X-Tag: A Fiducial Tag for Flexible and Accurate Bundle Adjustment

Tolga Birdal[†,*], Ievgeniia Dobryden[†] and Slobodan Ilic[†,*]

[†] Computer Aided Medical Procedures, Technische Universitat München, Germany
[*] Siemens AG, München, Germany,
tolga.birdal@tum.de, ievgeniia.dobryden@tum.de, slobodan.ilic@siemens.com

Abstract

In this paper we design a novel planar 2D fiducial marker and develop fast detection algorithm aiming easy camera calibration and precise 3D reconstruction at the marker locations via the bundle adjustment. Even though an abundance of planar fiducial markers have been made and used in various tasks, none of them has properties necessary to solve the aforementioned tasks. Our marker, X-tag, enjoys a novel design, coupled with very efficient and robust detection scheme, resulting in a reduced number of false positives. This is achieved by constructing markers with random circular features in the image domain and encoding them using two true perspective invariants: cross-ratios and intersection preservation constraints. To detect the markers, we developed an effective search scheme, similar to Geometric Hashing and Hough Voting, in which the marker decoding is cast as a retrieval problem. We apply our system to the task of camera calibration and bundle adjustment. With qualitative and quantitative experiments, we demonstrate the robustness and accuracy of X-tag in spite of blur, noise, perspective and radial distortions, and showcase camera calibration, bundle adjustment and 3d fusion of depth data from precise extrinsic camera poses.

1. Introduction

Identification and pose estimation of planar fiducial markers has a long gone history in photogrammetry, augmented reality and computer vision. 2D planar markers, one common form of fiducials, are the primary instruments for obtaining reference coordinates in controlled scenes. They were successful in constraining the algorithms in many tasks such as 3D reconstruction and camera calibration [5]. These simple artificial landmarks can be designed in a task specific way, and can be located with high speed, high repeatability and accuracy, contrary to the natural features.

In spite of all the developments in this field (see Fig. 3), practitioners still face the problem of mis-detected codes, low true positive rates, or inaccurate localization of the markers due to various distortions. Moreover, different applications have different demands, requiring custom code designs. Some of the available markers are not fully perspective invariant [29], while the others which have this property either require a good estimate of the intrinsics [6] for getting the marker pose or the detection complexity enormously increases with the increase of their number [7]. In this work, we propose the novel X-tag as a flexible alternative, which enjoys true projective invariance, high accuracy localization and fast identification. In the core of the method, we use a random-dot style marker design, which is described by a set of extended joint projective invariants, composed of multiple cross ratios and intersection preservation constraints. We then use a geometric-hashing framework, as illustrated in Fig. 2, to index a set of pre-generated dot positions. Simply, this forms the marker database. The decoding is cast a retrieval problem, in which the same features, extracted from query tags, are matched across the database through an inverted file. The correct matches are subject to further verification using Homography constraints. In contrast to previous works, which are also based on random dot patterns [29], our marker is truly projective invariant and thus is robust to viewpoint changes. This lets

Figure 1: Our markers, can be used in very cluttered scenes.

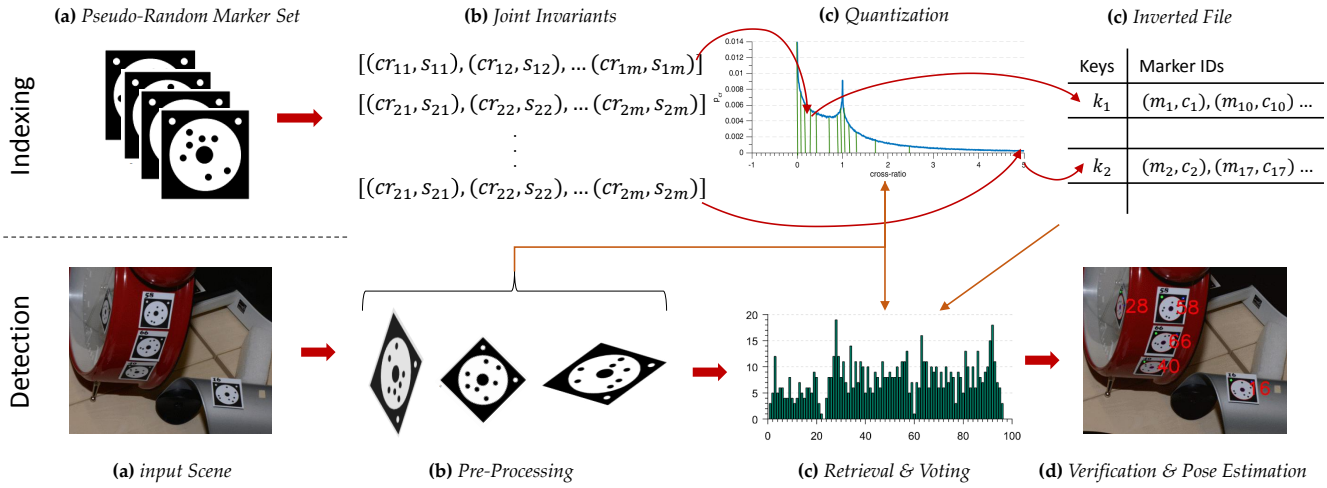


Figure 2: Our pipeline. *See text for details.*

us to find more correct tags, enabling more advanced applications such as camera calibration, bundle adjustment and 3D object reconstruction. Due to the adjustable size of the marker, we could design codes which are resilient against radial distortions. Moreover, thanks to the increased number of internal dots, we could obtain more reliable pose estimates, and thus more reliable initialization for procedures such as bundle adjustment.

