Deep Single Image Camera Calibration with Radial Distortion

Manuel López-Antequera* Roger Marí † Pau Gargallo* Yubin Kuang*
Javier Gonzalez-Jimenez‡ Gloria Haro§

Abstract

Single image calibration is the problem of predicting the camera parameters from one image. This problem is of importance when dealing with images collected in uncontrolled conditions by non-calibrated cameras, such as crowd-sourced applications. In this work we propose a method to predict extrinsic (tilt and roll) and intrinsic (focal length and radial distortion) parameters from a single image. We propose a parameterization for radial distortion that is better suited for learning than directly predicting the distortion parameters. Moreover, predicting additional heterogeneous variables exacerbates the problem of loss balancing. We propose a new loss function based on point projections to avoid having to balance heterogeneous loss terms. Our method is, to our knowledge, the first to jointly estimate the tilt, roll, focal length, and radial distortion parameters from a single image. We thoroughly analyze the performance of the proposed method and the impact of the improvements and compare with previous approaches for single image radial distortion correction.

Figure 1. Our method is able to recover extrinsic (tilt, roll) and intrinsic (focal length and radial distortion) parameters from single images (top row). In the bottom row, we visualize the predicted parameters by undistorting the input images and overlaying a horizon line, which is a proxy for the tilt and roll angles.

1. Introduction

Single image calibration deals with the prediction of camera parameters from a single image. Camera calibration is the first step in many computer vision tasks e.g. Structure from Motion, especially in applications where the capturing conditions are not controlled is particularly challenging, such as those relying on crowdsourced imagery.

The process of image formation is well understood and has been studied extensively in computer vision [1], allowing for very precise calibration of cameras when there are enough geometric constraints to fit the camera model. This is a well established practice that is performed daily on an industrial scale, but requires a set of images taken for the purpose of calibration. Geometric based methods can also be used with images taken outside of the lab, performing best on images depicting man-made environments presenting strong cues such as vanishing points and straight lines that can be used to recover the camera parameters [2, 3]. However, since geometric-based methods rely on detecting and processing specific cues such as straight lines and vanishing points, they lack robustness to images taken in unstructured environments, with low quality equipment or difficult illumination conditions.

In this work we present a method to recover extrinsic (tilt, roll) and intrinsic (focal length and radial distortion) parameters given a single image. We train a convolutional neural network to perform regression on alternative representations of these parameters which are better suited for prediction from a single image.

We advance with respect to the state of the art with three main contributions: 1. a single parameter representation for k_1 and k_2 based on a large database of real calibrated cameras. 2. a representation of the radial distortion that is independent from the focal length and more easily learned by the network. 3. a new loss function based on the projection of points to alleviate the problem of balancing hetero-

*Mapillary, *firstname*@mapillary.com
†CMLA, ENS Cachan, mari@cmla.ens-cachan.fr
‡Universidad de Málaga, javiergonzalez@uma.es
§Universitat Pompeu Fabra, gloria.haro@upf.edu

geneous loss components.

To the best of our knowledge, this work is the first to jointly estimate the camera orientation and calibration jointly while including radial distortion.

2. Related Work

Recent works have leveraged the success of convolutional neural networks and proposed using learned methods to estimate camera parameters. Through training, a CNN can learn to detect the subtle but relevant cues for the task, extending the range of scenarios where single image calibration is feasible.

Different components of the problem of learned single image calibration have been studied in the past: Workman et al. [4] trained a CNN to perform regression of the field of view of a pinhole camera, later focusing on detecting the horizon line on images [5], which is a proxy for the tilt and roll angles of the camera if the focal length is known.

Rong et al. [6] use a classification approach to calibrate the single-parameter radial distortion model from Fitzgibbon [7]. Hold-Geoffroy et al. [8] first combined extrinsic and intrinsic calibration in a single network, predicting the tilt, roll and focal length of a pinhole camera through a classification approach. They relied on upright 360 degree imagery to synthetically generate images of arbitrary size, focal length and rotation, an approach that we borrow to generate training data. Classic and learned methods can be combined. In [9], learned methods are used to obtain a prior distribution on the possible camera parameters, which are then refined using classic methods, accelerating the execution time and robustness with respect to fully geometric methods. We do not follow such an approach in this work. However, the prediction produced by our method can be used as a prior in such pipelines.

When training a convolutional neural network for single image calibration, the loss function is an aggregate of several loss components, one for each parameter. This scenario is usually known as multi-task learning [10]. Works in multi-task learning deal with the challenges faced when training a network to perform several tasks with separate losses. Most of these approaches rely on a weighted sum of the loss components, differing on the manner in which the weights are set at training time: Kendall et al. [11] use Gaussian and softmax likelihoods (for regression and classification, respectively) to weight the different loss components according to a task-dependent uncertainty. In contrast to these uncertainty based methods, Chen et al. [12] determine the value of the weights by adjusting the gradient magnitudes associated to each loss term.

When possible, domain knowledge can be used instead of task-agnostic methods in order to balance loss components: Yin et al. [13] perform single image calibration of an 8-parameter distortion model of fisheye lenses. They note the difficulty of balancing loss components of different nature when attempting to directly minimize the parameter errors and propose an alternative based on the photometric error. In this work, we also explore the problem of balancing loss components for camera calibration and propose a faster approach based on projecting points using the camera model instead of deforming the image to calculate the photometric error.

