Deltille Grids for Geometric Camera Calibration

Hyowon Ha[†,*] Michal Perdoch[‡,*] Hatem Alismail[‡] In So Kweon[†] Yaser Sheikh[‡]
[†]Korea Advanced Institute of Science and Technology [‡]Oculus Research

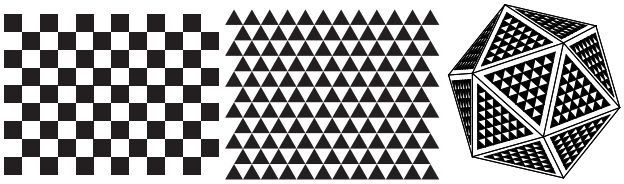
Abstract

The recent proliferation of high resolution cameras presents an opportunity to achieve unprecedented levels of precision in visual 3D reconstruction. Yet the camera calibration pipeline, developed decades ago using checkerboards, has remained the de facto standard. In this paper, we ask the question: are checkerboards the optimal pattern for high precision calibration? We empirically demonstrate that deltille grids (regular triangular tiling) produce the highest precision calibration of the possible tilings of Euclidean plane. We posit that they should be the new standard for high-precision calibration and present a complete ecosystem for calibration using deltille grids including: (1) a highly precise corner detection algorithm based on polynomial surface fitting; (2) an indexing scheme based on polarities extracted from the fitted surfaces; and (3) a 2D coding system for deltille grids, which we refer to as DelTags, in lieu of conventional matrix barcodes. We demonstrate state-of-the-art performance and apply the full calibration ecosystem through the use of 3D calibration objects for multiview camera calibration.

1. Introduction

Camera calibration is a foundational operation in 3D vision systems. Small errors incurred during the calibration of perspective cameras are magnified by back-projection and these errors often cascade through all successive modules of the system. Precision is, therefore, of paramount importance. Checkerboard patterns have become the default choice for high precision camera calibration, as they provide a uniform calibration pattern with distinct corners for precise fiducial localization [5, 6]. An entire ecosystem has been built around checkerboard patterns that includes efficient and precise corner detection algorithms [6, 18, 24], robust indexing schemes when only a partial checkerboard is visible [14, 30], and matrix barcodes for absolute localization of corners on the checkerboard [13, 27]. Yet, the advent of higher resolution cameras and the proliferation of multi-camera rigs calls for revisiting the merit of checkerboard patterns as the basis of the calibration pipeline.

[*]Equal contribution



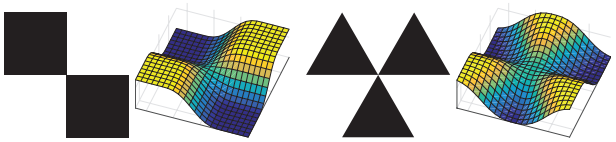
(a) Checkerboard (b) Deltille Grid (c) Icosahedron

Figure 1: Regular tiling with uniform but alternating colorings. (a) Checkerboard pattern, (b) a Deltille pattern with the same edge length as (a), (c) a 3D calibration object textured with a deltille grid.

In this paper, we advocate for the use of triangles rather than quadrilaterals as primitives for calibration patterns, and report significant advantages to their use. We present a complete calibration ecosystem based on deltille patterns [8] including an algorithm for high precision deltille corner detection, an indexing algorithm for partially viewed deltille grids, and a triangular barcode system, which we refer to as DelTags, for absolute localization of calibration objects. This ecosystem is applied to precise multi-view camera calibration using a 3D calibration object.

A *deltille* is a regular triangular tiling (so-called "delta tiles") with a Schläfli symbol of $\{3, 6\}$ [9]. It is one of only three regular tilings of the Euclidean plane and, besides the familiar checkerboard tiling, it is the only tiling that can be colored with alternating colors (specifically, with a uniform coloring of 121212) [16]. We refer to a deltille pattern with such a coloring as a *deltille grid* (see Fig. 1(b)).

There are three principal advantages to deltille grids compared to checkerboards: (1) Deltille grids achieve a higher density of elements, for a given edge distance, compared to square or skew-rectangular tiling, as can be seen in Fig. 1. This property allows for a greater number of fiducials to be detected, improving localization precision (as shown in our experiments). Furthermore, as it is an isotropic tiling, deltille grids help reduce localization bias when the pattern is observed under perspective transformation. (2) Each fiducial is characterized by the intersection of three alternating edges, which offer a greater number of constraints compared to the intersection of two edges for checkerboards (see Fig. 2). This property results in a more precise localization even in the fronto-parallel view, under



(a) Saddle surface (b) Monkey saddle surface

Figure 2: Two types of checkerboard corners and corresponding local surface shapes

blur, and intensity noise in the image. (3) As a texture primitive for 3D calibration objects, a triangle is a more attractive primitive than a quadrilateral. It is known that triangle meshes are always more accurate than quadrilateral meshes in approximating convex structures [11], which are desirable for multiview calibration. Consider, for example, a cube compared to icosahedron as a calibration object: the greater the number of regular faces, the less the bias due to viewpoint. Intuitively, triangles are simpler primitives to handle compared to quads: a triangle is flat and convex by construction and the quality of a triangle mesh is straightforward to characterize (in terms of the deviation from an equilateral triangle) [4].

It is timely to explore new approaches to high precision calibration, as high-resolution imagery is increasingly being captured from multiple views in environments, such as sports arenas, television studios, and crowd captures of events. The advantages of deltille grids allow us to access higher levels of precision in calibration, that can then filter down to the rest of the vision process such as special effects, 3D distance estimation, and 3D reconstruction.

