

A Robust Chessboard Detector for Geometric Camera Calibration

Mathis Hoffmann, Andreas Ernst, Tobias Bergen, Sebastian Hettenkofer and Jens-Uwe Garbas

Fraunhofer Institute for Integrated Circuits IIS, Am Wolfsmantel 33, 91058 Erlangen, Germany

Keywords: Chessboard Detection, Camera Calibration, Endoscope Calibration, Integral Image, Checkerboard Detection.

Abstract: We introduce an algorithm that detects chessboard patterns in images precisely and robustly for application in camera calibration. Because of the low requirements on the calibration images, our solution is particularly suited for endoscopic camera calibration. It successfully copes with strong lens distortions, partially occluded patterns, image blur, and image noise. Our detector initially uses a sparse sampling method to find some connected squares of the chessboard pattern in the image. A pattern-growing strategy iteratively locates adjacent chessboard corners with a region-based corner detector. The corner detector examines entire image regions with the help of the integral image to handle poor image quality. We show that it outperforms recent solutions in terms of detection rates and performs at least equally well in terms of accuracy.

1 MOTIVATION

Endoscopic procedures are frequently used in the treatment of various diseases and internal injuries. Minimally invasive surgery with endoscopic instruments is performed, for example, on abdominal organs (laparoscopy), joints (arthroscopy), or the brain (neurosurgery). Endoscopy requires good orientation, coordination, and fine motor skills of the surgeon. Therefore, computer systems that offer navigation support to the physician are of increasing significance. Navigation systems that relate the view through the endoscope to the geometry of the surgical site require a calibrated camera. In this context, calibration refers to the process of estimating the intrinsic camera parameters. The intrinsic parameters of a distorted pinhole camera model consist of the focal length, the principal point, as well as radial and tangential distortions. Usually, camera calibration is a two-step process: First, a known calibration pattern is detected in the images. Second, the calibration parameters are estimated based on correspondences between points on the pattern and their projections in the images. In this paper, we focus on the first step. We present a new method for detecting the calibration target robustly and precisely under hard constraints.

Camera calibration for endoscopic applications poses several challenges. Endoscopes often have wide-angle lenses with typical viewing angles between 90° and 120° . Therefore, distortion effects are very strong. Due to the optical setup of the endo-

scope, the light source is close to the optical center. This often causes strong inhomogeneity, glare, vignetting effects, and high image noise in badly illuminated regions. Within a clinical environment, non-technical staff must be able to perform the calibration process quickly. Consequently, the calibration method must reliably handle motion blur, defocussing, and recordings of partially captured patterns.

Various patterns have been developed. Still, the planar chessboard pattern is most established. Mallon et al. (Mallon and Whelan, 2007) have shown that the chessboard pattern outperforms circle patterns in the case of strong perspective or radial distortion. Self-identifying targets (like ARTags (Fiala and Shu, 2005)) are more complex and require high resolution and low-noise images. We will therefore focus on a classical chessboard pattern. Nonetheless, the proposed method can be adapted to other calibration targets as well.

2 RELATED WORK

A variety of calibration approaches have been proposed to determine intrinsic camera parameters from a set of images of a known calibration pattern. Substantial contributions include early works by Tsai (Tsai, 1987), Heikkilä and Silven (Heikkilä and Silven, 1997), and Zhang (Zhang, 2000). Various improvements and extensions of these methods have been suggested for application in endoscopic camera

calibration (Zhang et al., 2000; Wengert et al., 2006). Moreover, alternative camera models have been suggested, tailored to the optical setup of endoscopic cameras (Stehle et al., 2007; Li et al., 2008). Barreto et al. addressed the requirement of an easy-to-use calibration process and developed a single-shot calibration method to allow for an endoscope calibration procedure with minimal effort for the surgeon (Barreto et al., 2009). Ruffi et al. also proposed a method for the detection of a calibration pattern (Ruffi et al., 2008). Their approach is based on an adaptive thresholding of the input images, followed by a binary contour finder to detect the quadrangles of a chessboard pattern. A variation of this approach is implemented in the OpenCV computer vision toolbox. More recently, Placht et al. presented a chessboard detection method (*ROCHADE*) for highly distorted images. They detect edges based on a gradient image to create a graph representation of the detected chessboard quadrangles. Graph-matching is then applied to confirm correct detection results. Both methods require the entire calibration pattern to be visible in the image. This is a major drawback in the context of endoscopy. It impedes to capture chessboard corners in the outer image regions, where barrel distortion has the strongest effect. So, the most valuable image regions cannot contribute any point correspondences. Fuersattel et al. improved the *ROCHADE* method to also handle partially visible chessboards by including a subgraph-matching strategy (called *OCPAD*). Sun et al. proposed a chessboard detection algorithm that locates corners robustly under inhomogeneous illumination and also handles deformed chessboards (Sun et al., 2008). They detect corner points by evaluating bright and dark sequences along circles around corner point candidates.

Most of the established approaches either apply a binarization procedure to separate dark and bright regions or extract edges from the images to localize chessboard corners. In contrast, we propose to evaluate region statistics to robustly and accurately detect chessboard corners under difficult image conditions. Considering entire image regions (corresponding to chessboards quadrangles) makes the approach insensitive to image noise and blurry edges. Our method is explicitly designed to handle partially visible chessboards with strong distortion effects under inhomogeneous illumination.