3. Method

We briefly summarize our method and describe the details in subsequent sections.

We train a convolutional neural network to predict the extrinsic and intrinsic camera parameters of a given image. To achieve this, we use independent regressors that share a common pretrained network architecture as the feature extractor, which we fine-tune for the task. Instead of training these regressors to predict the tilt θ, roll ψ, focal length f, and distortion parameters k_1 and k_2, we use proxy variables that are directly visible in the image and independent from each other. To obtain training data for the network, we rely on a diverse panorama dataset from which we crop and distort panoramas to synthesize images taken using perspective projection cameras with arbitrary parameters.

3.1. Camera Model

We consider a camera model with square pixels and centered principal point that is affected by radial distortion that can be modeled by a two-parameter polynomial distortion.

The projection model is the following. World points are transformed to local reference frame of the camera by applying a rotation R and translation t. Let (X, Y, Z) be the coordinates of a 3D point expressed in the local reference frame of the camera. The point is projected to the plane $Z = 1$ to obtain the normalized image coordinates $(x, y) = (X/Z, Y/Z)$. Radial distortion scales the normalized coordinates by a factor d, which is a function of the radius r and the distortion coefficients k_1 and k_2:

$$
\begin{aligned}
r &= \sqrt{x^2 + y^2} \\
d &= 1 + k_1 r^2 + k_2 r^4 \qquad (1) \\
(x_d, y_d) &= (d\, x, d\, y). \qquad (2)
\end{aligned}
$$

Finally, the focal length f scales the normalized and distorted image coordinates to pixels: $(u_d, v_d) = (f\, x_d, f\, y_d)$.

In this work, we do not attempt to recover the position of the images nor the full rotation matrix, as that would require the network to memorize the appearance of the environment, turning our problem of single image calibration into a different problem, so-called *place recognition*.

Instead, we rely on the horizon line as a reference frame, leaving two free parameters: the tilt θ and roll ψ angles of the camera with respect to the horizon. This allows a

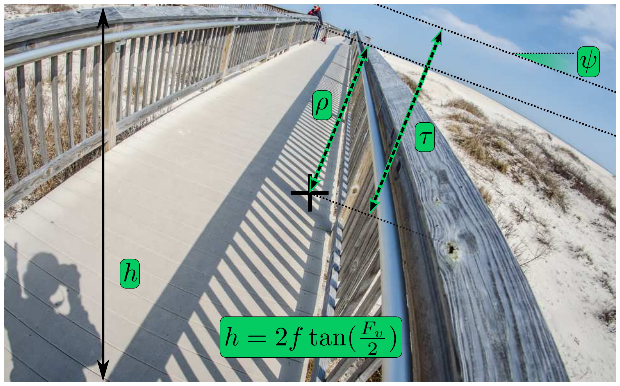


Figure 2. We use an alternative representation for the camera parameters that is based on image cues: The network is trained to predict the distorted offset ρ and vertical field of view F_v instead of the tilt θ and focal length f. The undistorted offset τ is where the horizon would be if there was no radial distortion.

network to be trained using images from a set of locations to generalize well to other places, as long as there is sufficient visual diversity.

Thus, the parameters to be recovered by the network are the tilt and roll angles (θ, ψ), the focal length f and distortion parameters k_1 and k_2.

3.2. Parameterization

As revealed by previous work [4, 5, 8], an adequate parameterization of the variables to predict can greatly benefit convergence and final performance of the network. For the case of camera calibration, parameters such as the focal length or the tilt angles are difficult to interpret from the image content. Instead, they can be better represented by proxy parameters that are directly observable in the image. We begin by following already existing parameterizations and propose new ones required to deal with the case of radially distorted images. We refer the reader to Figure 2 to complement the text in this section.

We start by defining the horizon line as done in [5]: *"The image location of the horizon line is defined as the projection of the line at infinity for any plane which is orthogonal to the local gravity vector."*. This definition also holds true for cameras with radial distortion, however, the projection of the horizon line in the image will not necessarily remain a straight line.[1]

The focal length f is related to the vertical and horizontal fields of view through the image size of height h and width w. The field of view is directly related to the image content and is thus more suitable for the task. We use the vertical field of view, defined as

$$F_v = 2 \arctan \frac{h}{2f}, \tag{3}$$

as a proxy for the focal length. During deployment of the network, the image height h is known and the focal length can be recovered from the predicted F_v.

The roll angle ψ of the camera is directly represented in the image as the angle of the horizon line, not requiring any alternative parameterization.

A good proxy for the tilt angle θ is the distance ρ from the center of the image to the horizon line. Previous work used such a parameterization for pinhole cameras with no distortion [5], however, the presence of radial distortion complicates this relationship slightly. We first define the *undistorted offset* τ as the distance from the image center to the horizon line when there is no radial distortion. It can be expressed as a function of the tilt angle and the focal length:

$$\tau = f \tan(\theta). \tag{4}$$

The *distorted offset* ρ is related with τ by the radial distortion scaling as expressed in Equation 2.

3.2.1 Distortion coefficients in real cameras

We simplify the radial distortion model by expressing k_2 as a function of k_1. This decision was initially motivated by a practical consideration: independently sampling k_1 and k_2 often results in unrealistically distorted images. For images from real lenses, the distortion coefficients seem to lie in a manifold. We confirm this by studying the distribution of k_1 and k_2 on a large collection of camera calibrations.

