BlobBoards: Robust Markers for Accurate Pose[1]

James Pritts
jpr@informatik.uni-kiel.de
Hendrik Sauer
hendrik-sauer@mailbox.org
Felix Seegräber
fse@informatik.uni-kiel.de
Till Sittart
till.sittart@till-s.de
Silja Janßen
sja@informatik.uni-kiel.de
David Nakath
dna@cs.uni-kiel.de
Marine Data Science
Kiel University
Kevin Köser
kk@informatik.uni-kiel.de

Abstract

We propose BlobBoards, a fiducial marker system comprising a dense, multi-scale field of Gaussian blobs and a feature-based pipeline for joint detection, identification, and pose estimation. Each board is registered from hundreds of blob features whose dense spatial coverage constrains pose, while multiple scales preserve detectability across large changes in focal length, distance, and obliquity. Learned local descriptors are matched to the reference pattern and spatially verified, so the correspondences determine pose and certify identity. Against motion-capture ground truth, BlobBoards achieve median translation errors of 3.6–5.0 mm, reducing AprilTag's median translation error by 89% on small boards and 70% on large ones. They also produce far fewer large-rotation failures than state-of-the-art tag systems. BlobBoards achieve the highest detection rate, 80% versus 74% for AprilTag and 58% for ArUco, with the largest margin on the smallest markers. Under 50% occlusion, they still detect 69% of boards with essentially unchanged median translation error, while AprilTag and ArUco detect none. In experiments BlobBoards give state-of-the-art detection rate, pose accuracy and occlusion robustness.

Figure 1: All 25 BlobBoards in the image are detected, identified, and registered against a 50-board gallery, so 25 of its entries are absent confusers, and identification is still perfect. Colour distinguishes detections; the overlaid pattern shows registration accuracy across large variations in scale, distance, and obliquity.

[1]Project page, code and data: https://blobboards.github.io

1 Introduction

Fiducial markers are a foundational tool in robotics and computer vision [18, 19], providing reliable references for camera pose estimation and instance identification in applications including robotic manipulation, SLAM, augmented reality, multi-camera calibration, and autonomous navigation. A useful marker should be accurately localizable, robust to viewpoint and partial visibility, and uniquely identifiable so that multiple markers can coexist in a scene.

Two design families dominate, and both designs inherently limit keypoint density. Grid targets such as the checkerboard localize saddle points to sub-pixel accuracy [16, 41], but each saddle needs uniform regions of alternating black and white, and the repeated grid carries no local identity: under partial visibility the correspondence between observed corners and board coordinates becomes ambiguous. Coded fiducials solve identity, but an independently identifiable marker of the ARToolKit [20], ArUco [14, 34] or AprilTag [23, 30, 39] family is a single high-contrast quadrilateral contributing exactly four corner measurements regardless of its size [18, 19]. Measurement density per unit marker area therefore *falls* as the marker grows. Occlusion of the segmented boundary disables the marker outright. Multi-marker calibration boards (AprilGrid [13], ChArUco [5, 14]), like the self-identifying checkerboard variants CALTag [1] and deltille grids [15], supply more measurements over a fixed board area by tiling the target with markers. In that case, the target is treated as a single rigid object and markers are used to place joint constraints on the camera.

BlobBoards are designed to maximize the density of localizable features. A board is a randomly generated field of Gaussian blobs distributed densely over its area and across multiple scales (fig. 1). Measurement density is a property of the pattern rather than of the board size. The markers used in our evaluation carry between 213 and 1027 blobs each over 4 to 12 cm footprints. High spatial density yields a well-constrained pose estimate from many distributed correspondences, while the scale distribution ensures that some subset of blobs remains at a feasible detection scale as focal length, distance and viewpoint change.

A robust affine adaptation localizes each blob to sub-pixel accuracy. Each adapted blob is canonicalized and described by the texture of its surrounding region, whose random spatial configuration is unique to the board. Tentative descriptor matches are spatially verified by registering the board to its image in a RANSAC estimator. Pose estimation therefore identifies and registers the board simultaneously.

Other markers have also left the square. RUNE-Tag [3] arranges dots on concentric rings and CCTag [6] localizes concentric contours to sub-pixel precision; both raise per-marker accuracy but keep a fixed, designed feature set. Closer still are codeless patterns recognized from the configuration of their own features: random dot markers [38] identify a marker from the local arrangement of scattered points, and Li *et al.* [24] build a calibration pattern from descriptor-matched features. BlobBoards keep that codeless principle but replace point dots with affine-covariant Gaussian blobs at many scales, so that each feature is independently localizable under viewpoint change and the feature count grows with printed area.

Evaluated against motion-capture ground truth, BlobBoard's dense feature coverage substantially improves absolute-pose accuracy over AprilTag and ArUco, with the advantage increasing as markers become smaller and more oblique. On the same captures BlobBoards also detect the largest fraction of visible boards, and this detection lead is likewise largest on the smallest markers, so BlobBoards improve on both detection and accuracy at once. Under occlusion, BlobBoard degrades gracefully because occluded regions remove local features rather than disabling the marker.

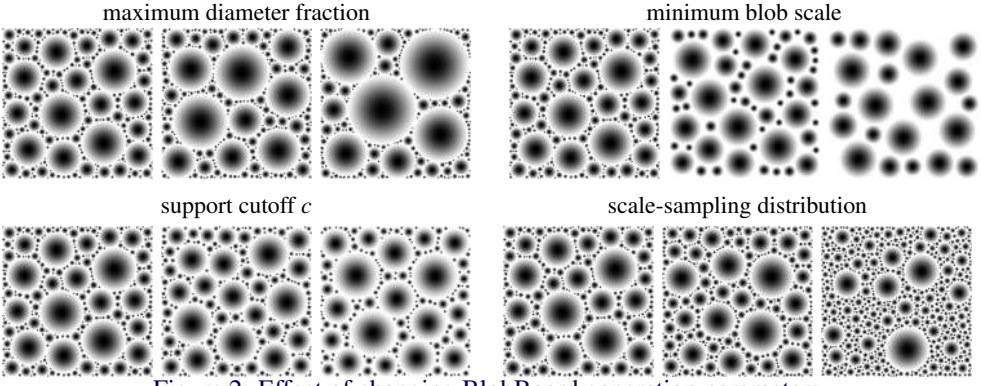


Figure 2: Effect of changing BlobBoard generation parameters.