Our design advances on the good traits of both its ancestors: The square and circular tags. It is easier to detect than square tags, while being even more accurate than the circular counterparts. We apply the X-tag to the problem of camera calibration, bundle adjustment and object reconstruction. Our results clearly outperform the state of the art.

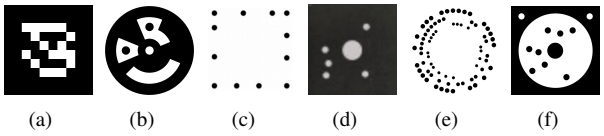
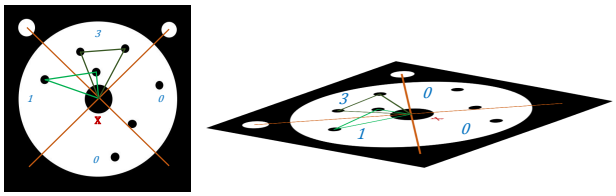


Figure 3: Markers from different methods. **(a)** AR-Tag, **(b)** Intersense, **(c)** Pi-Tag, **(d)** Linearis, **(e)** Rune-Tag, **(f)** Ours

2. Prior Art

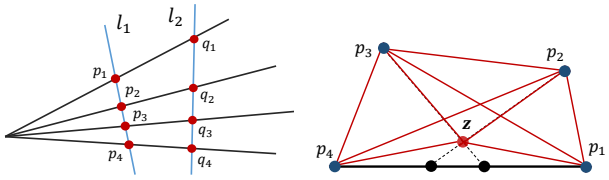
Markers enjoy a wide literature in computer vision and augmented reality. While the history is rather unclear, the current simple targets are square markers. They typically contain the description in the inner region of the square as a form of binary code, or a unique image/geometry. AR-Tag [13], Aruco [14], ARToolkit [18] and AprilTag [26] are some examples. On the pro side, these targets are very efficient to locate and identify either by correlation methods, or by a binary decoding schemes. However, the use of the squares, rectangles and lines limit the accuracy when detecting subpixel locations on the markers. This makes these markers inapplicable to certain scenarios, requiring high accuracy, such as camera calibration. Moreover, the necessity to spot a quad (collinearity) causes the marker to get affected from the radial distortions and occlusions easily. Thus, some of the aforementioned studies had to explicitly address such issues.

Motivated by the limitations of corner features of the square tags, the next generation fiducial tags made use of circular features, which are more accurate to localize and less sensitive to noise. Intersense [23] combines data-matrix concept with concentric circles to create bar-coded markers. Their design allows generation of 2^{15} codes for identification, but the pose estimation remains to be problematic [18]. Pi-Tag [7] uses a fiducial design composed of ordered circles. The detection benefits from cross-ratio invariants to handle perspective distortions. While, this approach is promising, the matching of cross-ratios is an issue, and the worst-case complexity is reported to be $O(N^4)$, which could quickly become impractical. Inspired by [24], random dots [29] choose to approximate the projectivitiy with affine constraints, resulting in an easier and more stable feature. The authors also devise a geometric hashing [30] framework to cast the code reading problem to a retrieval one. Yet, random dots still exhibit affine features and cannot handle full projectivity. In addendum, due to the frameless design, a large number of dots are required for reliability, increasing the computational load. In the recent state-of-the-art work [6], authors of Pi-Tag take a different standpoint proposing RuneTag, a non-concentric and disconnected arrangement of circular marks around multiple rings, invariant to the projective transformations. This



(a) (b)

Figure 4: (a) Canonical marker frame with features overlayed. (b) Projectively transformed version of (a).



(a) (b)

Figure 5: Illustrations of cross ratio.

reduces the burden imposed by the feature extraction. The result is a very robust and occlusion tolerant fiducial marker, being reasonably fast to detect. A common point in all three designs of [29, 7, 6] is the fact that the tags are composed of individual circles, which link to form the whole. While this eases the processing stage, introduction of clutter, especially in the form of false ellipses causes the runtime to significantly increase, if not fail the detection completely. Another observation is that, many of the codes are designed to be large and redundant, i.e. close-by placement of individual ellipses are prone to merge under camera noise or blur, especially in distant views. This is not desired for applications targeting camera calibration, as it is important to distribute as many markers in 3D space as possible.

The circular fiducials are also the method of choice, when implementing photogrammetry systems. Linearis [1], Aicon3d [3] or GOM TriTop [15] are some of those end-to-end measurement systems. The exact algorithms used in such products are not publicly available and hidden. Yet, we are aware, for example, that Linearis cannot handle large perspective distortions.

3. Method

We'll now deeply review the design, description and retrieval of X-tag, with an application to bundle adjustment.