We use Structure from Motion (SfM) with self-calibration to perform reconstructions on image sequences taken with real cameras to estimate their parameters. We

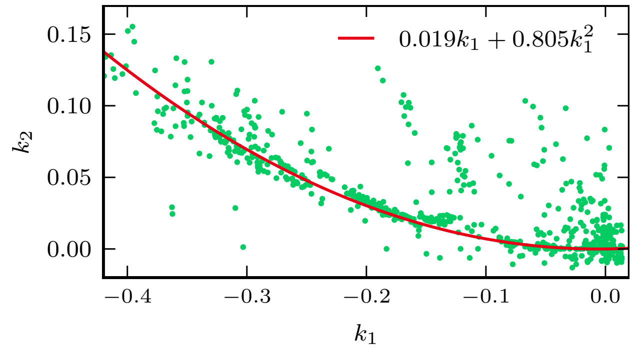
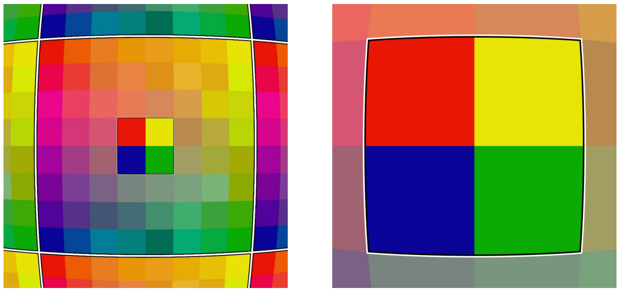


Figure 3. A distribution of k_1 and k_2 recovered from a large set of SfM reconstructions reveals that, for many real cameras, these parameters lie close to a one-dimensional manifold. We rely on this to simplify our camera model such that k_2 is a function of k_1.

[1] If there is radial distortion, the horizon line (and any other straight lines) will only be projected as a straight line in the image if it passes through the center of the image.



$f = 0.8, k_1 = -0.13$ $f = 3.2, k_1 = -2.05$

Figure 4. The apparent radial distortion $\hat{k}_1$ represents the distortion effect independently of the focal length f. In these images we fix $k_2 = 0$ and vary k_1 and f while keeping a constant value of $\hat{k}_1 = -0.2$. Note that the curvature of the lines remains constant after zooming in.

downloaded a collection of 1000 street-level imagery sequences of 100 geotagged images each from Mapillary.[2] These sequences were captured by a diverse set of over 300 cameras, including most popular consumer-grade smartphones and action cameras that have been in the market for the last 4 years. Sequences were selected such that the SfM reconstructions would constrain the camera parameters: they present loop closures or trajectories that are not a straight line (as reported by the GPS geotag). The camera parameters of each sequence are recovered as part of the reconstruction through bundle adjustment [14]. Since SfM is sensitive to the initial calibration parameters, we repeat the reconstructions initializing with the newly estimated camera parameters until convergence.

The resulting set of radial distortion coefficients is shown in Figure 3, confirming our initial observation. We obtain an analytic expression as a model of this distribution by fitting a second degree polynomial

$$k_2 = 0.019k_1 + 0.805k_1^2. \tag{5}$$

We observe two main groups of lenses: Fisheye lenses, exhibiting strong radial distortion, with $k_1 < 0$ and positive k_2 increasing in a quadratic manner with the magnitude of k_1, and conventional lenses, with both k_1 and k_2 close to 0.

3.2.2 Apparent distortion

Inferring the value of k_1 from an image is not trivial. A human observer would probably make a guess based on the bending of straight lines. Nevertheless, both the focal length and the radial distortion coefficients determine such bending. Radial distortion is more noticeable towards the boundaries of the image but, as the focal length increases, we gradually see a smaller crop of the center of the image.

[2] Mapillary is a crowdsourced street-level imagery platform.

As with the focal length and the tilt, we propose to use an alternative parameterization to express k_1 in terms of a visible magnitude, i.e. the distortion that is observed in the image. We will then train the network to predict an *apparent* distortion coefficient that we denote as $\hat{k}_1$.

As stated in Section 3.1, the camera model projects points (X, Y, Z) in the camera reference frame to 2D normalized camera coordinates $(x, y) = ({}^X\!/\!_Z, {}^Y\!/\!_Z)$.

In the absence of radial distortion, pixels are obtained from the undistorted normalized coordinates as $(u, v) = (f\,x, f\,y)$. When there is radial distortion, the radius of the normalized coordinates is first distorted before being converted to pixels $(u_d, v_d) = (f\,d\,x, f\,d\,y)$. In other words, since the distortion is applied to the normalized image coordinates, the visual effect not only depends on the distortion parameters, but also also on the focal length.

