows the test errors of the different algorithms and preprocessing methods.

All algorithms achieved error values under one millimeter. Gaussian Process regression with both kernels showed a superior performance to the classical approach. Fig. 5 shows the position error according to the test points actual depth and according to the image coordinates distance to the lens center, the so called *excentricity*. One can see that the depth error increases nonlinearly with increasing spatial distance to the camera. Calculation of errors shows that the depth error grows quadratically with the image position error, so this behaviour is expected and indicates the sanity of the learned model. Another hint that all of the used algorithms are able to model the lens distortions is the absence of a

Table 1. Test error for bundle adjustment and Gaussian Process Regression with various kernels, computed on a set of 792 data items. Root mean squared error of the spatial residua was used as error measure.

Method	Test Error	Preprocessing
Bundle adjustment	0.38mm	-
Inhomogeneous polynomial	0.29mm	scaled input
Inhomogeneous polynomial	0.28mm	transformed, scaled input
Squared exponential	0.31mm	scaled input
Squared exponential	0.27mm	transformed, scaled input

trend in the right figure. Again, the learning algorithms do better and show a smaller error for almost all excentricities.

The superiority of the squared exponential kernel to the polynomial can be explained by its ability to assign different length scales to different dimensions of the data and therefore set higher weights on more important dimensions. In our experiments $\frac{1}{\lambda_1^2}$ and $\frac{1}{\lambda_3^2}$ were always approximately five times larger than $\frac{1}{\lambda_2^2}$ and $\frac{1}{\lambda_4^2}$, which is consistent with the underlying physical process, where the depth of a point is computed by the disparity in the x-direction of the image coordinates. The same phenomenon could be observed for the transformed inputs, where higher weights where assigned to the x_1 and x_2.

6 Discussion

We applied Gaussian Process Regression to the problem of estimating the spatial position of a point from its coordinates in two different images and compared its performance to the classical camera calibration. Our results show that the generic learning algorithms performed better although maximal physical knowledge was used in the explicit stereo camera modelling.

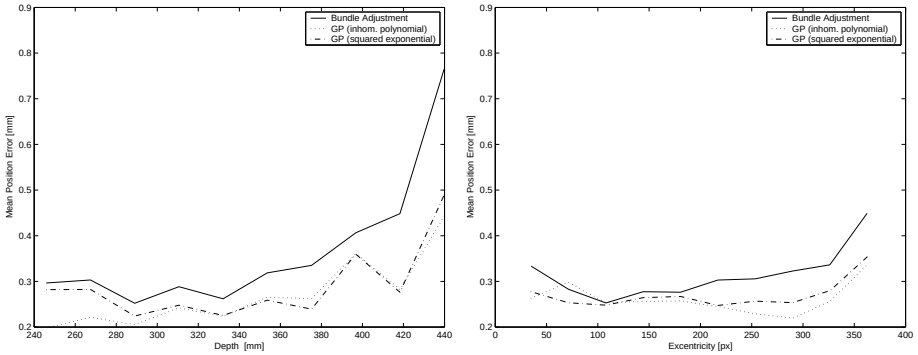

Fig. 2. Position error depending on the actual depth of the test point (left figure) and on the distance to the lens center, the so called *excentricity* (right figure).

An additional advantage of our approach is the mechanical and therefore simple way of model selection, while the correct parametrization of a camera model and elimination of correlating terms is a painful and tedious procedure. Moreover the convergence of the regression process does not depend on good starting values like the estimation of the camera model's parameters does.

A disadvantage of the machine learning approach is that it does not give meaningful parameters such as position and orientation in space or the camera's focal length. Moreover, it does not take into account situations where the exact spatial positions of the training examples are unknown, whereas classical camera calibration allows for an improvement of the spatial position in the training process.

The time complexity for all algorithms is $\mathcal{O}(m^3)$ for training and $\mathcal{O}(n)$ for the computation of the predictions, where m denotes the number of training examples and n the number of test examples. In both training procedures, matrices with a size in the order of the number of training examples have to be inverted at each iteration step. So the actual time needed also depends on the number of iteration steps, which scale with the number of parameters and can be assumed constant for this application. Without improving the spatial coordinates, the time complexity for the training of the camera model would be $\mathcal{O}(p^3)$, where p denotes the number of parameters. But since were are also updating the spatial observations, the number of parameters is upper bounded by a multiple of the number of training examples such that the matrix inversion in (7) is in $\mathcal{O}(m^3)$. An additional advantage of GP is the amount of time actually needed for computing the predictions. Although predicting new spatial points is in $\mathcal{O}(n)$ for GP and the camera model, predictions with the camera model always consume more time. This is due to the improvements of the initial prediction with a linear estimator which again is an iterative procedure involving an inversion of a matrix of constant size at each step.

References

1. Thomas Luhmann: Nahbereichsphotogrammetrie - Grundlagen, Methoden und Anwendungen. Wichmann (2000) [in German]
2. Nello Cristianini and John Shawe-Taylor: Support Vector Machines - and other kernel-based methods. Cambridge University Press (2000)
3. Steven M. Kay: Statistical Signal Processing Vol. I. Prentice Hall (1993)
4. S. Sundararajan, S.S. Keerthi: Predictive Approaches for Choosing Hyperparameters in Gaussian Processes. Neural Computation 13, 1103-1118 (2001). MIT
5. C. K. I. Williams and C. E. Rasmussen: Gaussian processes for regression. Advances in Neural Information Processing Systems 8 pp. 514-520 MIT Press (1996)
6. Fabian Sinz: Kamerakalibrierung und Tiefenschätzung - Ein Vergleich von klassischer Bündelblockausgleichung und statistischen Lernalgorithmen. http://www.kyb.tuebingen.mpg.de/∼fabee (2004) [in German]