Contributions. We present a calibration pipeline based on deltille grids with the following specific contributions:

1. **Deltille Corner Detection**: We present an approach for highly precise deltille corner detection based on fitting a monkey saddle point surface directly to the image intensity samples. The polynomial surface fit is convex and we employ a result from differential geometry to design a fiducial discriminant as a star-shaped umbilical point.
2. **Indexing for Partially-viewed Deltille Grids**: An auxiliary result of the polynomial fit is access to the polarities of the fit. Using the polarities at each point, we design a robust indexing method for partial deltille grid detection.
3. **DelTags**: We present a 2D coding system designed to maximize the visibility of the code within a triangular tile (or delta), that preserves the main error correction mechanisms of matrix barcodes (such as QR codes).

We apply this pipeline to multiview calibration using a 3D calibration object with a deltille grid. The corner detection and indexing scheme also apply to checkerboard patterns, and we qualitatively compare both modules and demonstrate state-of-the-art performance on standard datasets.

2. Prior Works

Checkerboard corner detection can be split into two tasks: checkerboard pattern detection and corner refinement. Typically, a combination of different methods for each part is encouraged to simultaneously achieve the best detection rate and corner accuracy.

Checkerboard Pattern Detection. The OpenCV checkerboard detection algorithm [6] is a widely used full-checkerboard detector. It detects corners by segmenting black quadrangles and finding intersections of their edges.Wang et al. [31] proposed a method to fit two groups of lines to find the pattern grid. The lines of the grid can also be found using the Hough transform [12, 17]. However, this kind of line fitting based detection methods is only feasible on images with small lens distortions. A version of the algorithm robust to strong lens distortion was proposed by Rufli et al. [29] and implemented in the OCamCalib Toolbox [30]. Placht et al. [28] presented a full checkerboard detector which looks for a graph structure of the checkerboard in a skeletal binary edge image. Fuersattel et al. [14] extend this method using a subgraph matching scheme to achieve partial checkerboard detection. However, it is sometimes difficult to obtain a binary edge image preserving a clear checkerboard structure if the image is blurry or unevenly lit. Geiger et al. [15] proposed a method which iteratively expands checkerboard hypotheses starting from seed points. Although many of these methods provide automatic detection of checkerboard, they often fail due to various factors such as image resolution, blur, noise, uneven illumination or strong geometric distortion. For this reason, some people prefer to use Bouguet's camera calibration toolbox for MATLAB [5], which allows the user to manually extract the checkerboard, to avoid false detections that can ruin the entire calibration result.

Corner Refinement. The Harris corner detector [18] is a common choice for localizing corner points. Bouguet [5] present a method which refines initial corners to better locations where the gradients in the vicinity are orthogonal to the selfward vectors. This well-known method is also adopted in OpenCV [6] and Geiger et al. [15]. Lucchese and Mitra [24] refine the initial Harris points by computing saddle points of polynomial surfaces fitted to the vicinity of the corners. The image is smoothed with a Gaussian filter before fitting polynomials. Chen et al. [7] propose a comparable saddle point extraction directly using a Hessian matrix, instead of fitting polynomials, for computational efficiency. Recently, Placht et al. [28] show that accuracy can be improved by using a cone-shaped filter instead of a Gaussian. Their saddle point method shows more robust characteristics against blur and intensity noise than other methods with the same amount of information *i.e*. small patch, which allows the use of a target with a higher density of features.

Other Primitives. Circular features or dots were also used as primitives for calibration targets [19, 20, 21]. It was shown in [25] that a compensated conic fitting can be used to remove localization bias of their centroid under projective transformation but the bias from nonlinear transformation like lens distortion still affects the precise localization. Methods based on concentric circles [22] and confocal conics [23] are also prone to nonlinear lens distortions. On the contrary, polynomial fitting based corner detection [24] – similar to our method – was proven to be invariant to both types of bias [25].

3. Method: Deltille Grid Detection

In the following sections we present the corner detection on a deltille pattern, a generalization of the precise localization of corners on rectangular grids [28, 2]. To highlight the analogies, we detail the rectangular case[1] and will refer to it throughout this section. In Sec. 3.2, we detail a robust method for *indexing* i.e. establishing 2D to 3D correspondences on partial targets and multifaceted calibration objects.

3.1. Corner Detection by Monkey Saddle Fitting

Let us represent an image of a deltille grid as a smooth two dimensional function of pixel coordinates with values in each point representing an intensity of a pixel. The intersection of the edges on a deltille grid forms a specific type of saddle point of the image function, see Fig. 2(b), usually called a *monkey saddle* [1]. Geometrically, there are three valleys and three ridges of the image intensity function that form a perfect saddle for the two legs and a tail of a monkey. The minimum degree of the polynomial that forms a monkey saddle surface is three, and the deltille grid corner is a critical point of this surface.

Let $f(\mathbf{x})$ be a smooth image intensity patch in the vicinity sought deltille grid corner $\mathbf{x}^*$, where alternating black and white regions meet at the intersection point of three triangles. Given an initial guess of the critical point location $\mathbf{x}^0$, we construct a sequence $\mathbf{x}^t$ by approximating the local surface by a series of third order polynomials centered at $\mathbf{x}^t$, whose critical points converge to $\mathbf{x}^*$. Instead of using Taylor series expansion that would involve up to third order derivatives in $\mathbf{x}^t$, we directly fit a third order polynomial

$$\tilde{f}(\mathbf{x};\mathbf{c}) = c_1x^3 + c_2x^2y + c_3xy^2 + c_4y^3 + c_5x^2 + c_6xy + c_7y^2 + c_8x + c_9y + c_{10}, \quad (1)$$

to the intensity surface function f around point $\mathbf{x}^t$. The polynomial surface fitting can be formulated as

$$\arg\min_{\mathbf{c}} \sum_{\mathbf{\Delta}_i \in \mathcal{N}} \|f(\mathbf{x}^t + \mathbf{\Delta}_i) - \tilde{f}(\mathbf{\Delta}_i;\mathbf{c})\|_2 \quad (2)$$

[1]Please refer to the extended version of the paper.

where $\mathbf{c} = [c_1, c_2, \ldots, c_{10}]^\top$ is the vector of polynomial coefficients and $\mathcal{N}$ is a set of n sampling points $\mathbf{\Delta}_i = (u_i, v_i)$ that are chosen on a grid within the distance r from $\mathbf{x}^t$.