Instead, we seek to represent the distortion effect as a relationship between the distorted (u_d, v_d) and undistorted pixels (u, v). Let us begin by expressing the radius of a point r in normalized coordinates and its equivalent in pixel units r_{px} :

$$r = r_{\text{px}}/f. \tag{6}$$

The same relationship holds when there is distortion:

$$r^{(d)} = r_{\text{px}}^{(d)}/f. \tag{7}$$

The undistorted and distorted points in normalized camera coordinates are related by Eq. 2 and can be expressed as

$$r^{(d)} = r\left(1 + k_1 r^2 + k_2 r^4\right), \tag{8}$$

in which we substitute r and $r^{(d)}$ from Eqs. 6 and 7 to obtain the relationship between the radii in pixel units, obtaining the *apparent distortion* coefficients $\hat{k}_1$and $\hat{k}_2$:

$$r_{\text{px}}^{(d)} = r_{\text{px}}\left(1 + \overbrace{\frac{k_1}{f^2}}^{\hat{k}_1} r_{px}^2 + \overbrace{\frac{k_2}{f^4}}^{\hat{k}_2} r_{px}^4\right) \tag{9}$$

$$\boxed{\hat{k}_1 = k_1/f^2} \tag{10}$$

Observe that for a fixed value of k_1, $\hat{k}_1$ decreases as f increases and vice-versa, representing the effect of radial distortion independently from f as shown in Figure 4. Given a prediction of $\hat{k}_1$ and f, both recoverable from the network outputs, k_1 and k_2 can be retrieved through equations 5 and 10.

In summary, we represent a camera's intrinsic and extrinsic parameters with $\mathbf{\Omega} = (\psi, \rho, F_v, \hat{k}_1)$, where ψ is the roll angle, ρ is the distorted offset, F_v is the vertical field of view and $\hat{k}_1$ is the apparent radial distortion.

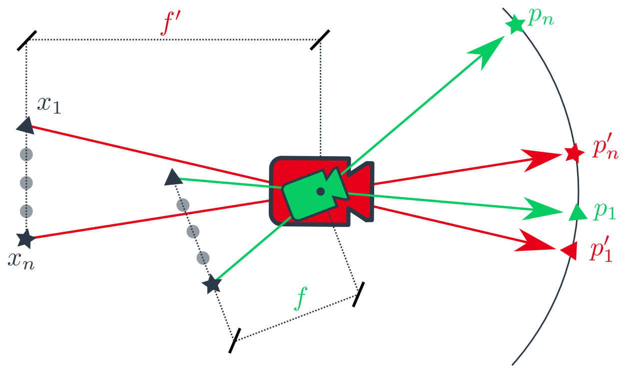


Figure 5. An illustration of the projections used for the bearing loss simplified by reducing it to two parameters: tilt θ (represented by the orientation of the cameras) and focal length f. Two cameras are used to project a regular grid of points $\boldsymbol{x}_1 \dots \boldsymbol{x}_n$ onto the unit sphere. The points $\boldsymbol{p}_1 \dots \boldsymbol{p}_n$, shown in green, are projected using the ground truth camera parameters $\boldsymbol{\Omega} = (\theta, f)$. The points $\boldsymbol{p}'_1 \dots \boldsymbol{p}'_n$ are projected using the predicted parameters $\boldsymbol{\Omega}' = (\theta', f')$ and are shown in red. We obtain gradients for the predicted camera parameters $\boldsymbol{\Omega}'$ through backpropagation of the mean squared distance between points $\boldsymbol{p}'_1 \dots \boldsymbol{p}'_n$ and $\boldsymbol{p}_1 \dots \boldsymbol{p}_n$.

3.3. Bearing Loss

When a single architecture is trained to predict parameters with different magnitudes, special care must be taken to weigh the loss components such that the estimation of certain parameters do not dominate the learning process. We notice that for the case of camera calibration, instead of optimizing the camera parameters separately, a single metric based on the projection of points with the estimated and ground truth camera parameters can be used. Let us begin with the observation that a camera model is essentially a simplified bidirectional mapping from pixel coordinates in the image plane to bearings (direction vectors) in 3D [15, 16]. The camera intrinsic and extrinsic parameters determine the direction of one such bearing for each pixel in the image. The proposed loss measures errors on these direction vectors instead of individual parameter errors, achieving the goal of representing all the parameter errors as a single metric.

Given an image taken with known camera parameters $\boldsymbol{\Omega} = (\psi, \rho, F_v, \hat{k}_1)$ and a prediction of such parameters given by the network $\boldsymbol{\Omega}' = (\psi', \rho', F'_v, \hat{k}'_1)$, the bearing loss is calculated as follows.

First, a regular grid of points $\boldsymbol{x}_1 \dots \boldsymbol{x}_n$ is projected from the image plane onto the unit sphere using the ground truth parameters $\boldsymbol{\Omega}$ obtaining the ground truth bearings $\boldsymbol{p}_1 \dots \boldsymbol{p}_n$.[3]

[3]In order to project the points, the original parameter set $\psi, \theta, f, k_1, k_2$ required by the camera model is recovered from the proxy parameters $\boldsymbol{\Omega}$ using equations 2, 3, 4, 5.

Then, the parameters $\boldsymbol{\Omega}'$ predicted by the network are used to project the same grid points onto the unit sphere, obtaining the set of predicted bearings $\boldsymbol{p}'_1 \dots \boldsymbol{p}'_n$. We define the bearing loss as the mean squared deviation between the two sets of bearings:

$$L(\boldsymbol{\Omega}', \boldsymbol{\Omega}) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{p}'_i - \boldsymbol{p}_i)^2. \tag{11}$$

This process is illustrated in Figure 3.3.