3 PROPOSED METHOD

Figure 1 shows an overview of our method. The individual steps are illustrated in Figure 2. The process

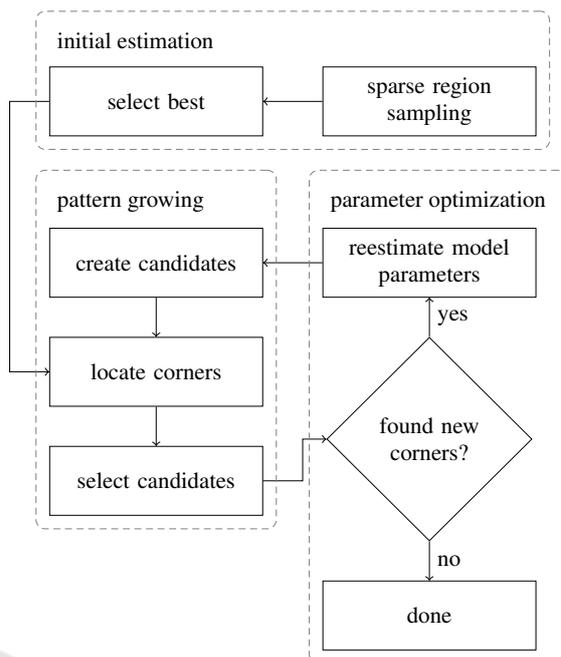


Figure 1: Overview of the proposed method. The process is divided into an initial estimation, a pattern growing and a parameter optimization phase.

consists of three phases. The initialization phase uses a sparse sampling strategy to find an initial guess for the chessboard position and size (Figures 2a and 2b). The guess is then improved to subpixel precise corner locations and allows an initial camera parameter estimation (Figure 2c). The pattern-growing phase iteratively searches new chessboard corners in the vicinity of detected corners and updates the camera parameters in the optimization phase (Figures 2d to 2f). The following sections explain the three phases in detail.

3.1 Initial Estimation

Our method requires an initial estimate of the chessboard size, pose and position. Therefore, we regularly scan the image with a 3×3 chessboard model to find positions of high correspondence between the model and the image (Figures 2a and 2b).

At each position, we sample the intensities of 5 points within each of the 9 chessboard patches (see Figure 3). The regular structure of a chessboard allows us to divide it into two groups of homogeneous intensities. A successful initial guess is characterized by a high intensity difference between both groups and a low intensity variance within each group. The Fisher linear discriminant is suited to identify the best guess. It is commonly used to maximize the spread of samples in pattern classification problems (Duda et al., 2001). Let $\hat{\sigma}_1$ and $\hat{\sigma}_2$ denote the standard de-

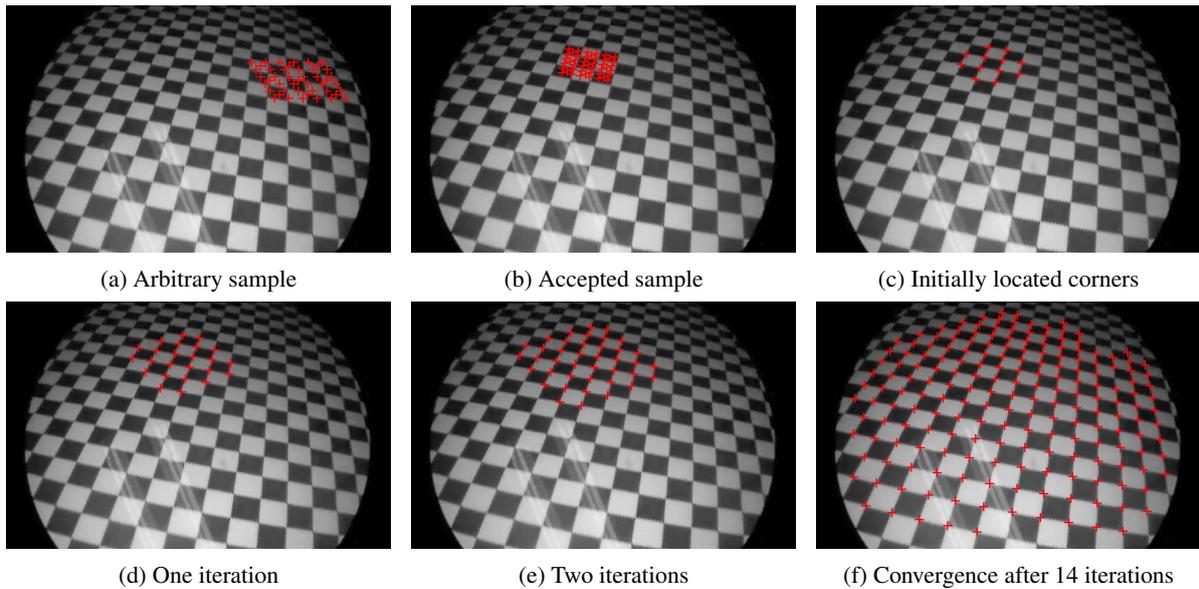
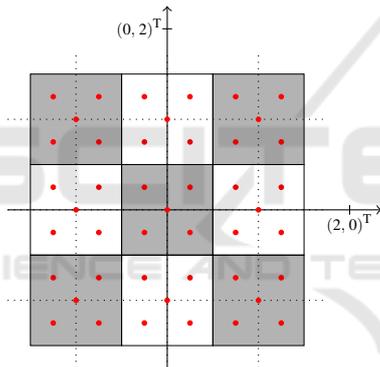


Figure 2: Individual steps of our method.


 Figure 3: Sparse sampling model. Red points indicate sampling coordinates. The grid size is 1 in x and y direction. Note that the model consists of two groups of homogeneous regions (black and white).

viation of all samples in the respective group. Let $\hat{\mu}_1$ and $\hat{\mu}_2$ denote the corresponding intensity means. We define the correspondence between the chessboard model and the image based on the idea of the Fisher discriminant as

$$s_{\mathbf{p}} := \frac{\hat{\mu}_1 - \hat{\mu}_2}{\hat{\sigma}_1 + \hat{\sigma}_2}, \quad (1)$$

where \mathbf{p} denotes the dependency on the model parameters, i.e. size, pose, and position. We test for different combinations of size, position, shearing and rotation to find a proper guess. Finally, we choose the sample that maximizes $|s_{\mathbf{p}}|$ and calculate an initial homography \mathbf{H}_0 based on correspondences between the model and image coordinates. We do not yet account for distortion and initialize the distortion \mathbf{d}_0 to

an identity mapping. We denote the complete procedure as sparse region sampling.