2 BlobBoards

A BlobBoard is a dense random packing of dark Gaussian blobs at different positions and scales. Each blob of scale σ_0 is truncated at radius $c\sigma_0$ and shifted and normalized to meet the white background continuously:

$$f(\mathbf{x}) = \left(1-\exp\left(-\|\mathbf{x}\|^2/2\sigma_0^2\right)\right) / \left(1-\exp\left(-c^2/2\right)\right), \quad \|\mathbf{x}\| \le c\sigma_0, \quad \mathbf{x}\in\mathbb{R}^2 \tag{1}$$

The compact support allows blobs to be packed densely without overlap, while varying σ_0 makes the pattern intrinsically multi-scale.

Up to the additive white background, an ideal blob is a negative isotropic Gaussian

$$I(\mathbf{x}) = -A\exp\left(-\|\mathbf{x}\|^2/2\sigma_0^2\right)$$

of contrast A. SIFT localizes features as scale-space extrema of the scale-normalized Laplacian-of-Gaussian

$$R(\mathbf{x},\sigma) = \sigma^2\nabla^2(G_\sigma * I)(\mathbf{x}),$$

where G_σ is the isotropic Gaussian kernel of standard deviation σ and $*$ denotes convolution; the Difference-of-Gaussians pyramid approximates this operator [25, 27]. Since the Gaussian family is closed under convolution, the response at the blob centre is

$$R(\mathbf{0},\sigma) \;=\; -2A\,\sigma_0^2\,\sigma^2/(\sigma_0^2+\sigma^2)^2, \qquad \left|R(\mathbf{0},\sigma)\right| \text{ maximal at } \sigma=\sigma_0. \tag{2}$$

The maximum response is $A/2$, independent of σ_0. An ideal Gaussian therefore has two useful properties for a multi-scale target. First, it is scale-covariant: the scale-space extremum occurs at the blob centre and its true scale. Second, its peak response is scale-independent: fine and coarse blobs of equal contrast respond equally strongly. Truncating the Gaussian gives it the compact support required to pack these accurately localized features at high spatial density, which is central to pose accuracy.

The finite-support profile of eq. (1) preserves both properties but introduces a predictable scale bias. Truncation alone perturbs the matched scale only slightly; the dominant effect is the removal of the pedestal $p = e^{-c^2/2}$ that makes the blob meet the background continuously. Carrying this through the centre response gives an extremum at $\beta(c)\,\sigma_0$. We call $\beta(c)$ the *pedestal scale bias*: it depends only on the support cutoff c and not on σ_0, so the response stays scale-covariant with a scale-independent peak, and each detected scale is

rescaled by the *pedestal correction factor* $1/\beta(c)$ to recover σ_0 before the affine adaptation of section 3.1. Supplementary section S9 gives the closed form of the truncated response (eq. (S1)) and of β.

Board generation. We generate each board by greedily packing blobs from a discrete set of scales while keeping their supports disjoint. Blob j is defined by $(\mathbf{c}_j, \sigma_j)$, where $\mathbf{c}_j$ is its centre in the metric coordinates of the printed pattern and σ_j its scale; its support is the disc of radius $c\sigma_j$ (eq. (1)). Scales are processed coarse-to-fine, placing large blobs first and using progressively smaller blobs to fill the remaining gaps. Candidate centres are sampled uniformly over the board and accepted only if their support does not overlap any placed blob. An R-tree accelerates overlap queries, and a placement is abandoned once its fixed trial budget is exhausted. The procedure is seeded for exact reproducibility, and is stated in full as supplementary algorithm S1.

Each accepted blob is then rasterised using eq. (1) with $\sigma_0 = \sigma_j$. Because supports are disjoint, no compositing rule is required. The generator outputs the printable image and the reference geometry $\mathcal{B} = \{(\mathbf{c}_j, \sigma_j)\}_{j=1}^{N}$, which is used for registration. Random placement also makes the neighbourhood of each blob distinctive; this surrounding texture, rather than the Gaussian itself, provides the appearance cue used for identification in section 3.2.

Board generation is controlled by the minimum blob scale, the maximum diameter fraction, the support cutoff c, and the scale-sampling distribution, which trade off density, scale coverage, and printability (fig. 2). Their values depend on working distance, camera resolution, and print resolution; the settings used in our experiments are given in supplementary table S1.

3 The BlobBoard Pipeline

BlobBoards detect, identify, and pose boards by formulating detection as spatial verification [32] over many local features (fig. 3). The pipeline has eight stages, grouped as below: **(1) detect** blob candidates in scale space and initialize an elliptic frame; **(2) adapt** that frame to the local image, recovering the blob's characteristic scales and affine shape $\mathbf{\Sigma}$ (section 3.1); **(3) canonicalize** the surrounding patch by whitening the ellipse and resampling it on a log-polar grid; **(4) describe** the patch with a learned 128-D embedding (section 3.2); **(5) match** image descriptors against a per-board reference gallery; **(6) spatially verify** the resulting correspondences with LO-RANSAC [7, 12, 33], which proposes and locally refines a geometric hypothesis; **(7) certify** the board's identity from the appearance of its verified inliers; and **(8) claim** the detections of each committed board, so that boards competing for the same features are resolved in rounds (section 3.3). A likelihood-ratio test then resolves the two-fold planar pose ambiguity [35] (section 3.4). Supplementary algorithm S2 states the whole pipeline in one place; the remainder of this section details each stage.