In practice, this least squares fitting problem can be efficiently solved for m corner points at once. The vector of coefficients $\mathbf{c}$ satisfies a linear system $\mathbf{Ac} - \mathbf{b} = \mathbf{0}$, where

$$\mathbf{A} = \begin{bmatrix} u_1^3 & u_1^2v_1 & u_1v_1^2 & v_1^3 & u_1^2 & u_1v_1 & v_1^2 & u_1 & v_1 & 1 \\ \vdots & \vdots & \vdots & & & & & & & \\ u_n^3 & u_n^2v_n & u_nv_n^2 & v_n^3 & u_n^2 & u_nv_n & v_n^2 & u_n & v_n & 1 \end{bmatrix}$$

$$\mathbf{b} = \left[f(\mathbf{x}^t + \mathbf{\Delta}_1), \cdots, f(\mathbf{x}^t + \mathbf{\Delta}_n)\right]^\top.$$

The sampling of subpixel intensities of f is done using bilinear interpolation.

By selecting the same grid of $\mathbf{\Delta}_i$ for each corner, the matrix $\mathbf{A}$ can be fixed and reused in fitting of all points at each iteration. Thus, we solve all surface fitting problems at each iteration by a single equation $\mathbf{AC} = \mathbf{B}$, where the n-by-m matrix $\mathbf{B}$ can be filled by concatenating the corresponding vectors of samples for all m corner points. To solve this over-determined problem (for $n \geq 10$), we compute $\mathbf{M} = \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top$, to estimate 10-by-$m$ matrix $\mathbf{C}$ of all polynomial surface coefficients by $\mathbf{C} = \mathbf{MB}$. Note that $\mathbf{M}$ needs to be computed only once in the beginning and utilized over the entire process.

The localization of the critical point of the fitted surface based on the conventional Newton's method turns out to be tricky, as the first derivatives of the surface are second-order polynomials:

$$\nabla_x\tilde{f}(\mathbf{x}) = 3c_1x^2 + 2c_2xy + c_3y^2 + 2c_5x + c_6y + c_8,$$
$$\nabla_y\tilde{f}(\mathbf{x}) = c_2x^2 + 2c_3xy + 3c_4y^2 + c_6x + 2c_7y + c_9.$$

Thus, unlike in the rectangular corner case, there might exist multiple stationary points (intersections of two ellipses) around the deltille corner point. Newton method is less likely to reach the true monkey saddle point and converges to some of these local stationary points.

To overcome this problem, we use another clue found in differential geometry. The critical point of a monkey saddle surface is regarded as an umbilical point, at which the Gaussian curvature becomes zero [3]. Consequently, the monkey saddle point is a degenerate critical point where all of its second derivatives are zero. This allows us to discriminate it by keeping only points $\mathbf{x}^*$ where $\nabla^2\tilde{f}(\mathbf{x}^*) = \mathbf{0}$.

In each iteration, we compute the location of the degenerate critical point $\mathbf{x}_d$ from the local surface fitted in $\mathbf{x}^t$, satisfying:

$$\nabla^2\tilde{f}(\mathbf{x}_d)\begin{cases} \nabla^2_{xx}\tilde{f}(\mathbf{x}_d) = 6c_1x_d + 2c_2y_d + 2c_5 = 0, \\ \nabla^2_{xy}\tilde{f}(\mathbf{x}_d) = 2c_2x_d + 2c_3y_d + c_6 = 0, \\ \nabla^2_{yy}\tilde{f}(\mathbf{x}_d) = 2c_3x_d + 6c_4y_d + 2c_7 = 0. \end{cases} \quad (3)$$

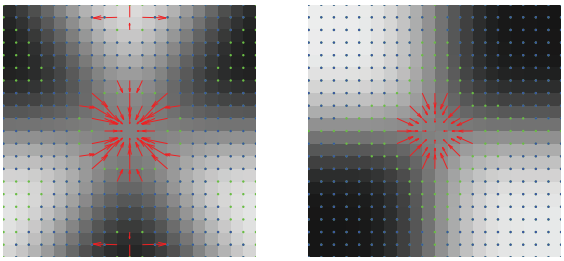


Figure 3: An iteration of the procedure for: deltille (left) and rectangular corner (right). From an initial grid of points $\mathbf{x}^0$, the red arrow is shown to $\mathbf{x}^1$ if point satisfies Eq. 5, and its displacement $\mathbf{x}_d < r$. Blue points are filtered by $J \geq 0$, green points are filtered out when $\mathbf{x}_d \geq r$.

Since Eq 3 is an over-determined linear system, $\mathbf{x}_d$ is computed by solving the corresponding least squares problem. The monkey saddle point location $\mathbf{x}^{t+1}$ is updated as:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \mathbf{x}_d \tag{4}$$

Any point in the vicinity of a monkey saddle should have both positive and negative principal curvatures [3], therefore to detect all corner points on a deltille grid, the iteration is initialized densely from every point that satisfies $\det \nabla^2 f(x) < 0$. To filter out non-monkey-saddle points that share the same critical point property (Eq. 3), we explore previous studies on the umbilical classification which was first proposed by Darboux [10]. Berry and Hannay [3] named the three typical forms of umbilical points: star, monstar, and lemon, based on the pattern of the principal direction vector field around the umbilic. Specifically, in our case, the star shape of umbilical point represents the monkey saddle point. The discriminant for star shaped umbilics distinguishing them from others is given by:

$$J(c_1, c_2, c_3, c_4) = 3(c_1 c_3 + c_2 c_4) - (c_2^2 + c_3^2) < 0, \tag{5}$$

and we consider any local surface satisfying $J < 0$, is potentially a monkey saddle point. The convergence flow around a sample deltille and rectangular corner is depicted in Fig. 3.