3.2 Pattern Growing

The initial estimate provides an approximation of \mathbf{H}_0 and \mathbf{d}_0 that is valid in a close vicinity to the initial guess. We use the approximation to predict corner locations of the sparse sampling model in the image. We drop the four outer corners of the sparse sampling model (see Figure 3), because their prediction is often less accurate and locate the remaining 12 corners with subpixel accuracy using the corner detector depicted in the following sections. Figure 2c shows the result of this step. The update of the projection parameters yields \mathbf{H}_1 and \mathbf{d}_1 . Then, we iteratively search nearby chessboard corners until we can't detect any new corner. Therefore, we consider the four direct neighbors of the chessboard corners that have already been detected and apply the corner detector to each of them. We append the new corners to the set of detected corners on success and drop them otherwise. After every iteration, we update \mathbf{H} and \mathbf{d} to improve the projection parameters for the next iteration.

Let us give an example: In the first iteration, we consider the neighbors of the 12 initial corners. Assume, we begin with $\mathbf{v}_i = (0.5, 1.5)^T$ of Figure 3. The four direct neighbors of \mathbf{v}_i are given by

$$n(\mathbf{v}_i) := \left\{ \begin{bmatrix} x_i \pm 1 \\ y_i \end{bmatrix}, \begin{bmatrix} x_i \\ y_i \pm 1 \end{bmatrix} \right\}. \quad (2)$$

We apply the corner detection only to the new points $(1.5, 1.5)^T$ and $(0.5, 2.5)^T$ and add them to the set of

detected corners on success. We apply this procedure to all 12 points and subsequently update the projection parameters to get \mathbf{H}_2 and \mathbf{d}_2 . The result of this iteration is shown in Figure 2d.

The pattern-growing phase relies on the region-based corner detector introduced in Section 3.2.3. It makes our method robust against noise and blur. First, we introduce a morphable model in Section 3.2.1 and its projection into the image space in Section 3.2.2 to explain the corner detector in detail.

3.2.1 Morphable Corner Model

The morphable model is defined by a linear combination of N deformed templates \mathbf{T}_k with coefficients α_k and an undeformed template \mathbf{T}_0 . The coefficients determine the shape of the model. Each template consists of M vertices¹ and is defined by

$$\mathbf{T}_k := \begin{bmatrix} x_{k1} & y_{k1} \\ \vdots & \vdots \\ x_{kM} & y_{kM} \end{bmatrix}, \quad \text{with } k \in [0, N]. \quad (3)$$

We associate one parameter α_k with each deformation target \mathbf{T}_k for $k > 0$ to specify the weight of each template in the linear combination. In addition, we introduce global translation and scaling parameters Δ_x , Δ_y , and s . The morph parameter vector combines all parameters in

$$\mathbf{p} := (\Delta_x, \Delta_y, s, \alpha_1, \dots, \alpha_N) \quad (4)$$

and parametrizes the morphable model m :

$$\begin{aligned} m: \mathbb{R}^{N+3} &\rightarrow \mathbb{R}^{M \times 2} \\ \mathbf{p} &\mapsto \mathbf{M} = \begin{bmatrix} \check{x}_1 & \check{y}_1 \\ \vdots & \vdots \\ \check{x}_M & \check{y}_M \end{bmatrix} \\ &:= s \left(\mathbf{T}_0 + \sum_{k=1}^N \alpha_k (\mathbf{T}_k - \mathbf{T}_0) \right) + \begin{bmatrix} \Delta_x & \Delta_y \\ \vdots & \vdots \\ \Delta_x & \Delta_y \end{bmatrix}. \end{aligned} \quad (5)$$

In this work, we use the elementary templates shown in Figure 4. Some of the illustrated shearing templates do not introduce additional deformations. However, they enable identical deformations by diverse parameter combinations and simplify alignment to the chessboard corners. Experiments confirmed a better convergence with these additional templates.

The morphable model is influenced by the point distribution model (pdm) proposed by Cootes and

¹We denote unmorphed coordinates in model space by (x, y) and morphed coordinates in model space by (\check{x}, \check{y}) . Further, we denote image coordinates by (u, v) , their homogeneous representations by $(\check{u}, \check{v}, \check{w})$ and distorted image coordinates by (\check{u}, \check{v}) .

Taylor (Cootes and Taylor, 1992). The deviation of each deformation template from the default template is similar to the statistical modes of variation in the pdm. However, we define the deformation templates manually and add scaling and translation in the model coordinate frame.

3.2.2 Projection Model

The morphable model resides in the model coordinate frame and \mathbf{M} is a matrix of model coordinates. We use planar calibration targets. Therefore, the image coordinates of the model lie in a two dimensional linear manifold of \mathbb{R}^3 . A homography defines the linear projection of each vertex (\check{x}, \check{y}) from \mathbf{M} into the image frame by

$$\begin{bmatrix} \check{u} \\ \check{v} \\ \check{w} \end{bmatrix} = \mathbf{H} \begin{bmatrix} \check{x} \\ \check{y} \\ 1 \end{bmatrix}. \quad (6)$$

The homography matrix $\mathbf{H} \in \mathbb{R}^{3 \times 3}$ is defined up to scale (Zhang, 2000). We get the inhomogeneous representation of the image coordinates by $u = \check{u}/\check{w}$ and $v = \check{v}/\check{w}$, respectively. Afterwards, we apply the Brown-Conrady model to account for non-linear radial and tangential distortion (Brown, 1971):

$$\begin{aligned} \check{u} &:= u(1 + k_1 r^2 + k_2 r^4) + (p_2(r^2 + 2u) + 2p_1 uv), \\ \check{v} &:= v(1 + k_1 r^2 + k_2 r^4) + (p_1(r^2 + 2v) + 2p_2 uv), \end{aligned}$$

where the coefficients k_1, k_2 cause radial distortion and p_1, p_2 cause tangential distortion. The radius r is the distance of each vertex to the distortion center (c_x, c_y) :

$$r^2 = (u - c_x)^2 + (v - c_y)^2.$$

The distortion parameter vector summarizes all coefficients in $\mathbf{d} := (c_x, c_y, k_1, k_2, p_1, p_2)$. We combine the linear and non-linear transform in

$$\begin{aligned} m': \mathbb{R}^{N+3} \times \mathbb{R}^{3 \times 3} \times \mathbb{R}^6 &\rightarrow \mathbb{R}^{M \times 2} \\ (\mathbf{p}, \mathbf{H}, \mathbf{d}) &\mapsto \mathbf{M}' = \begin{bmatrix} \check{u}_1 & \check{v}_1 \\ \vdots & \vdots \\ \check{u}_M & \check{v}_M \end{bmatrix}. \end{aligned} \quad (7)$$

The number of coefficients used for distortion modeling influences the accuracy of the model. Tsai shows that a radial distortion model with one coefficient is sufficient for industrial machine vision applications (Tsai, 1987). Barreto et al. show that more than a single radial distortion coefficient can be advantageous for endoscopy applications (Barreto et al., 2009). We focus on endoscopy images with strong distortion effects. Therefore, we use two radial and two tangential coefficients, similar to others (Zhang, 2000; Wei and Ma, 1994).