3.1 Detect & Adapt

Detect. We detect blob candidates with a GPU implementation of ASIFT [29, 40]: a Difference-of-Gaussians scale space evaluated over a bank of 25 synthetic affine warps — an identity warp plus rotations at two tilt levels spanning the target viewpoint range. Extrema are localized to sub-pixel position $\boldsymbol{\mu}_0$ and continuous scale σ, which the pedestal correction factor $1/\beta(c)$ maps back to the printed scale (section 2). An extremum is isotropic in the warped image; pulled back through the inverse warp it is an ellipse in the input image whose

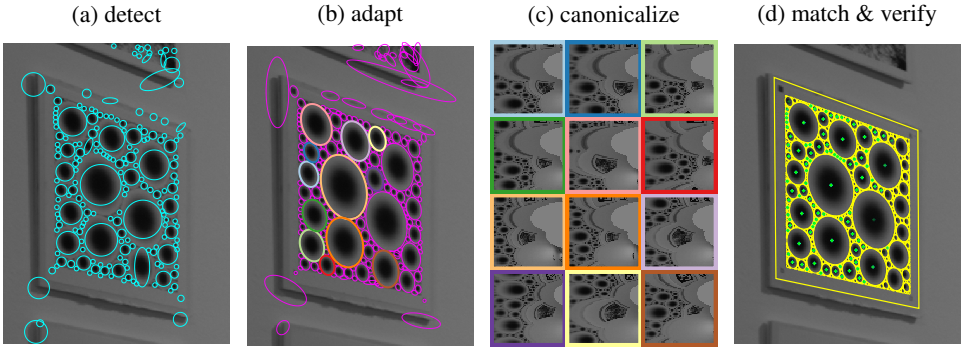


Figure 3: Blob detection-to-pose pipeline on a detected board. **(a)** Affine detections from the ASIFT scale space. **(b)** IRLS adaptation to anisotropic ellipses, magenta except the twelve pose inliers, which are coloured individually. **(c)** Those inliers canonicalized to 64×64 log-polar patches, each bordered in its ellipse colour from (b). **(d)** Descriptors matched and spatially verified: the pattern is projected under the recovered pose, with reference blob centres in green at opacity proportional to their geometric consistency with it.

size is the detected characteristic scale and whose aspect and orientation are the warp's. This ellipse is the seed shape matrix $\mathbf{\Sigma}_0$. A physical blob may fire under several warps, so overlapping ellipses are non-maximally suppressed by response. Detection returns the seed elliptic frame $\mathcal{E}_0 = (\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$ with its response and polarity.

Adapt. The seed $\mathbf{\Sigma}_0$ thus combines the characteristic scale of the detection with the shape of the warp that produced it. It is an initial estimate of the imaged blob's scale and shape. Adaptation refines $(\boldsymbol{\mu}_0, \mathbf{\Sigma}_0)$ against the local pixel data. The refined shape matrix $\mathbf{\Sigma} \succ 0$ encodes the blob's two characteristic scales and its orientation, and its c^2 level set is the elliptic frame used for canonicalization (section 3.2). We model the local intensity as a dark Gaussian on a bright background,

$$I(\mathbf{x}) \approx b - A \exp\left(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \qquad \mathbf{x} = (x,y)^\top, \tag{3}$$

with b estimated from the local maximum intensity in the fitting window, $A > 0$ the contrast and $\boldsymbol{\mu}$ the blob centre.

Fitting eq. (3) directly is nonlinear through the exponential. Writing $f_i = b - I(\mathbf{x}_i)$ for the inverted intensity and $\boldsymbol{a}_i = (1, x_i, y_i, x_i^2, 2x_iy_i, y_i^2)^\top$, taking logarithms over the samples with $f_i > 0$ gives

$$z_i \equiv \log f_i = \boldsymbol{\theta}^\top \boldsymbol{a}_i, \tag{4}$$

where $\boldsymbol{\theta} \in \mathbb{R}^6$ collects the coefficients of the expanded quadratic: the quadratic terms give the upper triangle of $-\frac{1}{2}\mathbf{\Sigma}^{-1}$, the linear terms then give $\mathbf{\Sigma}^{-1}\boldsymbol{\mu}$ and hence $\boldsymbol{\mu}$, and the constant term $\log A - \frac{1}{2}\boldsymbol{\mu}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu}$ then gives A. Adaptation is therefore a quadratic surface fit to the log-intensity profile.

The logarithm makes homoscedastic intensity noise heteroscedastic: to first order, intensity noise of variance ς^2 induces log-space variance ς^2/f_i^2. The corresponding generalized least squares (GLS) weight is therefore proportional to f_i^2 — equivalently, the residuals are whitened by f_i — giving

$$L(\boldsymbol{\theta}) \approx \sum_i \tilde{w}_i \left(\log f_i - \boldsymbol{\theta}^\top \boldsymbol{a}_i\right)^2, \qquad \tilde{w}_i = w_i f_i^2, \tag{5}$$

where f_i^2 is the deterministic GLS weight and w_i is a robust weight from a Student-t/uniform mixture whose inlier partition is re-selected at every iteration by the same exact marginal-likelihood sweep over the sorted residuals [33] — rejecting neighbouring blobs, occluder edges and background clutter with no noise scale to tune, since it is marginalized. Since $f_i^2 = (\partial f/\partial z)^2$, eq. (5) is the first-order intensity-space objective expressed in linearized coordinates. It is solved by M-estimation [4, 17] using iteratively reweighted least squares (IRLS) initialized from $\mathbf{\Sigma}_0$. We batch the solve on the GPU over all detections. Each valid detection returns the refined elliptic frame $\mathcal{E} = (\boldsymbol{\mu}, \mathbf{\Sigma})$.

3.2 Canonicalize & Describe

Canonicalize. To first order, perspective imaging maps the neighbourhood of a planar blob by a local affinity. The adapted shape matrix $\mathbf{\Sigma}$ captures this deformation, and the whitening transform

$$\mathbf{x}' = \mathbf{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu})$$

maps the blob's elliptic frame to the circle $\|\mathbf{x}'\| = c$ [2], removing local affine deformation up to an in-plane rotation. A log-polar resampling [10, 11] then maps this residual rotation to translation along the angular axis.