Since the discriminant is only based on the highest order coefficients, it is invariant to x-y domain shift of surface fitting. In other words, the discriminant can be checked for all points from the first iteration of our process regardless of how precisely the points are localized. This is important to significantly reduce outliers and allows us to efficiently process only monkey saddle like points.

3.2. Robust Indexing for Deltille Grids

The appearance of the calibration pattern in images changes significantly due to image acquisition effects such as vignetting, lens aberrations and rasterization on the sensor as well as geometric effects such as strong radial and perspective distortion. Calibration with multifaceted calibration objects *e.g.* Fig. 1(c) or partially visible patterns requires robust indexing – process of establishing correspondences between the image corners and pattern points. In the following, we detail individual steps of the indexing procedure as shown in Fig. 4.

3.2.1 Polarities

Let us assume that each converged point $\mathbf{x}^*$ from a previous iterative procedure is a precisely localized monkey saddle point. The polynomial surface $\tilde{f}(\mathbf{x}^*)$ can be approximated as a multiplication of three hyperplanes (intensity ramps) intersecting at the origin:

$$\tilde{f}(\mathbf{x}) \approx k\,(x\sin\theta_1 - y\cos\theta_1)\,(x\sin\theta_2 - y\cos\theta_2)\,(x\sin\theta_3 - y\cos\theta_3) + l. \tag{6}$$

The angles θ_i of these hyperplanes, which we denote as *polarities*, describe the orientation and shape of the surface (see Fig. 4(a)) that can be further utilized in grid indexing. In particular, by relating Eq. 6 with Eq. 1, we obtain

$$\begin{aligned}
c_1 &= k\sin\theta_1\sin\theta_2\sin\theta_3,\\
c_2 &= -k\sin\theta_1\sin\theta_2\cos\theta_3 - k\sin\theta_1\cos\theta_2\sin\theta_3 - k\cos\theta_1\sin\theta_2\sin\theta_3,\\
c_3 &= k\sin\theta_1\cos\theta_2\cos\theta_3 + k\cos\theta_1\sin\theta_2\cos\theta_3 + k\cos\theta_1\cos\theta_2\sin\theta_3,\\
c_4 &= -k\cos\theta_1\cos\theta_2\cos\theta_3.
\end{aligned}$$

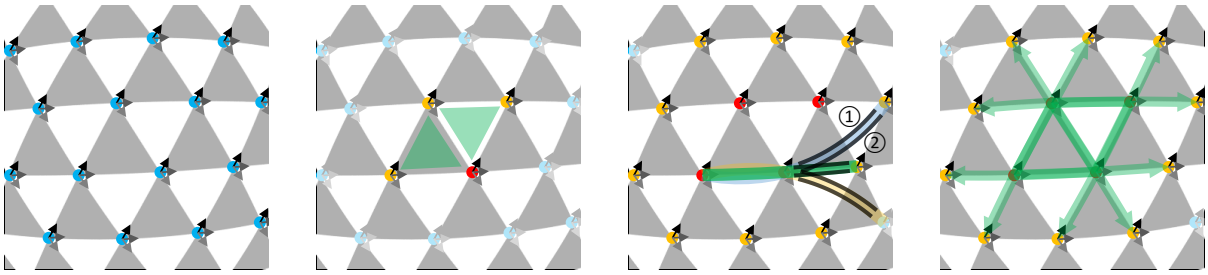
Three angles θ_1, θ_2, and θ_3 are obtained as the three solutions of the following third order equation:

$$c_4\tan^3\theta + c_3\tan^2\theta + c_2\tan\theta + c_1 = 0. \tag{7}$$

Two of its three roots may be complex numbers. We embed roots into a unit sphere using real part as azimuth and imaginary part as elevation, and then use the cosine distance to find similar neighboring corners on the deltille grid.

3.2.2 Initial Quad Selection

The initial quad consists of two triangle elements of the deltille grid (Fig. 4(b)). The first point of a quad is randomly selected, then out of all nearest neighbors, a diagonal neighbor with the same polarity is selected. Each nearest neighbor shadows an angular span of γ degrees and as a result, only a couple of closest neighbors in the vicinity of the point survive. Out of them, the third and fourth point in the quad that has the same polarities are selected and interior of the triangles inspected for homogeneity. Polarities are deemed the same if they are within δ degrees. An initial quad is formed for the next stage if after this procedure exactly two off-diagonal points remain out of the neighbors.



(a) Corners with polarities (b) Initial quad detection (c) Circle fitting (d) Grid growing

Figure 4: Illustration of deltille grid detection indexing: (a) detected corner points (blue dot) with polarities (black and gray arrows), (b) initial quad with initial point (red dot) and selected neighboring points (orange dots), (c) edge growing by fitting circular arcs (blue, green, yellow) into considered neighboring points (orange dots) and checking homogeneity along the arc, dark regions. (d) grid growing after the first iteration.

3.2.3 Edge Growing by Circle Fitting

In the last stage, the deltille grid is grown from each initial quad. From each edge already in the grid, candidates extending the edge are sought. First, the same procedure as in the previous step is used to reduce the number of nearest neighbors by shadowing. Then, a candidate neighbor with a correct polarity is selected and a circular arc is fitted into the triplet of two edge points and the candidate (see Fig. 4(c)). Homogeneity along both sides of the considered edge, marked by (1) and (2), is inspected by comparing corresponding intensity values on both sides. If consecutive samples on each side of the edge are homogeneous, and the intensity gradient between samples on opposite sides does not change sign, the candidate is inserted into the grid together with the extending edge (Fig. 4(d)). The procedure continues in this breadth-first search until there are no candidates that can be added.