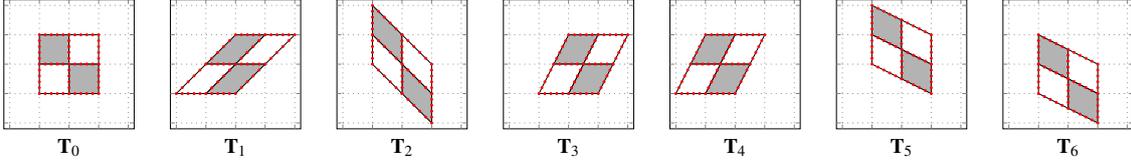


Figure 4: Different morph templates. \mathbf{T}_0 : unmorphed template; \mathbf{T}_1 and \mathbf{T}_2 : shearing at the model center; \mathbf{T}_3 - \mathbf{T}_6 : shearing by moving only the top, bottom, left and right side.

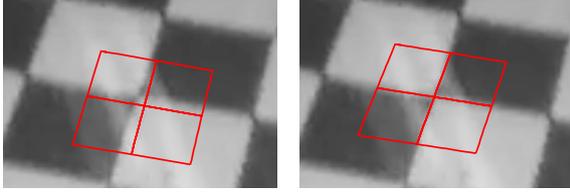


Figure 5: Subpixel precise corner detection using a 2×2 corner model. The corner model is shown before (left) and after (right) optimization.

3.2.3 Region-based Corner Detection

Our morphable corner detection model \mathbf{M}' consists of four distinct regions that form a 2×2 chessboard pattern (see Figure 5). The outline of each region is given by a sequence of vertices $(\check{u}_i, \check{v}_i)$, $i \in [k_r, l_r]$ with $l_r \geq k_r + 2$ and $k_{r+1} := l_r + 1$ for all $r \in [1, 4]$. Here, r denotes the region index and $k_r, l_r \in [1, M]$ are the boundaries of region r . We apply the 2×2 corner model to the domain of corner detection. Therefore, we need a criterion that measures how well the model separates the four chessboard regions in the image. In Section 3.1, we applied a variant of the Fisher discriminant to find the sample that separates the groups of black and white chessboard patches best. Based on the same idea, we derive a local measure that provides better guidance to the optimizer for subpixel precise corner detection. We use the Levenberg-Marquardt method to minimize the model residual. Therefore, we split the Fisher discriminant into two independent terms, where each term accounts for the variance within one region and the separation of that region from one of its neighbors. In contrast to the original formulation, we use signed values to provide better guidance to the optimizer. Finally, we turn the maximization into a minimization problem.

Let μ_r and σ_r be the pixel value mean and the standard deviation of all pixels in region r . We define the separation between two regions r_a and r_b by

$$\frac{\sigma_{r_a} + \sigma_{r_b}}{\mu_{r_a} - \mu_{r_b}} = \frac{\sigma_{r_a}}{\mu_{r_a} - \mu_{r_b}} + \frac{\sigma_{r_b}}{\mu_{r_a} - \mu_{r_b}}.$$

We assume that the four regions are indexed in clockwise order, such that

$$(r_a, r_b) \in \{(1, 2), (2, 3), (3, 4), (4, 1), (2, 1), (3, 2), (4, 3), (1, 4)\}$$

denote pairs of neighboring regions. Then, we get the separation of some region r_a to one of its neighbors r_b by

$$\hat{e}: [1, 4] \times [1, 4] \rightarrow \mathbb{R} \\ (r_a, r_b) \mapsto e_{r_a r_b} := \frac{\sigma_{r_a}}{\mu_{r_a} - \mu_{r_b}}. \quad (8)$$

Finally, we define the separation objective for \mathbf{M}' by

$$\mathbf{e} := (e_{12}, e_{21}, e_{23}, e_{32}, e_{34}, e_{43}, e_{41}, e_{14})^T. \quad (9)$$

For now, let us assume that we roughly know the parameters \mathbf{H} and \mathbf{d} for the projection of \mathbf{M} into the image space. Therefore, we can predict the approximate locations of the model coordinates (Δ_x, Δ_y) in the image. In the first iteration, \mathbf{H} and \mathbf{d} are initialized as described in Section 3.1. In subsequent iterations, these estimates are updated by taking newly detected corners into consideration. We describe the update procedure in Section 3.3.

We search for the exact image location $\mathbf{v}^* = (\check{u}_0^*, \check{v}_0^*)$ of the chessboard corner. This is done by minimizing \mathbf{e} with respect to the morph template parameters while keeping \mathbf{H} and \mathbf{d} constant. Therefore

$$\mathbf{p}^* = \arg \min_{\mathbf{p}} \|\mathbf{e}\|_2, \quad (10)$$

where \mathbf{p} is initialized with $(\Delta_x, \Delta_y, s, 0, \dots)$. We allow the model to be translated and sheared with six different shearing templates. This simplifies minimization because it lowers the risk that the error only decreases when several parameters are adjusted simultaneously. The solution, \mathbf{v}^* is then given by the center vertex of $m'(\mathbf{p}^*, \mathbf{H}, \mathbf{d})$ (see Figure 5).