The Gaussian blob is not discriminative, since every blob has the same radial profile. The unique signal lies in the random configuration of neighboring blobs. We sample an annulus outside the blob in whitened coordinates, where radii are in units of the blob's scale: the inner radius is the support cutoff c and the outer radius the hyperparameter c_{out}, and we resample $r = \|\mathbf{x}'\| \in [c, c_{\text{out}}]$ onto a fixed 64×64 log-polar grid. Whitening and log-polar resampling are combined into a single per-feature warp, yielding an affine-normalized representation of the surrounding board texture.

Describe. Each canonical patch is embedded into a 128-dimensional unit-norm descriptor using a HardNet-style convolutional network [28], following the log-polar descriptor setting of Ebel *et al.* [10]. The network takes the 64×64 patch as input, uses 5×5 convolutions with padding 2, and maps the final feature map to a 128-D descriptor with a dense layer. Because we do not assign a dominant orientation to the blobs, we max-pool along the angular axis after the final convolution. The log-polar transform converts residual rotation and isotropic scaling into translations along the angular and radial axes respectively; angular max pooling makes the descriptor rotation-invariant. Convolutions wrap circularly on the angular axis, which is periodic, and pad at the radial edges, which are not; downsampling is antialiased. Figure 4 gives the full architecture.

Curriculum training. Training minimizes the supervised contrastive loss [21]. Every blob on every board in the dataset is uniquely identified by its board ID and blob index, and each (board ID, , blob index) pair defines a class. At each training stage, we optimize with AdamW [26] using a learning rate of 10^{-4} and a batch size of 4,096, and select models by false-positive rate at 95% recall (FPR95) on the validation set.

We first train for 1000 epochs on synthetic images of 40 dynamically generated boards composited over OpenLORIS [37] backgrounds under random homographies [9], which provide exact board-to-image blob correspondences. Training images are augmented with Gaussian blur, Gaussian noise, and color jitter.

This preliminary network is then used within the RANSAC pipeline to bootstrap labels from 634 indoor and outdoor videos and photographs of 50 boards. For each image, the estimated board homography is used to retain spatially verified blob detections and assign

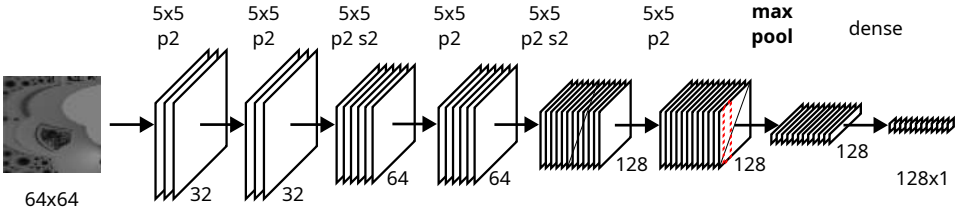


Figure 4: Descriptor architecture, adapted from HardNet [28]: 64×64 log-polar input, 5×5 kernels with padding 2, and column-wise max pooling of the rotation-equivariant feature map to a 1×16 map that a dense layer maps to the 128-D descriptor. Kernel size, padding (p) and non-default stride (s) are shown above each layer; channel count below.

each the class label of its corresponding reference blob; unmatched detections are retained as confusers. The boards are split $45/5$ into training and validation sets, approximately 90:10 by patch count. A second network is trained for 50 epochs on these data and used for one further bootstrap iteration. Random downsampling and homographic augmentation provide additional hard samples. The final network used for evaluation is trained on the resulting third-generation real-image patches.

Reference-pattern patches and their corresponding real observations share the same class label, so the reference-to-observation matches required at test time are explicit positive pairs during training. The batch-sampling procedure is detailed in supplementary section S4.

3.3 Match, Verify, Certify & Claim

Match. For each board, canonical patches are extracted from the reference pattern using the known blob geometry and their descriptors precomputed. This gallery is built once and reused across queries, so it incurs no per-query descriptor cost. Matching remains linear in gallery size: for N image detections and B boards with M blobs each, one $N \times M$ cosine-similarity matrix is computed per board, for $O(NBMD)$ cost with D-dimensional descriptors. Verification fits at most one hypothesis per live board per round, giving $O(BK)$ fits for the K boards ultimately committed. Neither term is quadratic in B.

Detections are matched against each board's gallery independently, by mutual nearest neighbours in cosine similarity above a threshold, which suppresses chance matches between locally similar neighbourhoods across boards. Matching proceeds in two phases: a strict mutual-nearest-neighbour pass proposes boards, and a permissive k-nearest-neighbour pass (k=3) later rematches the detections that remain once boards have been accepted. Each candidate board yields a set of tentative detection-to-blob correspondences with associated descriptor similarities.

Spatial verification. Each candidate correspondence set is verified by RANSAC [12]. Verification is agnostic to the geometric model: any solver registering reference blobs to image detections can be used. The principal cases are P3P [22, 31] for a calibrated camera and a planar homography [16] when the camera is uncalibrated. Each accepted model is refined on its inlier set using the local-optimization step of LO-RANSAC [7], and hypotheses with fewer than $N_{\min}$ final inliers are discarded.

Appearance certification. At BlobBoard densities, geometric consensus alone is insuffi-

cient for reliable identity verification. High-frequency image content, such as a tree, can generate many detections, some of which may by chance be geometrically consistent with a proposed registration to a gallery board. Requiring stronger geometric consensus would suppress these false hypotheses but also reject difficult boards with few usable correspondences. We therefore use a permissive geometric threshold and certify identity from appearance: at least half of the geometric inliers must have cosine similarity ≥ 0.6 to their matched reference blobs. Each inlier already names a reference blob and board, so this requires no additional feature extraction or decoding. False hypotheses have median per-inlier similarity 0.07–0.37, compared with 0.73–0.97 for genuine boards.