3.1. Marker Design

X-tag consists of a random arrangement of several (>5) black circular marks, distributed around a black central dot and on a white background. The design includes an additional black frame, acting as a contrast agent to ease the localization. The actual shape of the frame is irrelevant and can also be designed to be circular. Additionally, two white circles are placed on one side of the base frame. They are used for multiple purposes of feature extraction, verification and pose estimation. The marker is shown in Fig. 4.

3.2. Marker Description

We describe X-tag via a new *extended invariant set*(EIS). EIS is composed of two parts: Cross ratios and intersection preservation constraints encoded by the sector type.

3.2.1 Cross Ratio

Cross ratio (CR) is the fundamental invariant in projective geometry. Its simplest form is defined for a pencil of lines, passing through a center O and intersecting two lines ℓ_1 and ℓ_2 at points $\{p_1, p_2, p_3, p_4\}$ and $\{q_1, q_2, q_3, q_4\}$ respectively. This configuration is visualized in Fig. 5(a). The cross ratio of 4 such collinear points is defined as:

$$cr(p_1, p_2, p_3, p_4) = cr(q_1, q_2, q_3, q_4) = \frac{|p_1p_3||p_2p_3|}{|p_1p_4||p_2p_4|} \quad (1)$$

This invariant is naturally extended to 2D space [27], when the points are non-collinear, but co-planar. In this case, the configuration of five points defines the cross ratio using the ratio of product of triangle areas:

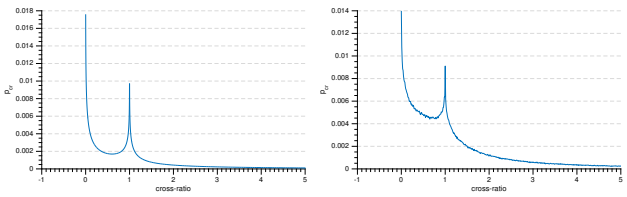
$$cr_{2D}(\mathbf{z}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = \frac{\Delta(\mathbf{z}, \mathbf{p}_1, \mathbf{p}_2)\Delta(\mathbf{z}, \mathbf{p}_3, \mathbf{p}_4)}{\Delta(\mathbf{z}, \mathbf{p}_1, \mathbf{p}_3)\Delta(\mathbf{z}, \mathbf{p}_2, \mathbf{p}_4)} \quad (2)$$

Δ denotes the triangle area. This is illustrated in Fig. 5(b). Clearly, one could generate multiple CR, by altering the permutations of points. Thus, a set of points define 24 CR, of which only 6 are unique.

Probability Distribution Both PDF and CDF of cross ratio are analytically defined in multiple works [2, 17]. We found out that also in practice, distribution of a set of CR closely approximates the analytical one. Fig.6 plots the analyitcal PDF and CDF of cross ratios as well as the estimated one over a synthetic point dataset.

Joint Invariants Even though [4] raise a contradictory claim, it is well known that cross ratio is very sensitive to noise [22, 21, 20]. This sensitivity and the non-uniqueness of single cross ratios, however, can be circumvented up to a certain extent by relying on multiple invariants extracted from multiple sets of points [4]. Such a set is termed as the joint invariants, and defines the point set uniquely up to a projective transformation.

X-tag's first invariant feature consists of the set of all cross ratios that can be computed from the inner dots, taking the central dot as the 5^{th} point $\mathbf{z}$. Fixing such a point



(a) (b)

Figure 6: **(a)** Analytical distribution of cross ratios. **(b)** Approximated distribution of cross ratios.

increases the stability as opposed to complete randomness [29]. To further boost the discrimination power, we introduce the latter descriptor, *sector type*, which relies on the preservation of the intersection of lines.

3.2.2 Sector Type

Formally, we partition our fudicial into 4 regions. 4 partitions are given birth by the intersection of two lines formed by joining the outer white dots, with the center point. These lines also separate the large inner white circle into 4 partitions. For each CR computation, i.e. for each 4 points taken out of the randomly generated points, our descriptor encodes the presence of the points within the sector. This can be seen in Figure 4. In particular, the sector type $s = \{x \in \mathbb{R} : 4 \leq x \leq 500\}$ is computed as the polynomial expansion:

$$s(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4) = s_1\alpha^3 + s_2\alpha^2 + s_3\alpha + s_4 \qquad (3)$$

where s_i is the number of points out of $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$, lying within sector i. For our configuration we set $\alpha = 5$. Note that due to the intersections being preserved, this is a true projective invariant.

Our final descriptor is the concatenation of the two distinct invariants described here, forming the set $\mathbf{D} = \{\mathbf{d}^i = \{cr^i_{2D}, s^i\}\}$.

3.3. Tag Localization

The first step in the pipeline is the low level image processing, in which we identify and localize the X-tag candidates. To do so, we apply a simple image processing algorithm. First, the dark regions are selected as marker candidates. Then, a connected component labeling is performed and blobs, which do not satisfy a relaxed set of constraints (area and dimensions) are discarded. Later, each candidate blob is tested for inclusion of a light (e.g. white) region and sufficient circular points. We also check for the two white dots, and the center dot, explicitly. At each step of this operation, elliptical regions are selected via the properties of the elliptic axis. Note that, even if we find false positives at this stage, the database search and verification are likely to fully suppress them.