To optimize this loss, the mapping from pixels to bearings must be differentiable. This includes the radial undistortion step, which does not have a closed-form solution. Although there are several solutions for r in $r^{(d)} = r(1 + k_1 r^2 + k_2 r^4)$, the correct solution is that where r is closest to $r^{(d)}$, which can be reliably found by performing fixed point iteration[4] of the function $r_{n+1} = r^{(d)}/(1+k_1 r_n^2 + k_2 r_n^4)$ initialized at $r_0 = r^{(d)}$. This process is differentiable and can be used during training to backpropagate gradients through the bearing loss.

3.3.1 Disentangling sources of loss errors

The proposed loss solves the task balancing problem by expressing different errors in terms of a single measure. However, using several camera parameters to predict the bearings introduces a new problem during learning: the deviation of a point from its ideal projection can be attributed to more than one parameter. In other words, an error from one parameter can backpropagate through the bearing loss to other parameters.

For example, picture a scenario where, for a training sample, the network predicts all parameters perfectly except for an excessively small field of view: The predicted bearings $\boldsymbol{p}'_1 \dots \boldsymbol{p}'_n$ are projected onto a smaller area on the unit sphere than the ground truth bearings $\boldsymbol{p}_1 \dots \boldsymbol{p}_n$.

In this case, there is more than one parameter that could be modified to decrease this distance: both the focal length and the radial distortion parameters can be changed to decrease the loss, but only the value of the focal length should be modified, as the radial distortion has been perfectly predicted in this example. In other words, there will be gradients propagating back through both parameters, even though one of them is correct, causing the network to deviate from the optimal solution. In practice, this slows down learning and causes the accuracy to stagnate.

To avoid this problem, we disentangle the bearing loss,

[4]We implement this by repeatedly iterating and breaking on convergence in PyTorch, but it rarely requires more than 4 steps to converge, so it could be unrolled to a set number of iterations if using a framework that relies on fixed computational graphs.

Parameter	Distribution	Values
Pan ϕ	Uniform	$[0, 2\pi)$
Distorted offset ρ	Normal	$\mu = 0.046, \sigma = 0.6$
Roll ψ	Cauchy	$x_0 = 0, \gamma \in \{0.001, 0.1\}$
Aspect ratio w/h	Varying	$\{1/1\ 9\%, 5/4\ 1\%, 4/3\ 66\%,$ $3/2\ 20\%, 16/9\ 4\%\}$
Focal length f	Uniform	$[13, 38]$
Distortion k_1	Uniform	$[-0.4, 0]$
Distortion k_2		$k_2 = 0.019k_1 + 0.805k_1^2$

Table 1. Distribution of the camera parameters used to generate our training and validation sets. Units: f- mm, ψ- radians, ρ- fraction of image height.

evaluating it individually for each parameter ψ, ρ, F_v, $\hat{k}_1$:

$$L_\psi = L((\psi \quad, \rho^{GT}, F_v^{GT}, \hat{k}_1^{GT}), \mathbf{\Omega})$$
$$L_\rho = L((\psi^{GT}, \rho \quad, F_v^{GT}, \hat{k}_1^{GT}), \mathbf{\Omega})$$
$$L_{F_v} = L((\psi^{GT}, \rho^{GT}, F_v \quad, \hat{k}_1^{GT}), \mathbf{\Omega})$$
$$L_{\hat{k}_1} = L((\psi^{GT}, \rho^{GT}, F_v^{GT}, \hat{k}_1 \quad), \mathbf{\Omega})$$

$$L^* = \frac{L_\psi + L_\rho + L_{F_v} + L_{\hat{k}_1}}{4} \tag{12}$$

This modification of the loss function greatly increases convergence and final accuracy, while maintaining the main advantage of the bearing loss of expressing all parameter errors in the same units.

3.4. Dataset

We use the SUN360 panorama dataset [17] to artificially generate images taken with by cameras with arbitrary pan ϕ, tilt θ, roll ψ, focal length f and distortion k_1. High resolution images of 9104×4452 pixels are used to render the training and evaluation images as follows:

First, we divide the SUN360 dataset into training, evaluation and test sets of 55681, 1298 and 165 images, respectively. Separating the panorama dataset before generating the perspective images ensures that no panoramas are used to generate crops that end up in different datasets.

Then, from each panorama in the training and validation sets, we generate seven perspective images by randomly sampling the pan ϕ, offset ρ, roll ψ, aspect ratio, focal length f and the distortion coefficient k_1 from the probability distributions found in Table 1, resulting in a dataset of $389,767$ training and $9,086$ validation images.

In a practical scenario, the distribution of the training set should be designed to mimic that of the images that will be used when deploying the network. For this paper we have selected simple distributions that are consistent with those found in large online image databases: we take the same distributions as in previous work [8], except for the inclusion of k_1 for radial distortion. Additionally, we have modified the distribution of f to be uniform in order to avoid obtaining images with large focal lengths since the effect of radial distortion in such images is negligible[5].

For the test set we followed a different approach, sampling from the 165 panoramas in the panorama test set more extensively and evenly by taking 100 crops from each panorama and using uniform distributions also for the roll angle $\psi \sim \mathcal{U}(-\pi/2, \pi/2)$, distorted offset $\rho \sim \mathcal{U}(-1.2, 1.2)$ and aspect ratios $w/h \sim \mathcal{U}\{1/1, 5/4, 4/3, 3/2, 16/9\}$. This results in $16,500$ images for our test set.