Claim. Because multiple boards may compete for the same detections, correspondences are claimed in rounds rather than all at once. In each round, all live candidates are fit, the best-supported board is committed, and detections within its quadrilateral are claimed. The remaining candidates are then rematched one-to-one against the unclaimed detections before the next round. Claiming removes correspondences explained by an accepted board, preventing false hypotheses from borrowing its support, while rematching recovers neighbouring boards from the remaining features. The output is the set of verified (board id, pose) pairs passing the geometry, appearance, and orientation tests.

3.4 Planar pose ambiguity

A planar target admits a two-fold pose ambiguity [35]: two poses with plane normals reflected about the viewing direction can produce nearly identical reprojections. Perspective separates the two branches when the board is large in the image and strongly foreshortened, but the distinction becomes weak near fronto-parallel views or at small projected extent. Appearance cannot resolve this ambiguity either, because the rotation-invariant canonical descriptor can certify essentially the same correspondences under both poses. We therefore construct the competing mirror pose, refine both branches from the same correspondences using identical local optimization, and retain the branch with the higher scale-marginalized inlier/outlier score [33]. The score models inlier reprojection error as Gaussian, marginalizes the unknown noise scale σ^2 in closed form under a Jeffreys prior, and models outliers uniformly. For pose $\boldsymbol{\theta}$ with inlier set I among N correspondences,

$$S(\boldsymbol{\theta}, I) = \log\Gamma\left(\tfrac{\nu}{2}\right) - \tfrac{\nu}{2}\log\left(\pi\sum_{i\in I} r_i^2\right) - d_g\,(N - |I|)\log 2a, \quad \nu = d_g|I| - n_{\boldsymbol{\theta}}, \tag{6}$$

where r_i is the reprojection residual, $d_g = 2$ is the residual dimension per correspondence, $n_{\boldsymbol{\theta}} = 6$ is the number of pose degrees of freedom, and $a = 5$ pixels is the outlier half-width. Because σ^2 is marginalized, the score has no noise scale to tune. For each branch, the maximizing inlier set I is found exactly by sweeping the sorted residuals [33]. Since each branch is independently optimized and scored with its own maximizing inlier set, their score difference is a generalized log-likelihood ratio, of which only the sign is used (table 1). Thus geometry localizes the board, appearance certifies its identity, and the likelihood ratio resolves its orientation.

4 Evaluation Protocol

We compare BlobBoards with AprilTag 3 (*tagStandard41h12*) [23, 30] and OpenCV's ArUco detector [5, 14] with the ArUco 3 *MIP_36h12* dictionary [34], the state-of-the-art

Figure 5: Experimental setup, one capture per marker type. Each type is mounted on a matched rigid panel carrying 30 markers at three physical scales (18 small, 8 medium, 4 large), observed while motion capture tracks the panel's 6-DoF pose. Estimated board frames are overlaid; for BlobBoards the reference pattern is additionally projected under the recovered pose. Further captures are in supplementary fig. S1.

baselines for independently identifiable single markers and the class to which this evaluation is scoped. Both recover pose from their four detected corners on undistorted images with the calibrated intrinsics, using OpenCV's `IPPE_SQUARE` solver [8], which selects between the two planar poses (section 3.4). All patterns are rendered at 1200 dpi on the same printer and rigidly mounted on flat aluminium panels, each carrying 30 markers at three physical scales: 18 small (4 cm), 8 medium (6 cm) and 4 large (12 cm). Pattern area is matched across marker types at each scale, so every method receives the same image area; what differs is the measurement density inside it, a median of 213, 375 and 1027 blobs at the three sizes against four quad corners for a tag of any size (generation settings in supplementary table S1). The 30 BlobBoards are disjoint from the 50 boards used to train the descriptor (section 3.2), so every evaluated blob class is unseen, and no evaluation image was used at any stage of training, validation, or bootstrapping.

A fixed DSLR observes the panel while an infrared motion-capture system tracks rigid marker rigs on both the panel and the camera. Captures are made at fixed camera distances of 1, 2, and 3 m while rotating the panel in nominal 10° increments from $-90°$ to $+90°$ about fronto-parallel; the three marker types are captured at matched distances and viewpoints (see fig. 5). The actual obliquity of every frame is obtained from the calibrated motion-capture pose (section 4.1).

4.1 Ground-Truth Calibration

Motion capture provides the panel-rig-to-camera-rig pose $\mathbf{A}_i$ for each image i. Converting this to the camera frame requires two fixed transforms: the camera-rig-to-optical-frame transform $\mathbf{X}$, shared by all 30 boards, and the board-to-panel-rig transform $\mathbf{Y}_m$, one per board. We estimate these from the board pose $\hat{\mathbf{B}}_{im}$ returned by the marker system:

$$\hat{\mathbf{B}}_{im} = \mathbf{X}\mathbf{A}_i\mathbf{Y}_m,$$

an instance of robot–world/hand–eye calibration [36]. Relative motion between two frames of the same board cancels $\mathbf{Y}_m$, allowing the rotation of $\mathbf{X}$ to be estimated jointly from all boards. Each board rotation then follows from its own frames, after which the translations of $\mathbf{X}$ and all $\mathbf{Y}_m$ are recovered jointly by linear least squares (supplementary section S9). The calibrated pose of each board is then available for every image.

Entangled errors. The panel and camera mounts introduce mechanical error, while the board poses $\hat{\mathbf{B}}_{im}$ used for calibration are themselves estimated by the marker system. Me-