Indexing Markers and Database Creation For each marker with id i, we obtain $\mathbf{D}_i$, a long, extended descriptor. Indexing such a descriptor for nearest neighbor retrieval purposes is not always trivial. Here, we explain how we benefit from the special nature of our descriptors to use them in a geometric hashing framework.

The basic idea behind our algorithm is that we quantize the descriptor for each marker sequentially and store the quantized codes in an inverted file, along with the occurrence information c_i and the marker id m_i. c_i also helps us to compress the inverted file, as multiple occurrences of the same marker are stored as a single entry in the hashtable. The *Indexing* part of Fig. 2 illustrates this scheme.

Because the probability distribution of cross ratios is highly non-linear, a simple uniform quantization of the features wouldn't work, i.e., many cross ratios would fall in the same bin. Thus, we rely on a quantization scheme, which is aware of the joint feature distribution (see section 3.2.1). Our essential idea is to create a binning such that the integral of PDF in each bin is roughly equal. Formally, let f denote the PDF, F the CDF and F^{-1} the inverse CDF of cross ratios. Because we now (or can) estimate both F and F^{-1}, we choose to map any given cross ratio cr via CDF and to perform a uniform quantization in this domain. Formally a $b - bit$ quantized value $\mathbf{f}_q[cr]$ is obtained by:

$$\mathbf{f}_q[cr] = b\left\lfloor \frac{F(cr) + E_q[cr]}{b} \right\rfloor, \quad E_q[cr] \sim U\big(\big[-\frac{\delta}{2}, \frac{\delta}{2}\big]\big) \qquad (4)$$

$E_q[n]$ is uniformly distributed, due to the assumption that the errors are uniformly spread into the bins. In our implementation, we prefer to use the approximate CDF F^*, instead of the analytical F as F^* is a better representative of the data-subset. Such quantization requires a look-up over the CDF, which we perform via binary-search. Faster implementations might benefit also from interpolation search, as the distribution is available. By quantizing directly on F, we could avoid using F^{-1} to map back to the PDF, f. However, Fig. 2 plots the partitions in the PDF domain.

While the cross ratios are non-uniformly distributed real values, the sector type is a simple integer and is very friendly for indexing operations. Our hash index is simply $h_{cr} = \{\mathbf{f}_q[cr], s_{cr}\}$.

Identification of Tag IDs Once the features are extracted for all combinations of points $\{\mathbf{p}_i\}$ in a candidate scene, we could resolve the tag id using the inverted file. To avoid the distance computation overhead and to retain the robustness, we achieve this through a procedure, similar to Hough voting. Each quantized feature h_{cr} retrieves a set of probable markers from corresponding bucket and casts a vote to the corresponding marker id. The vote is proportional to the occurrence in the database. Ideally, after voting for all joint invariants, the maximum vote reveals the marker ID.

3.4. Verification and Pose Estimation

Even though the voting is very robust, it doesn't always guarantee the best solution. For that reason, we retain a set of surviving hypotheses for further verification. Moreover, the match-ability of the marker necessitates the correct identification of only three points: The center and two support points. This leaves us with one unknown to determine the projective transformation. Note that, using conics for pose estimation might be bad in this situation because it is very likely that a single ellipse would appear as a small dot.

To find the ID of the 4^{th} point, we could simply enumerate over all the possible point combinations and evaluate the reprojection error, but to save computation, we instead apply similar voting procedure as we use for matching of marker IDs. For each dot in marker cross-ratios with all other points are calculated and stored in the hashtable. On the verification stage the voting for dot ID is performed similarly to voting for the marker ID using all cross ratios for given dot. Resolving the correspondences finally becomes more efficient since we verify the best hypothesis first. Formally, the fourth landmark is found via:

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} \sum_{j=1}^{n} \|\mathbf{H}(\mathbf{p})\mathbf{m}_j - \mathbf{r}_j\| \tag{5}$$

where $\mathbf{H}(\mathbf{p})$ is a homography found after matching reference point to point $\mathbf{p}$ via voting procedure, $\mathbf{m}_{1..n}$ are dots locations inside marker, $\mathbf{r}_{j..n}$ are their correspondences in reference frame. Under occlusions or noise, a one-to-one correspondence is enforced for robustness. $\mathbf{H}$ is computed from 4 points using DLT algorithm [16]. The final pose is estimated using PnP algorithm [19] using all dot coordinates in marker. This is superior to standard square tags in two aspects: 1. The used dots are circular and are more accurate to localize. 2. We have always $N_d > 4$ dots in our marker. As we utilize all the found dots, our estimation is expected to be more correct.