Note that the scale parameter s of the model controls the size of the 2×2 search template relative to the chessboard size. We use $s = 0.5$, that is, each patch of the search template covers approximately a quarter of the area of the chessboard patch. If the corner detector fails with this configuration (possibly because the point is located close to the image border), we reduce s and try it again.

Applying the morph parameters in the model coordinate frame makes them scale independent. For example, adding 1 to Δ_x always moves the template by the width of one chessboard patch in the image frame, independent of the image resolution and patch size in the image. This allows fixed delta values to compute the numerical derivative of Equation (9).

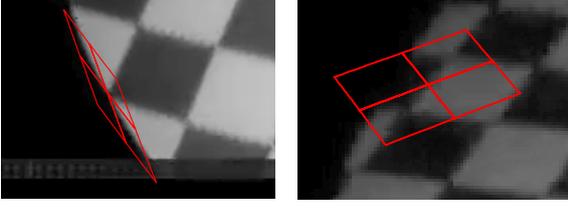


Figure 6: Examples of candidates that have been dropped due to a degenerate result (left) or a high residual (right).

3.2.4 Candidate Selection

In some cases, the corner detector provides invalid locations in the image as shown in Figure 6. Therefore, we introduce constraints that allow us to reject invalid results. First, the detected corner must be in the vicinity of the corner that is predicted by the current model state. This ensures that the optimization did not converge to a neighboring point. In practice, it is sufficient to allow a deviation of one half of the chessboard patch width. This can be easily verified with the translation parameters (Δ_x, Δ_y) of m . They must not differ more than 0.5 from their initialization. Second, the area of the corner detection model before the optimization must not differ too much from the area after optimization. This constraint prohibits degenerate configurations. Finally, the residual error $\|\mathbf{e}\|_2$ must meet the detection threshold. This rejects chessboard corners on the border or outside of the pattern.

3.2.5 Region Statistics

The calculation of μ_r and σ_r is done multiple times in every iteration of the optimization. Therefore, it requires a very fast algorithm. Ernst et al. have extended the integral image approach of Viola and Jones (Viola and Jones, 2001) to approximate statistics within polygonal image regions very efficiently (Ernst et al., 2013). We summarize the method in the following. The integral image \mathbf{I}_Σ of an image \mathbf{I} is given by

$$\mathbf{I}_\Sigma(x, y) := \sum_{i \leq x, j \leq y} \mathbf{I}(i, j). \quad (11)$$

The sum of pixel values s_r within some polygonal region r in an image of size $w \times h$ is approximated by

$$\hat{s}: \mathbb{R}^{M \times 2} \times \mathbb{R}^{w \times h} \rightarrow \mathbb{R}$$

$$(\mathbf{M}', \mathbf{I}_\Sigma) \mapsto s_r := \frac{1}{2} \sum_{i=k_r}^{l_r} [\mathbf{I}_\Sigma(\check{u}_i, \check{v}_{i+1}) - \mathbf{I}_\Sigma(\check{u}_{i+1}, \check{v}_i)], \quad (12)$$

with $l_r + 1 := k_r$. Similarly, \hat{a} provides the area a_r inside r :

$$\hat{a}: \mathbb{R}^{M \times 2} \rightarrow \mathbb{R}$$

$$\mathbf{M}' \mapsto a_r := \frac{1}{2} \sum_{i=k_r}^{l_r} (\check{u}_i \check{v}_{i+1} - \check{u}_{i+1} \check{v}_i). \quad (13)$$

Equations (12) and (13) allow the approximation of the mean pixel value μ_r within r by

$$\hat{\mu}: \mathbb{R}^{M \times 2} \times \mathbb{R}^{w \times h} \rightarrow \mathbb{R}$$

$$(\mathbf{M}', \mathbf{I}_\Sigma) \mapsto \mu_r := \frac{s_r}{a_r}. \quad (14)$$

Approximating the standard deviation σ_r within r requires a second integral image $[\mathbf{I}^2]_\Sigma$, where each pixel value is squared before the summation:

$$\hat{\sigma}: \mathbb{R}^{M \times 2} \times \mathbb{R}^{w \times h} \times \mathbb{R}^{w \times h} \rightarrow \mathbb{R}$$

$$(\mathbf{M}', \mathbf{I}_\Sigma, [\mathbf{I}^2]_\Sigma) \mapsto \sigma_r := \sqrt{\frac{\hat{s}(\mathbf{M}', [\mathbf{I}^2]_\Sigma)}{a_r} - \mu_r^2}. \quad (15)$$

The efficient approximation of μ_r and σ_r allows a fast implementation of the robust local corner detector depicted in Section 3.2.3. Subpixel precision is achieved by bilinear sampling. We refer to the previous work for a detailed explanation of the method (Ernst et al., 2013).

3.3 Parameter Optimization

The corner detector finds the image location of a chessboard corner \mathbf{v}^* that corresponds to a corner coordinate $\mathbf{v} = (\Delta_x, \Delta_y)$ in model space. With that relation, we define two sets

$$\mathcal{M}_t := \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{N_t}\}$$

and

$$\mathcal{I}_t := \{\mathbf{v}_1^*, \mathbf{v}_2^*, \dots, \mathbf{v}_{N_t}^*\}.$$

Here, \mathbf{v}_i denotes the model coordinate corresponding to \mathbf{v}_i^* and $t \in \mathbb{N}$ denotes the iteration number of the pattern-growing step.