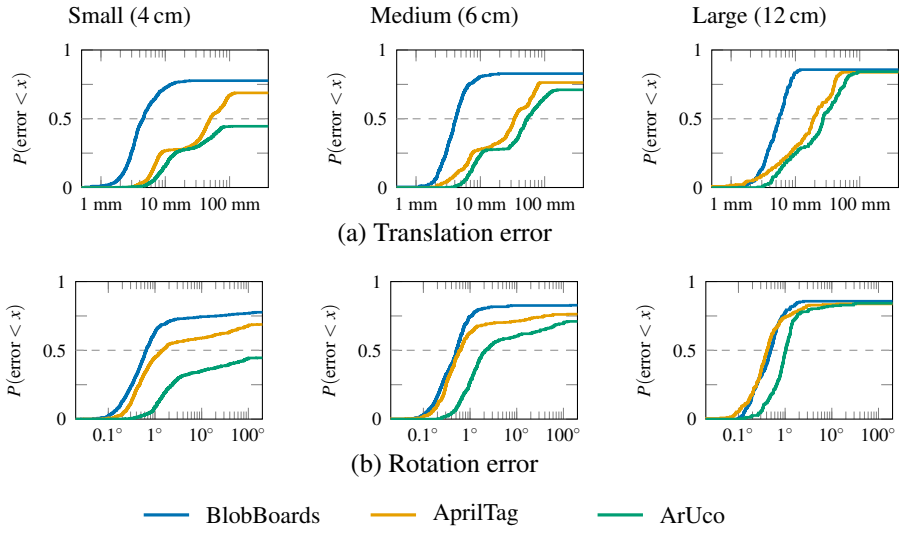


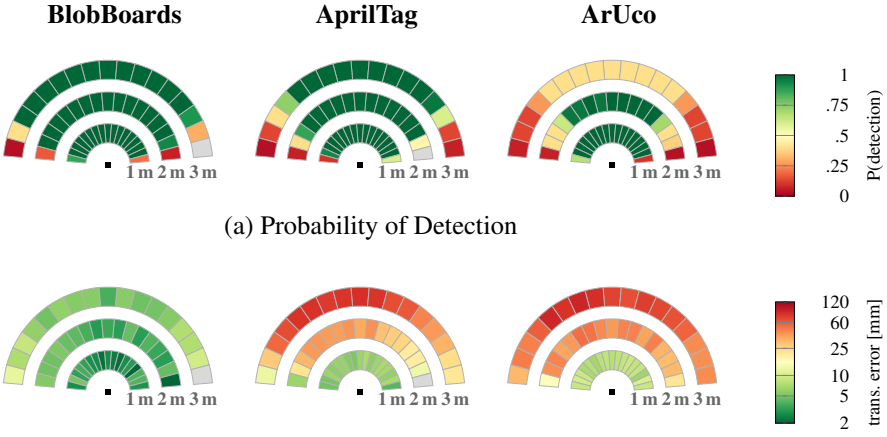
Figure 6: Recall as a function of the error threshold, by marker size. Columns show small (4 cm), medium (6 cm), and large (12 cm) markers; rows show translation and rotation error. A missed detection never enters the numerator, so each curve saturates at the corresponding probability of detection. Dashed lines mark recall 0.5.

chanical and detector errors are therefore entangled in the recovered transforms. To limit this coupling, calibration is performed once per marker system on a set of images disjoint from evaluation, spanning poses easy to recover for both motion tracking system and marker detectors. The fixed mount transforms can therefore absorb only systematic, pose-independent bias, not pose-dependent detector error (supplementary section S8). The calibration residual provides an empirical estimate of the ground-truth uncertainty, and supplementary table S3 reports it for all three marker systems. Translation is well clear of it (0.68 mm against detected-board medians of 3.6–5.0 mm), whereas the median rotation error of 0.46° is within a factor of two. Median rotation differences between the systems are therefore smaller than this ground truth can resolve, and we compare rotation only on the large flip errors of table 1, which are one to two orders of magnitude above the residual.

5 Results

We evaluate probability of detection, absolute pose accuracy, and robustness to occlusion. Translation and rotation are reported separately because they respond differently to marker size and viewpoint. We report two complementary quantities. *Recall* at error threshold x, written $P(\text{error} < x)$, is the fraction of all visible board-viewing attempts that are detected, correctly identified, and posed with error below x. Missed detections therefore count against the method at every finite threshold, preventing a detector from appearing accurate by declining difficult cases. *Summary statistics*, including medians and percentiles, are computed over correctly detected boards only and isolate pose accuracy after successful recovery.

Figure 6 plots $P(\text{error} < x)$ for translation and rotation at each marker size, equivalently the empirical CDF of error over all attempts with failures assigned infinite error; supple-



(a) Probability of Detection

(b) Median Translation Error [mm]

Figure 7: Detection and pose accuracy over the capture floor plan. Concentric rings denote camera distance (1–3 m) and wedges denote nominal 10° viewing-angle stations. Colour encodes (a) probability of detection and (b) median translation error. Grey wedges indicate visited stations with no detections.

mentary fig. S2 pools the three sizes. Each curve saturates at the method's *probability of detection* (PoD), the fraction of attempts that are detected and correctly identified, and hence its maximum recall $\lim_{x\to\infty} P(\text{error} < x)$.

For every correctly identified board, we compare the estimated camera-frame pose with motion-capture ground truth. Translation error is the Euclidean distance between estimated and ground-truth board origins; rotation error is the geodesic angle of the residual rotation. Because marker dimensions and the motion-capture reconstruction are metric, translation error is reported directly in millimetres.

Translation. BlobBoards maintain detected-board median translation errors of 3.6, 3.7, and 5.0 mm for the small, medium, and large boards respectively. In contrast, the tag baselines degrade strongly as the marker shrinks and with increasing viewing distance and obliquity. Relative to AprilTag, BlobBoards reduce translation error by 9.1× on small boards, 6.9× on medium boards, and 3.3× on large boards. Over all attempts, misses included, the 50%-recall translation errors are 4.5, 39.7, and 62.9 mm for BlobBoards, AprilTag, and ArUco. The gain grows as the marker shrinks, as the geometry predicts: a tag supplies four corner measurements whatever its size, whereas a BlobBoard's constraints scale with its blob count and remain well distributed as the projection is foreshortened.

Rotation. BlobBoards also produce the tightest rotation-error distribution. Their median detected-board rotation error is 0.46°, compared with 0.5° for AprilTag and 1.38° for ArUco, and the 50%-recall rotation errors over all attempts are 0.58°, 0.78°, and 12.55°. The difference from AprilTag lies primarily in the tails: AprilTag's 90th-percentile rotation error grows from 1.22° on large boards to 24.74° on small ones, whereas the BlobBoard 90th percentile stays between 0.98° and 1.76°; the typical errors of the two methods are similar; those rotation tails are the planar pose ambiguity of section 3.4.