4. Multi-Camera Bundle Adjustment

A useful application area for X-tag is camera calibration and bundle adjustment (BA)[28]. We propose to use X-tag as a calibration target and compute the extrinsics and intrinsics with BA. Our idea is to make the user entirely free from the using precise targets. We rather rely on the central ellipse of X-tag to give us the image cue. Our approach is similar to [12], but we do not constrain ourselves to planar targets. Given a set of images, captured either from moving cameras, or changing scenes, we run the following optimization:

$$\min_{\mathbf{P},\mathbf{X}} \sum_{i=1}^{m}\sum_{j=1}^{n} \rho\big(w_{ij} d(\mathbf{P}_i\mathbf{X}_j, \mathbf{x}_{ij})^2\big) + \sum_{i=1}^{k}\sum_{j=1}^{k} (d(\mathbf{X}_i, \mathbf{X}_j) - \sigma_{ij})^2$$

where $\mathbf{X}_1..\mathbf{X}_n$ are 3D points, $\mathbf{P}_1..\mathbf{P}_m$ are projection matrices of m cameras, $\mathbf{x}_{ij}$ is image coordinate of point j for camera i. The distance $d(\cdot)$ between any two points is subject to a weighting w_{ij} which based on the detection quality of image points. $\rho(\cdot)$ is a robust Cauchy norm. We compute its scale parameter from the elliptic axis properties. The second term is regularization that brings the reconstructed scene to metric space by keeping distance between known 3D points $(\mathbf{X}_i, \mathbf{X}_j)$ at the value σ_{ij}. We initialize this BA procedure from the pose of the most frequently visible marker. The pose is estimated using the inner random dot locations, w.r.t. the canonical marker frame. In BA, we simultaneously solve for $(\mathbf{P}, \mathbf{X})$, using Brown distortion model [10].

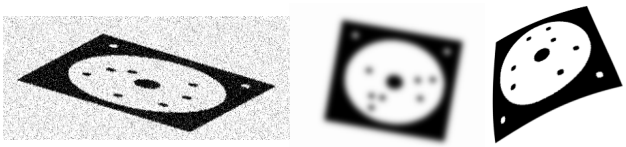


Figure 7: Shots from synthetic scenes for different noise, blur and radial distortion conditions.

5. Experimental Evaluation

We assess the performance of our method with extensive qualitative and quantitative evaluations.

Evaluation Metrics Throughout this paper, the individual errors for the distinct pose components (rotations and translations) read as:

$$\epsilon_{\mathbf{R}}(\mathbf{R}_1, \mathbf{R}_2) = \arccos\left(\frac{\text{trace}(\mathbf{R}_1^{-1}\mathbf{R}_2) - 1}{2}\right) \tag{6}$$

$$\epsilon_{\mathbf{t}}(\mathbf{t}_1, \mathbf{t}_2) = \|\mathbf{t}_1 - \mathbf{t}_2\| \tag{7}$$

We will also speak about performance metrics of accuracy, which is defined as $Ac = (TP + TN)/N_{exp}$.

5.1. Experiments on Synthetic Data

We first evaluate the validity and robustness of our approach on a synthetic set. This way, we observe the performance under various degradation and capture the behavior of parameters. For this stage, our synthetic data is composed of $N_M = 2000$ markers. For testing, we sample 200 of this set and combine it with 20 other markers, which are outside of the database. The test data is subject to 50 warps per image, each having a different augmentation. These augmentations include blur, additive noise and radial distortion. The synthesized images are shown in Fig. 7.

Effect of Hashtable Size As the initial stage of experimentation, we would like to tune our system to use the optimal parameters. We asses how the performance, as well as

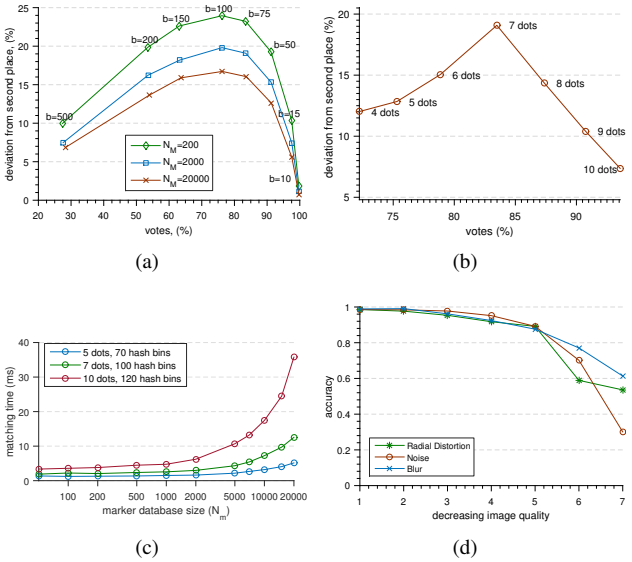


Figure 8: **(a)** Maximum percentage of votes and deviation from next best place in voting table for different sizes of marker databases **(b)** Votes for markers with different number of dots using hashtable with 100 bins **(c)** Matching time for one marker using optimal size of hash table **(d)** Marker detection rate on synthetic scenes

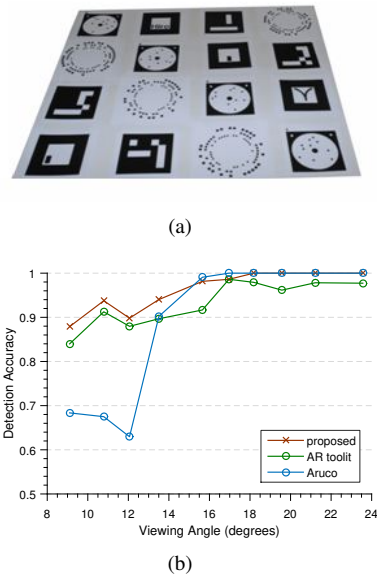


Figure 9: **(a)** Robustness to Viewing Angle.