3.3. DelTags

In the previous sections, we detailed a robust approach to establish neighborhood structure of partial rectangular and deltille grids. However, more than a local neighborhood structure of the grid needs to be inferred in order to define unique correspondences between 2D observations and 3D locations of points on the calibration object with multiple faces, or when only a part of the grid is visible.

Inspired by the recent success of various families of QR tags for localization [27] in robotics and AR applications [13], we introduce a novel class of QR codes *DelTags* that can be easily embedded in deltille grids. DelTags from a family dbhh (see Fig. 5) are triangular embeddings of binary codes uniquely encoding position and orientation similarly to AprilTags [27] with the following properties:

1. **Distinguishable**, to keep low false positive rates, each DelTag of length b is guaranteed to be in Hamming distance of at least h from any other DelTag in the family.
2. **Robust to rotations**, each rotation of a DelTag is in at least in Hamming distance h from all other codes and their rotations.
3. **Robust to bit-flips**, having all other DelTags of the same family in at least h Hamming distance provides error correction of up to $\lfloor (h-1)/2 \rfloor$ bit errors and detection of $\lfloor h/2 \rfloor$ errors.

(a) (b) (c)

Figure 5: Example of DelTags family d25h7. (a) Indexes of bits embedded in a DelTag, (b) embedding of tag with value 4945677, black triangle encodes bit with value 0, white triangle with value 1, (c) deltille board with DelTags from family d25h7.

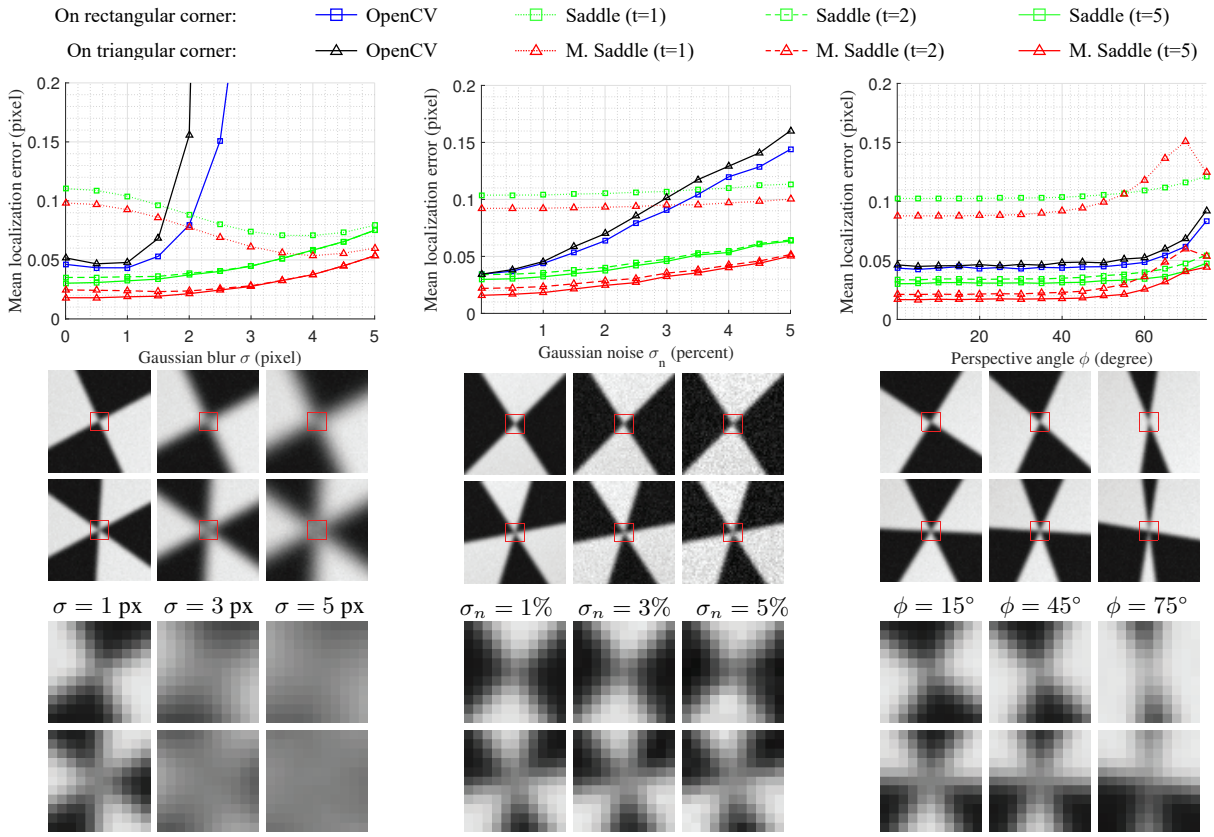
4. Experimental Results

We have evaluated several aspects of our method in the experimental validation, and in camera calibration using the standard pinhole camera model [32]. First, we have focused on the precision of our deltille and rectangular corner extraction algorithms based on realistic simulation in Sec. 4.1. For benchmarking our deltille grid indexing algorithm against state-of-the-art methods, we applied it both to conventional checkerboard and deltille grid detection and compared their performance in Sec. 4.2. The precision in real world calibration scenarios was compared in Sec. 4.3. Finally, we demonstrate that the use of deltille grids allows simpler calibration of a multi-view camera rig in Sec. 4.4.

4.1. Simulation

This experiment was focused on the validation of the precision and convergence of our polynomial fitting based corner methods. As a reference method, we used the OpenCV's subpixel localization implementation [6].