Detection. Because the panel geometry, board identities, and motion-capture pose are known, every visible marker is an attempt. The evaluation gallery contains the panel's own 30 boards, all present in every capture, so identification is a 1-of-30 assignment without ab-

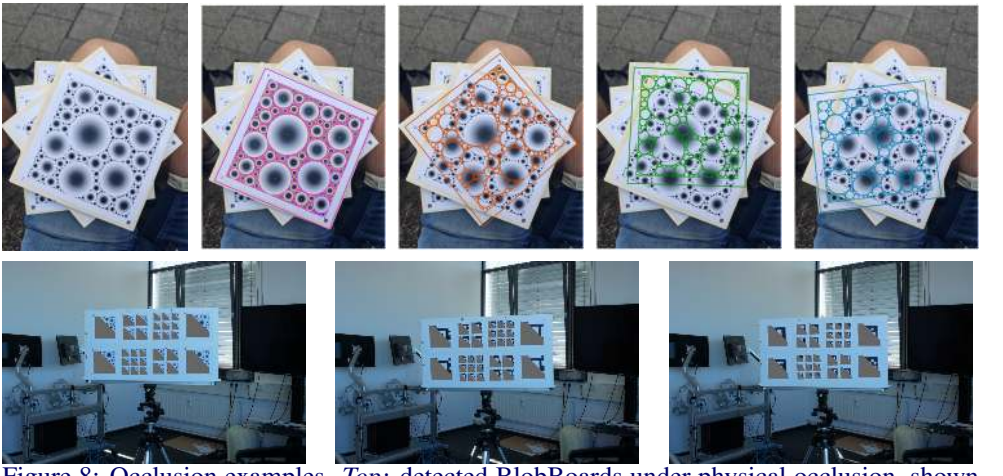


Figure 8: Occlusion examples. *Top:* detected BlobBoards under physical occlusion, shown with recovered board projections and affine-adapted feature ellipses. *Bottom:* the synthetic 70% occlusion mask applied to BlobBoards, AprilTag, and ArUco. Masks are defined in board coordinates and warped using the ground-truth homography, giving matched occlusion across methods and viewpoints.

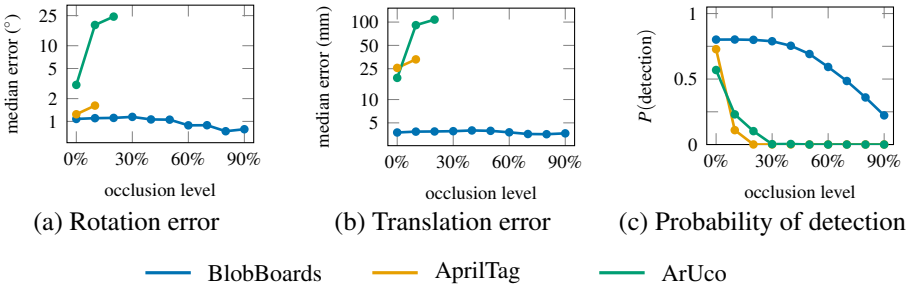
sent distractors. Across 56 captures, BlobBoards returned 1346 poses, each with a unique gallery identity within its capture. Misassignments are directly detectable from the rigid panel layout: any swap would displace a board by at least the 47.8 mm minimum centre spacing, an order of magnitude beyond BlobBoards' median absolute translation error of 4.5 mm, so no consistent pose can survive a swap. Thus, no reported detection is a misidentification, and the appearance certification of section 3.3 admitted no false board.

BlobBoards achieve the highest overall PoD, 80%, compared with 74% for AprilTag and 58% for ArUco. The advantage is largest for the smallest markers, with detection rates of 78%, 69%, and 45%, respectively. All methods degrade toward grazing views, where BlobBoards and AprilTag have comparable coverage at the most extreme angles; fig. 7 shows the station-wise variation, and supplementary fig. S3 the same measurements as marginals against obliquity and distance, with interquartile bands.

For BlobBoards, most remaining failures under severe foreshortening occur after feature detection: blobs are still detected and affine-adapted, but descriptor matching becomes less reliable where real training tracks are sparse. Pose accuracy remains strong for successfully identified boards.

Identification is challenging because each image contains up to 30 markers of the same type, matched jointly against a gallery of visually similar random patterns. Spatial verification resolves local descriptor ambiguities by requiring a common pose, while appearance certification rejects geometrically plausible hypotheses formed from unrelated detections.

Robustness to occlusion. We evaluate occlusion by masking a controlled fraction of each marker in reference coordinates and warping the mask into the image with the ground-truth homography, ensuring the same occluded area across marker types, scales, and viewpoints. Each mask is a deterministic region cut by a single 45° line anchored at a pattern corner and offset to match the target area exactly (fig. 8; construction in supplementary section S12). Because the occluder always crosses the marker border, the tag results characterize sensitivity to border-crossing occlusion rather than an average over occluder placements.



(a) Rotation error (b) Translation error (c) Probability of detection

Figure 9: Performance over the synthetic occlusion sweep for BlobBoards, AprilTag, and ArUco: **(a)** median rotation error, **(b)** median translation error, and **(c)** probability of detection. Medians are omitted when fewer than 20 detections survive at an occlusion level, which is why the tag curves stop early.

The effect is immediate. At only 10% occlusion, AprilTag PoD falls from 74 to 0.11 and ArUco to 0.23, while BlobBoards remain essentially unchanged at 0.8. The surviving tag estimates also degrade sharply: ArUco reaches 18.79° and 91.2 mm, and AprilTag 33.0 mm, compared with 1.1° and 3.9 mm for BlobBoards. From 20% occlusion onward AprilTag reports no detections, and ArUco reaches zero by 30%.