the computation time is affected with the varying number of markers, number of bins and number of dots. Therefore, we conduct incremental evaluations. First, we want to find the optimal size of a hash table for the marker set. Ideally, the true marker ID should get the most votes from the hash table. So by tuning the hash table size we aim for the maximum percentage of votes for marker with the highest rank and largest deviation from the second best. While the desired number of dots is a parameter for our method, within the context of experiments, we fix it to 7. On our synthetic set, we conduct the aforementioned performance analysis and plot this in Fig. 8(a). It is visible that independent of the database size the optimal number of bins for markers with 7 dots is 100. It is interesting to see that only the number of cross ratios for one marker influences the optimal hash table size.

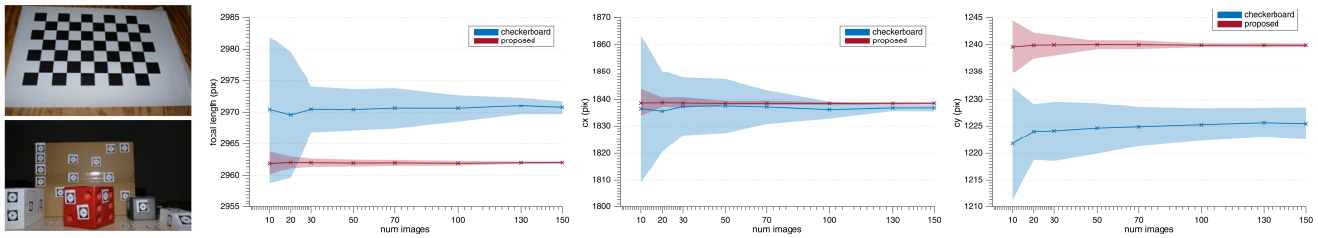
In a further experiment in Fig. 8(b), we fix the number of markers in our database to 2000 and also the hashtable bin sizes to 100. By varying the number of dots we could see that having 100 bins for markers with 7 dots will provide optimal voting results for that configuration. It is therefore immediate that when more dots are desired, the hashtable size should be tuned accordingly.

Next we evaluated matching time for markers with optimal hash table size. The time of matching depends both on number of cross ratios for one marker and the number of markers in database. Matching one marker takes $O(n\log(b) + bN)$ time in worst case, where n is number of cross-ratios, b - number of bins, and N - size of markers database. The matching time for markers with 5, 7 and 10 dots for Intel i7 3.20 GHz processor is shown on Fig. 8(c). In practice matching a single marker takes less than $3ms$, given that in real life, a database of 2000 markers is more than sufficient. Note that, thanks to the grouped inverted file structure and the quick voting, our computational time only marginally increases even when the database size is significantly increased. In that manner, our approach is very scalable and therefore amenable for real life applications requiring an abundance of fudicial tags.

Robustness to Image Distortions We assess the robustness (as detection accuracy) to different noise and perturbations and the computational performance in Fig. 8(d). The x-axis shows the image quality, which is gradually reduced by different augmentations. For real-life scenarios, quality level hardly exceeds 3. It is evident that while X-tag is generally robust, it is least affected by the blur. Increasing noise would have the most severe effect, while significant radial distortion is in general handled slightly better than noise.

5.2. Real Scenarios

Robustness to Viewing Angle To quantify the robustness under perspective changes, we print 4 of each RuneTag[6], Aruco[14], ARToolkit[18] and X-tag on a common paper as shown in Fig. 9(a). We image 4 different rotations of this pattern in varying distances and from severe ($8°$) to mod-



(a) (b) (c) (d)

Figure 10: **(a)** Images for checkerboard calibration and calibration with X-tag. **(b)** Focal length estimation **(c)** Estimation of principal point (x) **(d)** Estimation of principal point (y)

erate (25°) camera angles. We then run each detector and compute the overall detection rate. The results are plotted in Fig. 9(b). It is shown that while most methods perform very well, ours is slightly above all others, justifying the good detectability. RuneTag [6] and RandomDots [29] are not taken into this plot because: 1. RuneTag circles quickly get invisible with the increasing distance and tag starts to underperform. 2. RandomDots are only robust up to affine warps and cannot handle perspective variations. Thus, the detection rate appears to be low for this experiment. While the square fiducials are known the best for this type of challenge, our circular tag still outperforms the rest.

How Reliable is the Estimation of Intrinsics? As explained in Section 4, our method is suitable for complete bundle adjustment, where the intrinsics and extrinsics are jointly minimized. While it has always been a challenge to assess the accuracy of calibration (as exact principal point and focal length are not directly observable), we argue that, it is more important to obtain repeatable estimates, rather than accurate ones i.e. one could always use an offset to compensate for biases, once the repeatibility is achieved. We, therefore, use a slightly unorthodox experimentation and run our bundle adjustment multiple times for intrinsics estimation, repeatedly. We perform the same test with a OpenCV checkerboard calibration [9], which seems to be the de-facto standard in computer vision. The number of detected corners roughly equate to the number of detected ellipse centers. Fig. 10 plots our estimations along with the ones from checkerboard for principal point, as well as focal length. The standard deviation overlays the curve. It is apparent that our results are more deterministic and less prone to initialization errors, as well as errors in feature point computation. The deviation plots indicate that even with small number of images our estimations are more reliable than the standard techniques. Note that, an analogous experiment shows a similarity between OpenCV's checkerboard method and RuneTag calibration [6]. It is also worth mentioning that while both OpenCV and RuneTag rely on the availability of the 3D model of a calibration pattern, we are completely pattern-free and our markers could be positioned anywhere in the observed space.