We apply Equation (7) with $(\mathbf{H}_t, \mathbf{d}_t)$ on every $\mathbf{v}_i \in \mathcal{M}_t$ to predict the image locations \mathbf{v}'_i with the current projection model. The parameter optimization step aims at improving the projection model by minimizing the errors between the detected corners \mathbf{v}_i^* and the predicted corners \mathbf{v}'_i . Therefore, we find the model parameters for the next iteration by

$$(\mathbf{H}_{t+1}, \mathbf{d}_{t+1}) = \arg \min_{(\mathbf{H}_t, \mathbf{d}_t)} \left\| \begin{bmatrix} \mathbf{v}'_1 - \mathbf{v}_1^* \\ \vdots \\ \mathbf{v}'_{N_t} - \mathbf{v}_{N_t}^* \end{bmatrix} \right\|_2. \quad (16)$$

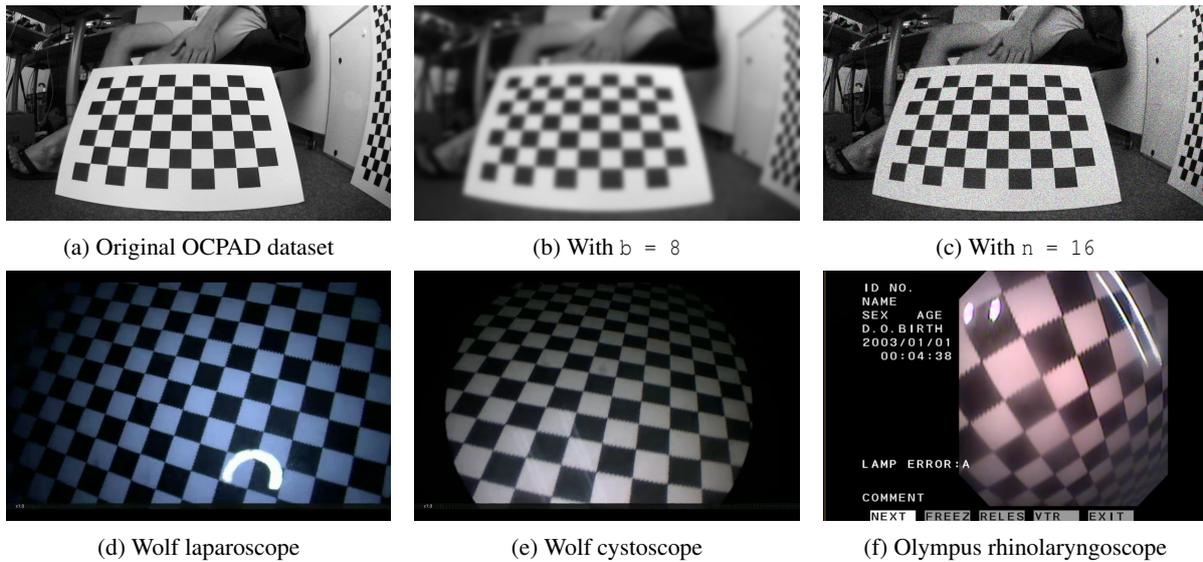


Figure 7: Example images of the datasets.

Algorithm 1: Chessdetect.

```

1:  $\mathcal{M}_0 \leftarrow \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{12}\}$ 
2:  $t \leftarrow 0$ 
3: Initialize  $\mathbf{H}_0$  and  $\mathbf{d}_0$  as described in Section 3.1
4: repeat
5:    $I_t \leftarrow \{\}$ 
6:   for all  $\mathbf{v} \in \mathcal{M}_t$  do
7:     Find  $\mathbf{p}^*$  using (10)
8:      $\mathbf{v}^* \leftarrow$  center vertex of  $\mathbf{M}_c^*$ 
9:     if constraints are met then
10:       $I_t \leftarrow I_t \cup \mathbf{v}^*$ 
11:     else
12:       $\mathcal{M}_t \leftarrow \mathcal{M}_t \setminus \mathbf{v}$ 
13:     end if
14:   end for
15:   Find  $\mathbf{H}_{t+1}$  and  $\mathbf{d}_{t+1}$  using (16)
16:    $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \cup \{n(\mathbf{v}) \mid \mathbf{v} \in \mathcal{M}_t\}$ 
17:    $t \leftarrow t + 1$ 
18: until  $|I_{t-1}| = |I_{t-2}|$ 

```

3.4 Summary

We have introduced all relevant parts of our method and summarize the procedure in Algorithm 1. To simplify notation, we skip the algorithm for the initial guess. Figure 3 shows our sparse sampling model and Figure 2c shows an example of the initial corners.

Line 1 initializes \mathcal{M}_0 with the corners of the sparse sampling model. Line 3 initializes \mathbf{H}_0 with the result of the initial guess and sets \mathbf{d}_0 to an identity mapping (see Section 3.1). The first iteration locates the corners \mathcal{M}_0 of the sparse sampling model with sub-pixel precision. The subsequent iterations search for

chessboard corners in the vicinity of already detected corners (see Figures 2d to 2f). The algorithm iterates until no new corners are detected.

Line 7 applies the region-based corner detector. Subsequently, line 8 extracts the center vertex of the corner model. Line 9 checks the constraints of Section 3.2.4. On success, the point is accepted in line 10. After all candidates in \mathcal{M} have been detected or dropped, the model parameters are estimated with the updated correspondences in line 15. Finally, line 16 generates new candidates for the next iteration.

In practice, it is sufficient to locate only the corners $\mathbf{v}_i \in \mathcal{M}$ that are new in the current iteration. In this case, it is recommended to locate all corners again after the algorithm terminated, using the projection parameters of the last iteration. Sometimes, the corner detector locates new corners in later iterations that have been dropped before. This is caused by more accurate projection parameters in later iterations that lead to a better initialization of the corner model in terms of translation and deformation.

4 EXPERIMENTS AND RESULTS

We evaluate our method with respect to image blur, image noise, and only partially visible patterns. In addition, we compare the performance with the Occluded Checkerboard Pattern Detector (OCPAD) by Fuersattel et al. (Fuersattel et al., 2016). To the best of our knowledge, this is the most recent approach that can also cope with partially visible patterns.