4. Experiments

We use a densenet-161 [18] pretrained on ImageNet [19] as a feature extractor and replace the classifier layer with four regressors, each consisting of a ReLU-activated hidden layer of 256 units followed by the output unit.

As explained before, images are generated with a variety of aspect ratios. We experimented with several ways of feeding such images to the network: resizing, center-cropping and letterboxing. Previous authors noticed better results by square-cropping the images [5]. Like Hold-Geoffroy et al. [8], we obtained best results by resizing the images to a square. Even though there is deformation in the image when its aspect ratio is changed, it appears to be that keeping all of the image content by not cropping the image is preferable to any negative effect the warping itself may produce. All images are thus scaled to 224×224 pixels before feeding them to the network.

We train the network by directly minimizing parameter errors as well as using the proposed bearing loss. In the first case, we minimize a sum of weighted Huber losses:

$$L^H = w_\psi L_\psi^H + w_\rho L_\rho^H + w_{F_v} L_{F_v}^H + w_{\hat{k}_1} L_{\hat{k}_1}^H \tag{13}$$

For the bearing loss, the predicted and ground truth parameters of each image are used to project bearings as described in Section 3.3.

In both cases we minimize the losses using an Adam optimizer with learning rate 10^{-4} in batches of 42 images. Through early stopping we finish training after 8-10 epochs. We use a step learning rate decay such that the learning rate is reduced by 30% at the end of each epoch.

4.1. Evaluation of the Loss Functions

We evaluate the bearing loss from Section 3.3 and compare it to the weighted Huber loss (Eq. 13). The Huber loss with unit weights performs better.

The results are comparable, except for the prediction of $\hat{k}_1$ which does not perform as well with the bearing loss.

[5] An additional problem with the choice of a long-tailed distribution to sample the focal length is that it may produce values for f equivalent to very low resolution crops.

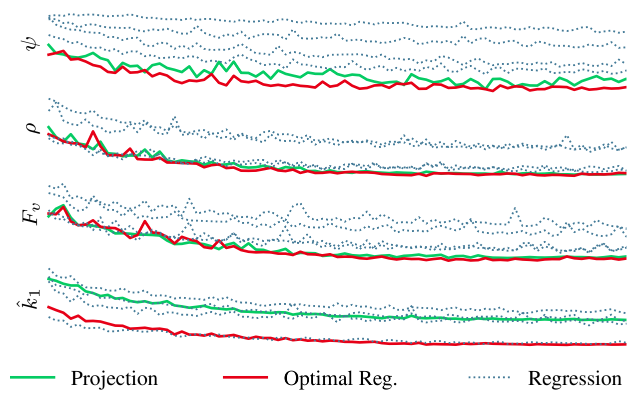


Figure 6. Optimal selection of the weights when combining different loss components can greatly influence training. In gray, models trained with one weight in $\{w_\psi, w_\rho, w_{F_v}, w_{\hat{k}_1}\}$ set to 100 and the rest to 1. In red, a model trained with all weights set to 1. In green, a model trained with the bearing loss. These results indicate that selecting appropriate weights is important for this task, but that with the proposed parameterization, selecting unit weights yields results that are better than the proposed bearing loss.

However, that may not always be the case, for example, when using a different camera model, as reported by Yin et al. [13], or when using a different parameterization than the one we propose here. It just happens that this parameterization is well suited to be trained with unit weights. To illustrate the effect of selecting less optimal weights, we have trained several networks using the weighted sum of Huber losses (Eq. 13) with different sets of weights and compare the resulting validation error curves in Figure 6.

For the rest of the paper, *our approach* or *our network* refer to a network trained by minimizing the proposed parameters $(F_v, \hat{k}_1, \rho, \psi)$ directly using a sum of Huber losses (Eq. 13) with $w_\psi = w_\rho = w_{F_v} = w_{\hat{k}_1} = 1$.

4.2. Effect of Distortion Parameterization

We compare the proposed parameterizations for the radial distortion coefficient and the radially distorted offset with a naive approach. For this purpose, we train a baseline network to predict the distortion coefficient and undistorted offset (k_1, τ), instead of the proposed apparent distortion and distorted offset $(\hat{k}_1, \rho)$. The remaining parameters (ψ, F_v) are as in our network. In both cases, we minimize the sum of Huber losses from Eq. 13 with unit weights. The remaining settings for the experiment are as described in Section 4. After training, we compare the predictions of both networks on the test set. Figure 7 shows scatter plots comparing the predictions of $\hat{k}_1$ and k_1, as well as those of the distorted offset ρ and undistorted offset τ, revealing that the proposed parameterization is easier to learn (more accurately predicted) than the baseline.