Image Rendering Pipeline. An image rendering pipeline was designed, to closely resemble the appearance of the rectangular and monkey saddle points in real world images.



(a) Gaussian blur test (b) Gaussian noise test (c) Perspective distortion test

Figure 6: Analysis of the influence of blur, noise and perspective distortion on the localization precision and convergence of our method. For details, please refer to Sec. 4.1.

First, a point on the view sphere centered on a corner is generated as an in-plane rotation α and zenith[2] ϕ angles. Then, a sharp image of a perspective camera at (α, ϕ) looking at rectangular/triangular corner is generated. To simulate defocus blur, the image is filtered with an isotropic Gaussian kernel with zero mean and standard deviation σ pixels. Finally, a Gaussian intensity noise with zero mean and standard deviation of σ_n intensities is then added to each pixel.

Evaluation Protocol. The influence of individual aspects – blur (σ), noise (σ_n) and viewpoint (ϕ) – of image rendering pipeline on the localization precision and convergence of our method was evaluated in three tests: 1) *blur test* $\sigma \in \langle 0, 5 \rangle$ *px* was with fixed $\sigma_n = 1.0\%$ and random $\phi \in \langle 0, 60 \rangle°$, 2) *noise test* $\sigma_n \in \langle 0, 5 \rangle\%$ was with fixed $\sigma = 1.0$ *px*, random $\phi \in \langle 0, 60 \rangle°$, and 3) *perspective distortion* test $\phi \in \langle 0, 75 \rangle°$ was with fixed $\sigma = 1.0$ *px* and $\sigma_n = 1.0\%$.

For each data point, a set of 1000 of 200×200 images was generated with the in-plane rotation α randomly sampled from $\langle 0, 2\pi \rangle$. The ground-truth location of $\mathbf{x}_g^*$ was randomly generated within a 1-pixel square of the center pixel, to avoid bias due to rendering artifacts such as aliasing. Examples of generated images are shown in Fig. 6. All tested methods were initialized from four corners of the pixel $\mathbf{x}_g^*$ and a fixed 11×11 window size was used for all methods. The localization error $|\mathbf{x}^* - \mathbf{x}_g^*|$ of OpenCV method was measured after the convergence, and that of ours after first, second and fifth iteration for a meaningful convergence analysis. Finally, localization errors from all 1000 images ($\times$ 4 corners) were averaged.

Gaussian Blur Test. The results of the image blur test are shown in Fig. 6(a). The reference OpenCV method diverges from the correct location for $\sigma > 2$ on both rectangular and deltille corner points. After only two iterations, both our regular and monkey saddle methods outperform the reference method over the full range of blurs. The monkey saddle point extraction slightly and consistently outperformed the rectangular saddle method.

Image Noise Test. Image noise influences the correctness of assumptions of the polynomial fitting model. The performance in the image noise test is summarized in Fig. 6(b). The reference and our methods are clearly influenced by the selected range of image noise. It increases the mean localization error about five times at the strongest 5% noise level. Both our methods outperform the reference method, reducing the mean localization error by two to three times.

[2] Angle from normal of the plane with corner point.

Table 1: Average detection rates of different checkerboard pattern detection methods (Sec. 4.2). The average computation time per frame is also shown. Note that we used the same algorithm (except the initial quad selection), as for deltille grids.

Type	Method	Dataset (# image)					
		Mesa (206)	*uEye* (206)	*GoPro* (100)	*Full* (162)	*Partial* (162)	*Fisheye* (256)
Partial	**Our method**	**100.00%** 13 ms	**100.00%** 249 ms	**100.00%** 740 ms	**100.00%** 184 ms	**99.18%** 199 ms	**99.26%** 298 ms
	OCamCalib [30]	99.99% 120 ms	**100.00%** 597 ms	**100.00%** 637 ms	99.95% 384 ms	92.25% 441 ms	96.05% 706 ms
	OCPAD [14]	98.00% 1111 ms	99.92% 2737 ms	**100.00%** 2329 ms	97.83% 2529 ms	40.94% 3631 ms	26.37% 9325 ms
Full	Geiger et al. [15]	**100.00%** 2144 ms	**100.00%** 4484 ms	**100.00%** 10740 ms	99.94% 4129 ms	94.85% 3467 ms	88.33% 6192 ms
	MATLAB Computer Vision Toolbox [26]	**100.00%** 19 ms	99.84% 203 ms	92.00% 726 ms	99.94% 136 ms	93.30% 129 ms	14.99% 255 ms
	ROCHADE [28]	96.60% 1034 ms	**100.00%** 2809 ms	**100.00%** 7195 ms	97.53% 3092 ms	36.42% 2574 ms	1.95% 8574 ms
	OpenCV [6]	94.66% 94 ms	**100.00%** 94 ms	**100.00%** 645 ms	98.77% 54 ms	38.27% 59 ms	11.72% 409 ms

Perspective Deformation Test. The out-of-plane rotation significantly influences appearance by changing the aspect ratio of the corner image. Fig. 6(c) shows that all three methods perform well for $\phi < 65$. Then, the quantization and image rasterization effects start to influence all methods due to the finite resolution of the generated images. The effect seems to be more pronounced for monkey saddle point that is a higher frequency structure, however, it still shows better performance than other methods in the same conditions. In practice, the deltille pattern allows tessellating regular objects into more facets, which helps to reduce the average view angles to object facets in multi-camera setups.

4.2. Deltille Grid Indexing Performance

In this experiment, we applied our deltille grid indexing method to a conventional checkerboard detection[3] to allow comparison with other state-of-the-art methods [26, 6, 14, 15, 28, 30]. All methods were run on the full and partial checkerboard datasets provided by [14, 28]. Additionally, we compared the results on our new challenging image dataset captured using a fisheye camera with 220° FOV that exhibits high lens distortion.