Table 1: Quantitative detection results.

set	total images	detected images		average points per image	
		Our method	OCPAD	Our method	OCPAD
Endoscope set 1	99	90	71	204.3	99.9
Endoscope set 2	97	88	47	171.3	108.8
Endoscope set 3	100	79	38	128.0	64.7
OCPAD original	64	64	63	53.9	53.7
blur = 2	64	64	64	54.0	54.0
blur = 4	64	64	61	53.3	53.6
blur = 8	64	13	0	39.1	0.0
noise = 4	64	64	58	53.9	53.3
noise = 8	64	64	30	53.9	51.4
noise = 16	64	59	0	51.8	0.0

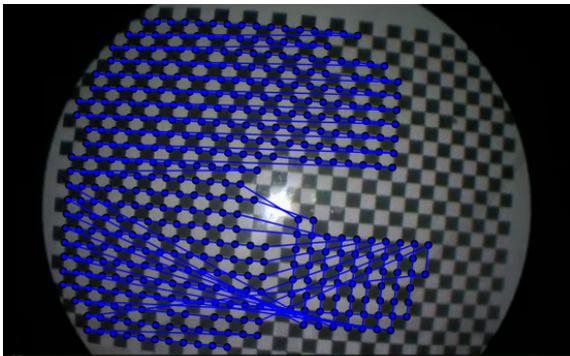


Figure 8: An image from endoscope set 1 that was discarded manually, because OCPAD delivers false correspondences.

4.1 Datasets

Fuersattel et al. created datasets that consist of fully visible as well as occluded chessboard patterns and made them publicly available. In a first step, we created variants of their fully visible checkerboard patterns dataset by artificially adding blur and noise to show how our method performs under these circumstances in a reproducible environment. We applied `mogrify`² on the dataset and used the following parameters:

- `mogrify -blur 0xb` blurs an image. Here, b denotes the standard deviation of the blur kernel. We use blur kernels with a standard deviation of 2, 4 and 8.
- `mogrify -attenuate n +noise gaussian` adds noise to an image. Here, n is the noise intensity. We use noise intensities of 4, 8 and 16.

To reduce evaluation times, we only use the first 64 images of the dataset. Otherwise, the `OpenCV calibrateCamera` function used for the evaluation

²<http://www.imagemagick.org/script/mogrify.php>

procedure in Section 4.2 takes very long. Examples of the datasets are shown in Figures 7a to 7c.

The good image quality of the OCPAD datasets and the low requirements on the chessboard detector impede differentiation of both methods on the original data. Although we raised the requirements by adding noise and blur, we know that the results on the artificially downgraded data do not necessarily generalize on real data. Therefore, we additionally compiled datasets that put high requirements on the chessboard detector using three different endoscopes:

- Set 1 was captured with a Panoview rigid laparoscope from Richard Wolf GmbH with 30° side view and a diameter of 10 millimeters.
- Set 2 was captured using a 0°, 4 millimeter Panoview rigid cystoscope from Richard Wolf GmbH.
- Set 3 was captured with a 3.9 millimeter video rhinolaryngoscope (Olympus ENF-VH). In contrast to the previous endoscopes, it has a flexible tube, where the image sensor is located at the distal end of the tube (chip-on-tip).

Figures 7d to 7f show example images of the datasets. Each endoscopy dataset consists of 100 images.

4.2 Procedure

We processed every image in the datasets with both chessboard detectors. Unfortunately, the OCPAD implementation³ requires to define the chessboard size in advance. It uses this knowledge to reject images, where only a small portion of the chessboard was detected. We do not have this prior knowledge for the endoscopy datasets, because the portion of the chessboard that is visible varies a lot within the sequences. Therefore, we choose the following strategy: First,

³<http://www.metrius.de/blog/portfolio-items/ocpad/>

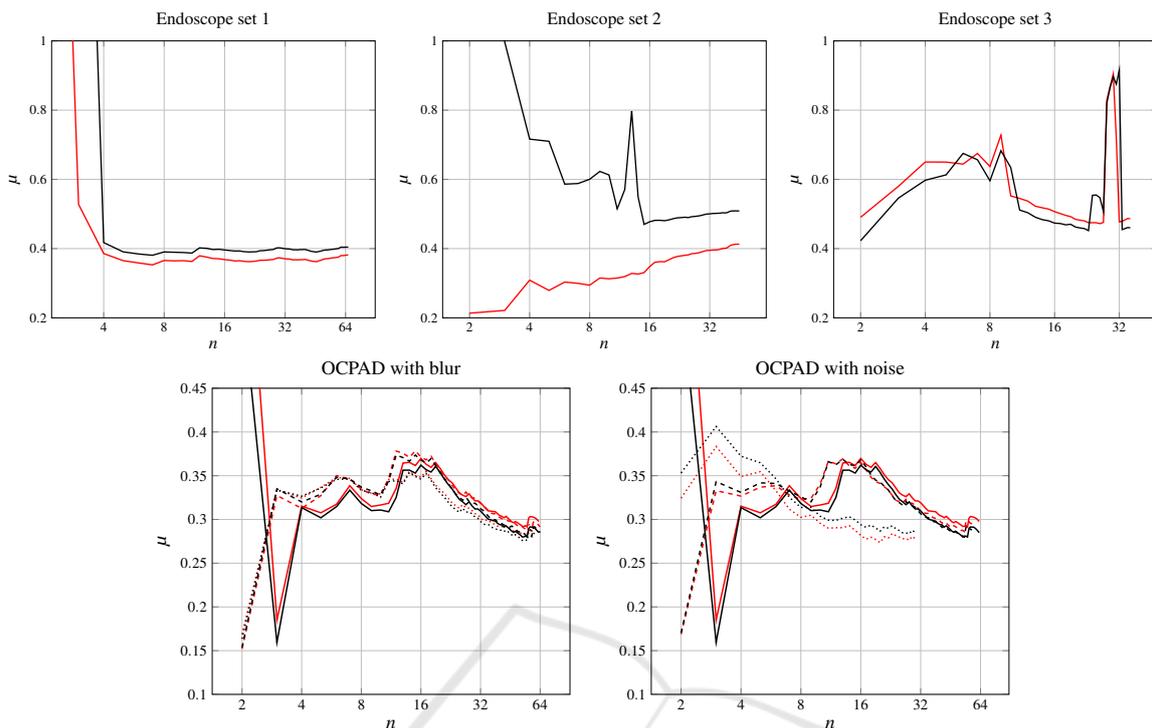


Figure 9: Average reprojection error μ of all images in the test sets. The camera parameters have been estimated with n images. The plots show the results of our region-based detector (red) and the OCPAD detector (black). In case of the datasets with artificial blur and noise, the solid lines correspond to the original data without blur or noise, whereas the dashed lines correspond to the sets with $b = 2$ or $n = 4$ and the dotted lines correspond to $b = 4$ or $n = 8$. Unfortunately, we have no results for $b = 8$ and $n = 16$, because OCPAD failed completely on these sets. Note, that the n -axis has a logarithmic scale and that the scale of the μ -axis varies between the endoscope and the OCPAD sets.

we apply OCPAD and set the chessboard size to 4×4 . If it succeeds, we increase the chessboard size by 1 in both dimensions and apply it again. We repeat this until OCPAD fails and keep the last valid result.