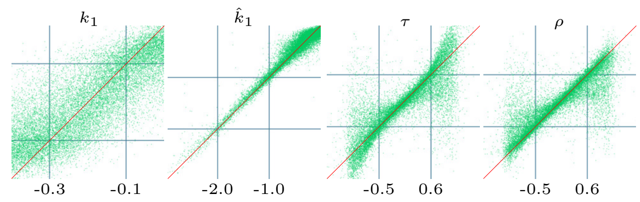


Figure 7. A comparison of the predictions of two networks: Our approach (predicting the apparent distortion $\hat{k}_1$ and the distorted offset ρ) and a baseline predicting the distortion coefficient k_1 and the undistorted offset τ. The horizontal and vertical axes in each plot represent the ground truth and predicted values, respectively. The diagonal line indicates a perfect prediction. Learning to predict $\hat{k}_1$ is an easier task than directly predicting k_1, as it is independent of the focal length f. The distorted offset ρ is also easier to predict than the undistorted offset τ, since it is directly visible in the image and is independent of the distortion.

4.3. Error Distributions

There is a lack of consensus when evaluating single image calibration networks: some previous works follow a classification approach and directly report accuracy values [5]. Others establish a threshold on the regression errors and also report accuracy values [8, 4]. Yin et al. [13] report peak signal-to-noise ratio structural similarity errors. Rong et al. [6] use a metric based on straight line segment lengths that is only meaningful for radial distortion correction. Hold-Geoffroy et al. [8] report error distributions grouped according to the ground truth values in a box-percentile chart.

We follow the evaluation procedure from [8] of reporting the error distributions of the predicted parameters. However, instead of reporting errors in terms of the alternative parameterization used to ease learning (roll ψ, distorted offset ρ, field of view F_v and apparent radial distortion $\hat{k}_1$), we report the errors in: roll ψ, tilt θ, focal length f and radial distortion coefficient k_1, since they are more commonly used than the proposed parameterization and can be easily compared with other approaches.

These error distributions are shown in Figure 8. The diagonal plots show the error distribution of the prediction of each parameter with respect to its ground truth value. We also study the error distributions of each parameter with respect to the ground truth values of the other parameters. This is shown in the off-diagonal plots, revealing some interesting insights. For example, the plots from the first column indicate the error distributions of all parameters with respect with the ground truth value of the tilt angle θ. Notice that when θ is small (i.e. when the horizon is close to the center of the image), the prediction errors for the tilt and roll angles are small as well, while the errors for the focal length f and the radial distortion coefficient k_1 are relatively large. This is expected as many lines in the world are vertical and parallel to the image plane when the tilt is zero,

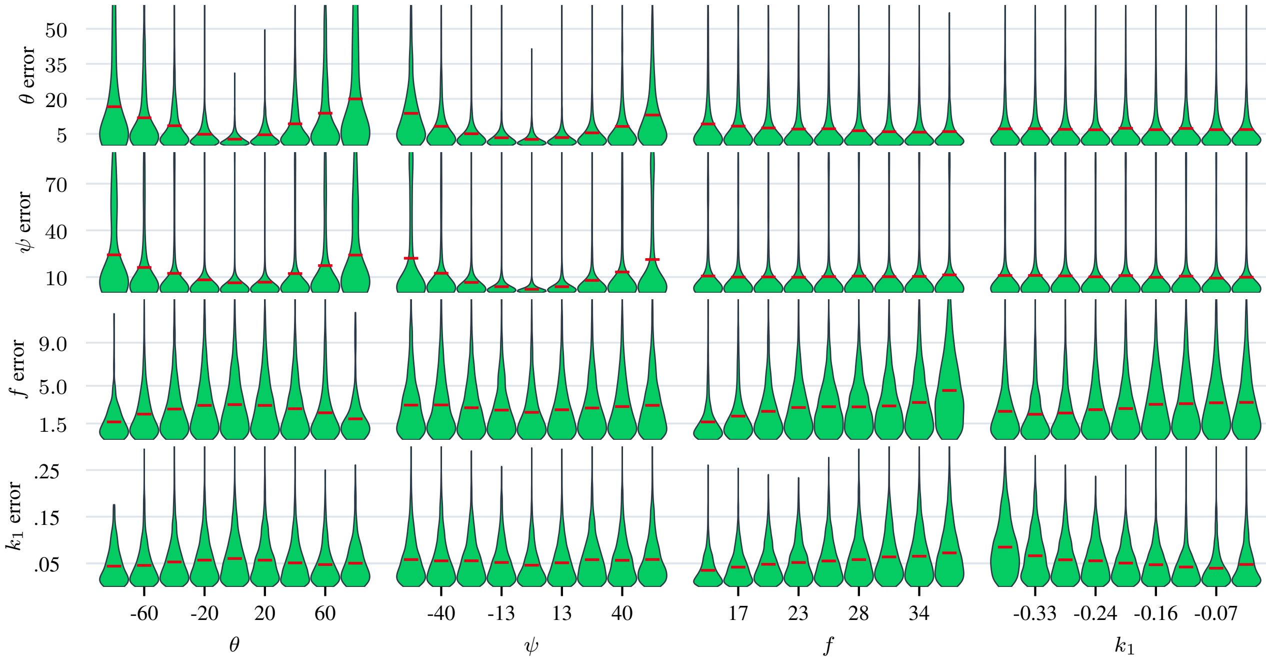


Figure 8. Errors on the test set of 16,500 images. The horizontal axis represents the ground truth values, while the vertical axis represents the absolute error of the predictions. We show errors as a function of the ground truth value of the same parameter, as well as as a function of other parameter's ground truth values.

providing no information for predicting the focal length.