Evaluation of Pose Estimation Here, we evaluate the power of a single tag for estimating extrinsic pose. For that, we set up a scene of 80 markers composed of 40 Aruco and 40 ours as shown on Fig. 11. This scene is then viewed from 100 distinct camera locations, including viewpoint variations. Afterwards, we run our bundle adjustment proposed in Section 4 on Aruco markers and our markers separately and multiple times. We always initialize the adjustments by using the pose of one of the markers selected as a reference. We deliberately alter the selected reference over different BA runs to reduce the selection bias. BA procedure corrects for 3D locations as well as camera poses. Finally, the refined pose of the selected reference tag is compared against the initial estimation, both for Aruco and ours, disjointly. The difference in these poses is naturally the computed update by BA. The smaller the update is, the more correct the initialization, and therefore the better the estimation of extrinsics from a single marker. The results, averaged over a set of runs, are shown in Table 1 for the scene in Fig. 11. The findings indicate that our markers are much better at providing camera pose than Aruco. The pose difference is computed via Eq. 6. The reason why the reprojection error enjoys a relatively higher improvement is because it absorbs both the errors on the pose and on the 3D structure. An improvement of both increases the impact on the reprojection.

Figure 11: Scene

Table 1: Errors

	Aruco	Ours
Rot Err	0.0216	0.0145
Tra Err	0.0091	0.0067
Repr Err	0.2472	0.0694

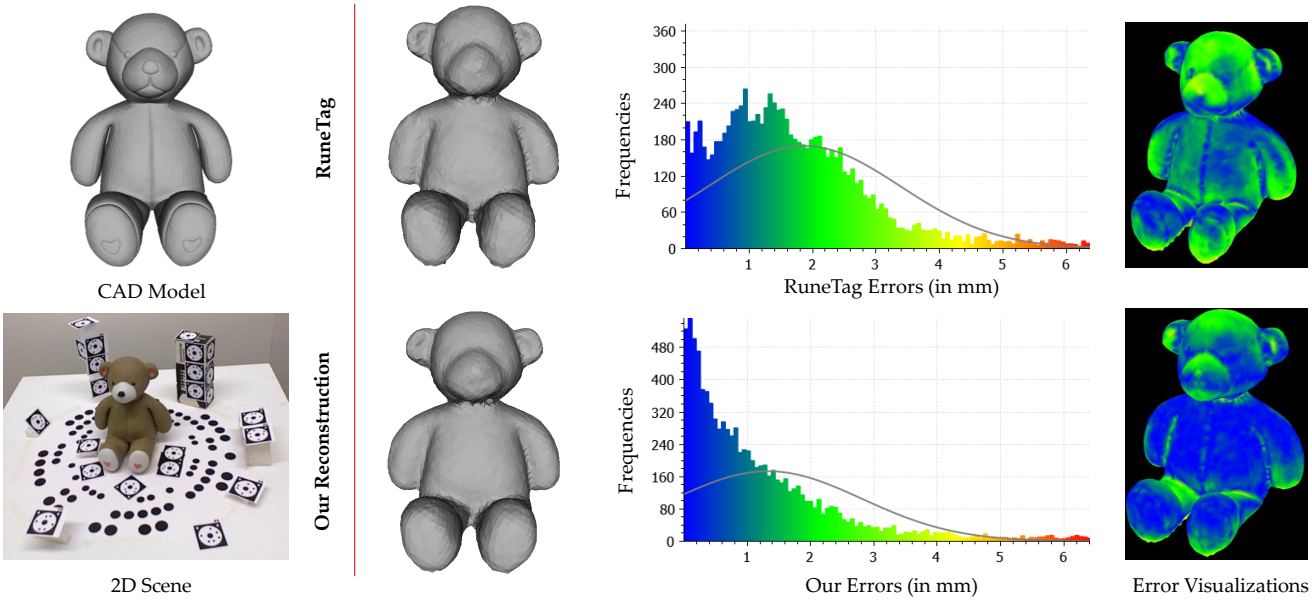


Figure 12: Comparisons of reconstructing Teddy object with the aid of RuneTag and X-Tag.