In rare cases, OCPAD fails with an exception or generates results that are obviously wrong (see Figure 8 for an example). In those cases, we simply remove the image from our datasets, such that it is completely ignored during the evaluation.

Based on the remaining results, we evaluate the detection rates and the number of point correspondences that have been found within each set. We only consider images, where the corresponding method has found at least 10 point correspondences. The detection rate is defined as the number of images (with at least 10 detected point correspondences) in relation to the total number of images. We also evaluate the average number of points that have been found in each image. Finally, we evaluate the calibration accuracy of both methods. To this end, we extract the set of points that have been detected by both methods and use them to estimate the camera parameters with the OpenCV library. The function `calibrateCamera` returns the final average reprojection error. The repro-

jection error does not only depend on the calibration and corner detection accuracy, but also on the quality of the camera model. However, the error imposed by the camera model exists in both methods.

4.3 Results

Table 1 summarizes the detection results of both methods. On the endoscope datasets, our method provides significantly higher detection rates and detects roughly twice as much point correspondences per image. On the artificially blurred images, both methods perform approximately equally well until the standard deviation reaches 8 pixels. In this case, the detection rates of both methods drop. On the noisy images, the detection rates of our method are approximately constant, whereas the detection rates of OCPAD decrease constantly.

Figure 9 shows the mean reprojection error for an increasing number of images on all datasets. We discuss the peculiarities of the results in more detail. Endoscope set 1 shows a high decrease of the reprojection error between 2 and 3 images with both methods. An insufficient number or inappropriate configuration

of correspondences in image 1 and 2 could explain this effect with a poor calibration. Note, that we used only the subset of correspondences that was detected by both methods for a fair evaluation of the reprojection error. The sudden increase of the error in endoscope set 3 around image 28 is probably caused by the image sequence, because the images that are added at that point are taken from a very different perspective.

Overall, our method provides significantly better detection rates on difficult endoscopy images as well as in presence of artificial noise and performs equally well in terms of accuracy on all datasets. Note, that the accuracy of our corner detector depends on the quality of the camera model. A more precise distortion model can lead to a more realistic deformation of the template in the image and a better alignment to the corner.

5 CONCLUSIONS

We introduced a new method that detects chessboard corners robustly and accurately even in presence of noise, blur and strong radial distortion. We showed that the region-based corner detector combined with the pattern-growing strategy detects significantly more chessboard corners than another recent approach in difficult images and performs equally well in terms of accuracy. Our method is well suited when the calibration pattern is only partially visible or when the image quality is low. Therefore, it is particularly qualified for endoscope calibration. The method can be implemented efficiently using an extended variant of the integral image to calculate region means and variances. Due to its efficiency and accuracy, it is well suited for clinical environments, although it is not limited to that application.

REFERENCES

- Barreto, J., Roquette, J., Sturm, P., and Fonseca, F. (2009). Automatic camera calibration applied to medical endoscopy. In *Proceedings of the British Machine Vision Conference*.
- Brown, D. C. (1971). Close-range camera calibration. *Photogrammetric Engineering*, 37.
- Cootes, T. F. and Taylor, C. J. (1992). Active shape models - 'smart snakes'. In *Proceedings of the British Machine Vision Conference*.
- Duda, R. ., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley, 2nd edition.
- Ernst, A., Papst, A., Ruf, T., and Garbas, J.-U. (2013). Check My Chart: A Robust Color Chart Tracker for Colorimetric Camera Calibration. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*.
- Fiala, M. and Shu, C. (2005). Fully automatic camera calibration using self-identifying calibration targets. Technical report, National Research Council Canada.
- Fuersattel, P., Dotenco, S., Placht, S., Balda, M., Maier, A., and Riess, C. (2016). OCPAD – Occluded Checkerboard Pattern Detector. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*.
- Heikkila, J. and Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.
- Li, W., Nie, S., Soto-Thompson, M., Chen, C.-I., and A-Rahim, Y. I. (2008). Robust distortion correction of endoscope. In *Proceedings of the International Society for Optical Engineering*.
- Mallon, J. and Whelan, P. F. (2007). Which pattern? Biasing aspects of planar calibration patterns and detection methods. *Pattern recognition letters*, 28(8).
- Rufli, M., Scaramuzza, D., and Siegwart, R. (2008). Automatic detection of checkerboards on blurred and distorted images. In *Proceedings of the International Conference on Intelligent Robots and Systems*.
- Stehle, T., Truhn, D., Aach, T., Trautwein, C., and Tischendorf, J. (2007). Camera calibration for fish-eye lenses in endoscopy with an application to 3d reconstruction. In *Proceedings of the IEEE International Symposium on Biomedical Imaging*.
- Sun, W., Yang, X., Xiao, S., and Hu, W. (2008). Robust Recognition of Checkerboard Pattern for Deformable Surface Matching in Multiple Views. In *Proceedings of the High Performance Computing & Simulation Conference*.
- Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4).
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.
- Wei, G.-Q. and Ma, S. D. (1994). Implicit and explicit camera calibration: theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5).
- Wengert, C., Reeff, M., Cattin, P. C., and Székely, G. (2006). Fully automatic endoscope calibration for intraoperative use. In *Proceedings of Bildverarbeitung für die Medizin*.
- Zhang, C., Helferty, J., McLennan, G., and Higgins, W. (2000). Nonlinear distortion correction in endoscopic video images. In *Proceedings of the International Conference on Image Processing*.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11).