As stated in Section 3.4, the training set should be generated to replicate the distribution of images that will be seen when deploying such a network. We expect the error distributions to change according to the distribution of the training data, since the span of these data directly relates to the difficulty of the problem. For this reason, the absolute errors seen in Figure 8 are not as relevant as the relationships among them. These errors should be studied for the specific application domain where a network like this is to be deployed.

4.4. Comparison with Geometric-based Undistortion

Plumb-line methods are a well-known approach for single image undistortion in the wild. These techniques predict lens distortion based on the detected curvature of lines. We compare our method to a state of the art plumb-line algorithm by Santana-Cedrés et al. [20]. Since they use a different parameterization for the radial distortion, we compare the photometric mean squared error on images from the test set undistorted by both methods [21].

We obtain a lower MSE in 89% of the images in the test set, but notice differences depending on the category of the source panorama: for outdoor images with few or no line segments (nature landscapes or open spaces with trees and monuments), our method performs best in more than 90% of the images. The difference narrows down for indoor and urban imagery, with our method outperforming [20] in 70-90% of the cases, depending on the category. This is expected as there are more line segments in images from these classes that the plumb-line algorithm can rely on.

We attribute the higher accuracy of learned undistortion to the ability of the model to interpret semantic cues. The reader is referred to the supplementary material for a detailed discussion.

5. Conclusions

We present a learning-based method that jointly predicts the extrinsic and intrinsic camera parameters, including radial distortion. The proposed parameterization is disentangled from the focal length and well suited for prediction. We also introduce a new loss function to overcome the problem of loss balancing. Finally, we validate the superior performance of the proposed method against geometric-based undistortion methods.

In future work, we will explore distortion calibration with single-parameter distortion models [7, 22]. More importantly, we will apply single image camera calibration in large-scale structure from motion on crowdsourced images with diverse camera models, where we see the potential of learning-based methods to enhance the robustness of the system.

References

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

[2] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *International Journal of Computer Vision*, vol. 4, no. 2, pp. 127–139, mar 1990. [Online]. Available: https://doi.org/10.1007/BF00127813 1

[3] J. Deutscher, M. Isard, and J. MacCormick, "Automatic Camera Calibration from a Single Manhattan Image," in *Computer Vision — ECCV 2002*, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 175–188. 1

[4] S. Workman, C. Greenwell, M. Zhai, R. Baltenberger, and N. Jacobs, "DEEPFOCAL: A method for direct focal length estimation," in *2015 IEEE International Conference on Image Processing (ICIP)*, sep 2015, pp. 1369–1373. 2, 3, 7

[5] S. Workman, M. Zhai, and N. Jacobs, "Horizon Lines in the Wild," pp. 1–12, 2016. [Online]. Available: http://arxiv.org/abs/1604.02129 2, 3, 6, 7

[6] J. Rong, S. Huang, Z. Shang, and X. Ying, "Radial Lens Distortion Correction Using Convolutional Neural Networks Trained with Synthesized Images," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham: Springer International Publishing, 2017, pp. 35–49. 2, 7

[7] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I. 2, 8

[8] Y. Hold-Geoffroy, K. Sunkavalli, J. Eisenmann, M. Fisher, E. Gambaretto, S. Hadap, and J.-F. Lalonde, "A Perceptual Measure for Deep Single Image Camera Calibration," *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Online]. Available: http://arxiv.org/abs/1712.01259 2, 3, 6, 7

[9] M. Zhai, S. Workman, and N. Jacobs, "Detecting Vanishing Points Using Global Image Context in a Non-Manhattan World," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5657–5665, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780979/ 2

[10] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997. 2

[11] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[12] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," *arXiv preprint arXiv:1711.02257*, 2017. 2

[13] X. Yin, X. Wang, J. Yu, M. Zhang, P. Fua, and D. Tao, "FishEyeRecNet: A Multi-Context Collaborative Deep Network for Fisheye Image Rectification," pp. 1–16, 2018. [Online]. Available: http://arxiv.org/abs/1804.04784 2, 7

[14] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372. 4

[15] P. Sturm and S. Ramalingam, "A Generic Concept for Camera Calibration," in *8th European Conference on Computer Vision (ECCV '04)*, ser. Lecture Notes in Computer Science (LNCS), J. Matas and T. Pajdla, Eds., vol. 3022/2004. Prague, Czech Republic: Springer-Verlag, May 2004, pp. 1–13. [Online]. Available: https://hal.inria.fr/inria-00524411 5

[16] S. Ramalingam, S. K. Lodha, and P. Sturm, "A generic structure-from-motion framework," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 218–228, 2006. [Online]. Available: https://hal.inria.fr/inria-00384319 5

[17] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba, "Recognizing scene viewpoint using panoramic place representation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2695–2702. 6

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3. 6

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. 6

[20] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, and L. Alvarez, "An iterative optimization algorithm for lens distortion correction using two-parameter models," *Image Processing On Line*, vol. 6, pp. 326–364, 2016. 8

[21] R. Szeliski, "Prediction error as a quality metric for motion and stereo," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 781–788 vol.2. 8

[22] C. Ishii, Y. Sudo, and H. Hashimoto, "An image conversion algorithm from fish eye image to perspective image for human eyes," in *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, vol. 2. IEEE, 2003, pp. 1009–1014. 8