In the evaluation, we first manually labeled all *visible* checkerboard corners and correct grid topology on all images. Then, we counted corners as *detected*, when they were located within a radius of 3 pixels from any visible corner, considering each visible corner at most once. The performance was measured by the average detection rate $\frac{1}{N}\sum_{i=1}^{N}\frac{\#\text{detected\_corners}_i}{\#\text{visible\_corners}_i}$, where N is the number of images in the dataset. To emphasize the indexing correctness, the number of detected corners was counted as zero, when the topology of the detected grid was incorrect. For a fair comparison, parameters of each method were optimized to result in the overall best performance and kept fixed throughout the evaluation.

[3] with modifications for the rectangular pattern, see extended version of the paper

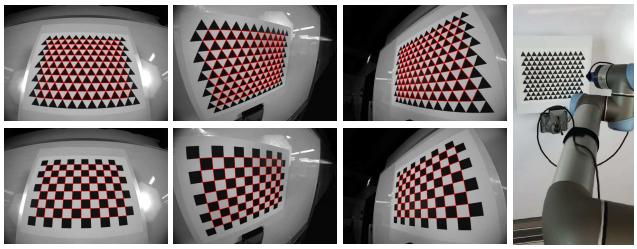


Figure 7: Corresponding deltille and checkerboard pattern images captured using the same trajectory of a robot arm. Both grids detected by our method are drawn in red.

The results summarized in Table 1 show that our method outperformed other methods on all datasets, even though they vary widely in resolution and lens distortion, from narrow FOV lenses with 176×144 images in (*Mesa*), 1280×1024 images in (*uEye*), 4000×3000 images with strong lens distortion (*GoPro*), 1280×720 images in (*OCPAD-full/partial*) up to 1600×1200 images with extreme distortion from 220° FOV lenses in (*Fisheye*) dataset. In particular, our method provides superior performance on the *Fisheye* dataset which is composed of challenging partial checkerboard pattern images. Covering full field of view helps with calibration of strong distortion.

4.3. Evaluation using a Robot Mounted Camera

In this experiment, we compare the performance of other methods with our deltille and checkerboard grid detection method in a real-world camera calibration scenario. A robot mounted camera setup was used to capture two calibration patterns separately. They were mounted at the same location, images were captured with the same trajectory and il-

Table 2: Results of camera calibration and pose estimation on the robot hand sequence, RMSE - Root Mean Squared Error of the reprojection.

Pattern	Method	# Used Corner	Calibration RMSE[px]	Pose Est. RMSE[px]
Deltille grid 10×11	Proposed	11,988	0.1006	**0.1161**
	Proposed with less points	9,592 (random)	**0.0995**	0.1173
Checker-board 8×11	Proposed	9,592	0.1096	0.1283
	Geiger *et al.*	9,592	0.1140	0.1348
	OpenCV	8,976	0.1152	0.1365
	Rochade	9,592	0.1192	0.1372
	OCPAD	9,591	0.1202	0.1382
	MATLAB	9,592	0.1267	0.1634
	OCamCalib	9,592	0.3118	0.3204

lumination conditions in a controlled indoor environment.

We manufactured an $8{\times}11$ rectangular checkerboard with 2cm squares and a 10×11 deltille grid of 2cm equilateral triangles covering similar areas of a high flatness board. A wide FOV camera (Basler Dart) with 1600×1200 resolution was mounted on the robot's end-effector and 109 views were captured covering a half sphere space in front of the board with $5°$ inclination and $30°$ azimuth intervals (see Fig. 7 with the setup and example images). We split each image sequence into two parts:

Calibration. All camera parameters were estimated by performing a calibration with the first half of the sequence in which the camera moved on a half sphere centered at the pattern center within mild perspective angles from $0°$ to $25°$ with respect to the normal direction of the pattern plane.

Pose Estimation. Fixing the obtained intrinsic camera parameters, only the extrinsic camera parameters were estimated on the other half of the sequence, where the camera covered stronger perspective angles from $25°$ to $45°$.

The reprojection errors (RMSE) for each sequence and each phase are summarized in Table 2. Presented rectangular and deltille grid methods outperformed the state-of-the-art methods. The lower RMSE in pose estimation using the deltille grid is achieved not only by more precise corner detection but also by more accurate camera parameters. Moreover, in the same conditions, our deltille grid detection resulted in even smaller calibration errors while observing 25% more corners than the checkerboard, which is a desirable property for accurate camera calibration.

4.4. Multi-view Camera Calibration

In the last experiment, we applied our method to calibrate a multi-camera rig with 4 high-resolution (12Mpx) cameras with narrow $20°$ field of view, approximately 0.8 meters from the target. We precisely manufactured planar triangular panels with high flatness and glued deltille patterns with DelTags on a non-reflective paper, and mounted them on a 3D printed skeleton of an icosahedron, see Fig. 8.

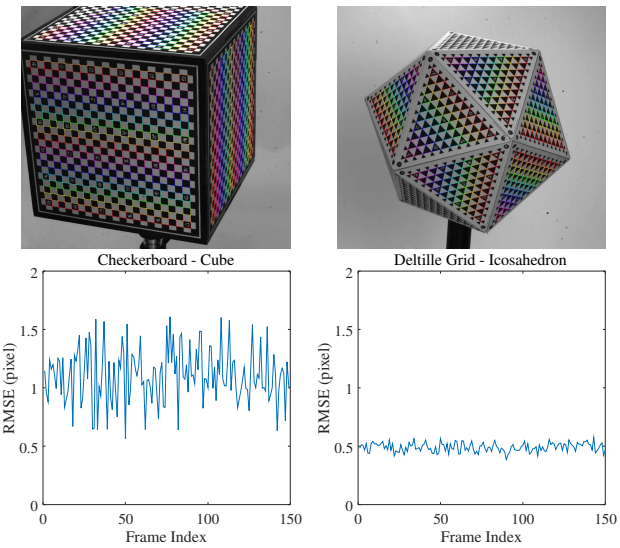


Figure 8: Top row - detections (drawn in rainbow colors) of a deltille icosahedron and checkerboard cube calibration targets from different views. Bottom row - mean reprojection errors over four cameras for each of 150 images.