Object Reconstruction Performance At last, we evaluate our method for the problem of 3D object reconstruction using depth sensors. Our procedure is similar to KinectFusion [25], however, we replace the ICP (iterative closest point) [8] stage with poses coming from X-tag detection, and perform the conventional SDF-fusion [11]. This way, the object is not needed to be tracked and we could operate with only a handful of scans. Our setup consists of *Teddy* object, which is a 3D print from an ideal CAD model. The object is positioned on a turn-table sequence. The tags are distributed around different regions of the space. Because the state of the art fiducial tag for object reconstruction is RuneTag [6], we find it sufficient to evaluate against this method. Therefore, we also augment the scene with RuneTag marks. A shot from this setup can be viewed in the first column of Fig. 12. Following the sequential image acquisition, we then run our bundle adjustment both for our tags and for RuneTag. Note once again that, X-tag BA assumes neither camera calibration nor an apriori 3D model, while RuneTag is designed to operate best on calibrated settings (While RuneTag could also handle uncalibrated case, this capability depends on an enumeration over all possible focal lengths until a reasonable estimate is found. We consider this to be still calibration dependent.). Therefore we initialize RuneTag with the correct bundle adjusted intrinsics and let it estimate only the camera poses. BA output provides us both refined poses and point coordinates. In this stage, we use only the poses and discard the 3D structure. We retrieve this structure from the depth images of the 3D scanner.

We convert the absolute poses to the relative ones and starting from an initial volume, we run an SDF Fusion to capture the final 3D reconstruction. Thanks to the presence of the ideal model, we compare both results to the ground truth. These comparisons are depicted in Fig. 12. The colors are associated to the unsigned error magnitudes. Because our markers are located on non-coplanar regions of the space, they are better at binding the 3D transformation. This demonstrates that, better geometric constraints are more favorable than the availability of prior calibration targets. Finally, one could always think of the bundle adjusted 3D points as our *calibration rigs*.

6. Conclusion

In this paper we proposed the *X-tag* and posed it as a flexible tag amenable for model-free calibration, pose estimation and 3D reconstruction. X-tag is truly invariant to projective changes, detectable in high clutter and its robustness to radial distortions are demonstrated. Moreover the matching time of a single marker is extremely fast and the devised method is suitable to scaling large marker sets. Our fiducials can be generated and spotted in varying sizes within the same application, without any constraints.

There are many possible future directions. Even though not experimented here, X-tag has large potential for occlusion handling. That's because the joint invariants of random arrangements make the code partially redundant. Moreover, multiple X-tag could be assembled to form and act as 3D models. In the future we will also focus on the extending X-tag to be able to do a complete SLAM.

References

[1] L. 3D. Photogrammetry by linearis 3d, 2016. 3

[2] K. Aastrom and L. Morin. Random Cross Ratios. Technical Report IMAG-RT - 92-088 ; LIFIA - 92-014, 1992. 3

[3] AICON3D. Aicon 3d systems - move inspect technology - dpa, 2016. 3

[4] R. Arora, Y. H. Hu, and C. Dyer. Estimating correspondence between multiple cameras using joint invariants. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 805–808. IEEE, 2009. 3

[5] B. Atcheson, F. Heide, and W. Heidrich. Caltag: High precision fiducial markers for camera calibration. 2010. 1

[6] F. Bergamasco, A. Albarelli, L. Cosmo, E. Rodola, and A. Torsello. An accurate and robust artificial marker based on cyclic codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016. 1, 2, 3, 6, 7, 8

[7] F. Bergamasco, A. Albarelli, and A. Torsello. Pi-tag: a fast image-space marker design based on projective invariants. *Machine vision and applications*, 24(6):1295–1310, 2013. 1, 2, 3

[8] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992. 8

[9] G. Bradski. *Dr. Dobb's Journal of Software Tools*. 7

[10] D. C. Brown. Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866, 1971. 5

[11] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 8

[12] S. Daftry, M. Maurer, A. Wendel, and H. Bischof. Flexible and usercentric camera calibration using planar fiducial markers. In *in British Machine Vision Conference (BMVC*. Citeseer, 2013. 5

[13] M. Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE, 2005. 2

[14] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. 2, 6

[15] GOM. Tritop: Gom, 2016. 3

[16] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 5

[17] D. Huynh. The cross ratio: A revisit to its probability density function. In *Proceedings of the British Machine Conference, pages*, pages 27–1. 3

[18] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999. 2, 6

[19] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 5

[20] J.-S. Liu and J.-H. Chuang. A geometry-based error estimation for cross-ratios. *Pattern recognition*, 35(1):155–167, 2002. 3

[21] P. Meer, R. Lenz, and S. Ramakrishna. Efficient invariant representations. *International Journal of Computer Vision*, 26(2):137–152, 1998. 3

[22] P. Meer, S. Ramakrishna, and R. Lenz. Correspondence of coplanar features through p2-invariant representations. In *Joint European-US Workshop on Applications of Invariance in Computer Vision*, pages 473–492. Springer, 1993. 3

[23] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 27–36, 2002. 2

[24] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *International Workshop on Document Analysis Systems*, pages 541–552. Springer, 2006. 2

[25] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011. 8

[26] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407. IEEE, 2011. 2

[27] M. Rodrigues. *Invariants for Pattern Recognition and Classification*. Series in machine perception and artificial intelligence. World Scientific Publishing Company Pte Limited, 2000. 3

[28] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustmenta modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 5

[29] H. Uchiyama and H. Saito. Random dot markers. In *2011 IEEE Virtual Reality Conference*, pages 35–38. IEEE, 2011. 1, 2, 3, 4, 7

[30] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE computational science and engineering*, 4(4):10–21, 1997. 2