Similarly, we built a comparable cube target with checkerboard patterns and AprilTags [27]. Both targets were precalibrated by finding relative poses of all planes.

A dataset with 150 images for each of the targets and each of the four cameras was taken in controlled conditions: synchronized shutter, same illumination and a similar distance from the cameras. Deltille grid and regular checkerboard detector were used to get 2D to 3D point correspondences from respective targets. Note that at least three faces of the icosahedron target were detected in any viewpoint in contrast to one face for the cube target.

The mean reprojection errors (RMSE) for each of the frames are shown in Fig. 8. Each *frame* represents the mean of images from all cameras. These results, as well as statistical analysis of the camera parameters on the sequence, show that the icosahedron provides stable, and more precise calibration with significantly lower reprojection errors.

5. Conclusions

We present a new calibration ecosystem based on a novel method for precise localization of monkey saddle points on deltille grids. The polarities, orientations of the intersecting edges derived from the fitted polynomial surface, were used together with DelTags to establish robust indexing on multifaceted objects and partially visible grids. A thorough experimental evaluation showed that deltille grid detection is superior to both the rectangular case, and state-of-the-art corner detection methods leading to more precise calibrations. The extended version of the paper, datasets and code are available at `https://github.com/deltille`.

References

[1] E. Abbena, S. Salamon, and A. Gray. *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 2006. 3

[2] A. S. Alturki and J. S. Loomis. X-corner detection for camera calibration using saddle points. *WASET International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(4):554–559, 2016. 3

[3] M. Berry and J. Hannay. Umbilic points on Gaussian random surfaces. *Journal of Physics A: Mathematical and General*, 10(11):1809–1821, 1977. 3, 4

[4] D. Bommes, B. Lévy, N. Pietroni, C. Silva, M. Tarini, and D. Zorin. State of the art in quad meshing. In *Eurographics*, 2012. 2

[5] J.-Y. Bouguet. Camera calibration toolbox for Matlab. 2004. 1, 2

[6] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media., 2008. 1, 2, 5, 7

[7] D. Chen and G. Zhang. A new sub-pixel detector for x-corners in camera calibration targets. In *WSCG*, pages 97–100, 2005. 2

[8] J. H. Conway, H. Burgiel, and C. Goodman-Strauss. *The symmetries of things*. CRC Press, 2008. 1

[9] H. S. M. Coxeter. *Regular polytopes*. Courier Corporation, 1973. 1

[10] G. Darboux. *Leçons sur la théorie générale des surfaces*. Gauthier-Villars, 1896. 4

[11] E. F. D'Azevedo. Are bilinear quadrilaterals better than linear triangles? *SIAM Journal on Scientific Computing*, 22(1):198–217, 2000. 2

[12] A. De la Escalera and J. M. Armingol. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, 10(3):2027–2044, 2010. 2

[13] M. Fiala. Artag, a fiducial marker system using digital techniques. In *IEEE Conference on Computer Vision and Pattern Recognition (ICCV)*, volume 2, pages 590–596, 2005. 1, 5

[14] P. Fuersattel, S. Dotenco, S. Placht, M. Balda, A. Maier, and C. Riess. Ocpad – occluded checkerboard pattern detector. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016. 1, 2, 7

[15] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943, 2012. 2, 7

[16] B. Grünbaum and G. C. Shephard. *Tilings and patterns*. Freeman, 1987. 1

[17] M. Hansard, R. Horaud, M. Amat, and G. Evangelidis. Automatic detection of calibration grids in time-of-flight images. *Computer Vision and Image Understanding*, 121:108–118, 2014. 2

[18] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988. 1, 2

[19] J. Heikkilä. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1066–1077, 2000. 3

[20] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1335–1340, 2006. 3

[21] J. Kannala, J. Heikkilä, and S. S. Brandt. Geometric camera calibration. *Wiley Encyclopedia of Computer Science and Engineering*, 2008. 3

[22] J. Kim, P. Gurdjos, and I. Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):637–642, 2005. 3

[23] J.-S. Kim, P. Gurdjos, and I. S. Kweon. Euclidean structure from confocal conics: Theory and application to camera calibration. *Computer Vision and Image Understanding*, 114(7):803–812, 2010. 3

[24] L. Lucchese and S. K. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In *Asia-Pacific Conference on Circuits and Systems*, volume 2, pages 191–195, 2002. 1, 2, 3

[25] J. Mallon and P. F. Whelan. Which pattern? biasing aspects of planar calibration patterns and detection methods. *Pattern Recognition Letters*, 28(8):921–930, 2007. 3

[26] Mathworks Inc. MATLAB: Computer Vision Toolbox (R2016a), 2016. 7

[27] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, 2011. 1, 5, 8

[28] S. Placht, P. Fürsattel, E. A. Mengue, H. Hofmann, C. Schaller, M. Balda, and E. Angelopoulou. Rochade: Robust checkerboard advanced detection for camera calibration. In *European Conference on Computer Vision (ECCV)*, pages 766–779, 2014. 2, 3, 7

[29] M. Rufli, D. Scaramuzza, and R. Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3121–3126, 2008. 2

[30] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5695–5701, 2006. 1, 2, 7

[31] Z. Wang, W. Wu, X. Xu, and D. Xue. Recognition and location of the internal corners of planar checkerboard calibration pattern image. *Applied Mathematics and Computation*, 185(2):894–906, 2007. 2

[32] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 5