BlobBoards instead degrade gradually. At 50% occlusion they still detect 69% of markers, with median errors of 1.05° and 3.9 mm; Supplementary fig. S5 shows that this coverage extends across much of the capture floor plan, with fig. S4 and fig. S6 giving the 10% and 70% levels of the same series. Some poses are recovered even at 90% occlusion. Moreover, translation accuracy of successful detections remains nearly constant across the sweep, with median error between 3.6 and 4.0 mm (fig. 9). Occlusion removes correspondences but poses remain accurate for BlobBoards.

Planar pose ambiguity. The rotation tails are the two-fold planar pose ambiguity [35] of section 3.4, which table 1 isolates at $> 10°$ rotation error. AprilTag produces 124 such errors and ArUco 152, against 45 for BlobBoards before the mirror test and 33 after — the likelihood-ratio test corrects 12 otherwise identical estimates without changing their detections, correspondences or calibration. The tag baselines already use the state-of-the-art planar solver IPPE [8]; BlobBoards decide the same branch from hundreds of verified correspondences and can therefore still improve on it.

6 Implementation

BlobBoards is implemented in Julia, with affine-warp detection, blob adaptation and descriptor extraction as batched GPU kernels. We report implementation timings rather than a hardware-normalized comparison, since BlobBoards uses a GPU and both baselines run on the CPU. Measured warm on the 12-megapixel evaluation images containing 30 boards, BlobBoards requires 2.03 ± 2.18 s per image on an NVIDIA RTX 4070; AprilTag requires 0.11 ± 0.01 s and ArUco 0.04 ± 0.01 s on the CPU. The cost is dominated by the dense multi-warp scale space, which is computed once per image and shared across all candidate boards, so it is amortized over the 30 boards rather than paid per board.

The baselines are implemented in Python. AprilTag 3 runs through `dt-apriltags` with `quad_decimate` 2 and edge refinement on; ArUco runs through `cv.aruco` with

flips ($> 10°$)	AprilTag	ArUco	BlobBoards	BlobBoards+LRT
1 m	3	21	0	0
2 m	27	96	4	2
3 m	94	35	41	31
total	124	152	45	33

Table 1: Planar-pose flips (rotation error $> 10°$ against motion-capture ground truth) by tripod distance. IPPE selects the lower-reprojection-error tag pose from four corners, whereas section 3.4 selects between BlobBoard branches using hundreds of verified correspondences. The final two columns use identical detections, correspondences, and calibration; the likelihood-ratio test (LRT) reduces the remaining flips from 45 to 33.

default detector parameters. Both are posed by `cv.solvePnP` with `SOLVEPNP_IPPE_SQUARE`. All RANSAC fits are seeded per board, so every figure and number here regenerates bit-exactly from the published dataset record and the released configuration; supplementary table S2 lists the detection, matching and verification settings, and the evaluation scripts are released with the dataset.

7 Conclusion

BlobBoards cast marker detection and identification as feature matching, spatial verification, and appearance certification rather than quad segmentation, making measurement density a property of the printed pattern rather than marker size. Gaussian blobs provide scale-covariant localization with scale-independent peak response and admit a closed-form correction for finite-support bias. Each blob is affine-adapted by a robust fit, while the two-fold planar pose ambiguity is resolved by a scale-marginalized likelihood test.

Against motion-capture ground truth, BlobBoards outperform state-of-the-art tag systems in detection rate and translation accuracy while substantially reducing the large-rotation failures caused by planar pose ambiguity. The gains are largest for the smallest, most oblique, and most occluded boards, where conventional tags are most fragile. Because a BlobBoard is constrained by hundreds of features distributed across its printed area rather than four corners, loss of visible area removes measurements progressively, allowing identity and pose estimation to degrade gracefully rather than fail abruptly.

Acknowledgements. James Pritts is funded by Kiel Training for Excellence, an EU Horizon Europe programme under the Marie Skłodowska-Curie Actions (MSCA), grant agreement No 101081480. The motion-capture setup was funded through the project *OP der Zukunft* as part of the Recovery Assistance for Cohesion and the Territories of Europe (REACT-EU) program and provided by the Kurt Semm Centre for laparoscopic and robot assisted surgery at the University Hospital of Schleswig-Holstein.

References

[1] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. CALTag: High precision fiducial markers for camera calibration. In *Vision, Modeling and Visualization (VMV)*, pages 41–48, 2010.

[2] Adam Baumberg. Reliable feature matching across widely separated views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 774–781, 2000.

[3] Filippo Bergamasco, Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–120, 2011.

[4] Michael Bosse, Gabriel Agamennoni, and Igor Gilitschenski. Robust estimation and applications in robotics. *Foundations and Trends in Robotics*, 4(4):225–269, 2016.

[5] Gary Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Lilian Calvet, Pierre Gurdjos, Carsten Griwodz, and Simone Gasparini. Detection and accurate localization of circular fiducials under highly challenging conditions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 562–570, 2016.

[7] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized RANSAC. In *DAGM Symposium on Pattern Recognition*, pages 236–243, 2003.

[8] Toby Collins and Adrien Bartoli. Infinitesimal plane-based pose estimation. *International Journal of Computer Vision*, 109(3):252–286, 2014.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 224–236, 2018.

[10] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond Cartesian representations for local descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, pages 253–262, 2019.

[11] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[12] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[13] Paul Furgale, Joern Rehder, and Roland Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1280–1286, 2013.

[14] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[15] Hyowon Ha, Michal Perdoch, Hatem Alismail, In So Kweon, and Yaser Sheikh. Deltille grids for geometric camera calibration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5354–5362, 2017.

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.

[17] Peter J. Huber. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, 1981.

[18] Michail Kalaitzakis, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Experimental comparison of fiducial markers for pose estimation. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 781–789, 2020.

[19] Michail Kalaitzakis, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Fiducial markers for pose estimation: Overview, applications and experimental comparison of the ARTag, AprilTag, ArUco and STag markers. *Journal of Intelligent & Robotic Systems*, 101(4):71, 2021.

[20] Hirokazu Kato and Mark Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *IEEE and ACM International Workshop on Augmented Reality (IWAR)*, pages 85–94, 1999.

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 18661–18673, 2020.

[22] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2969–2976, 2011.

[23] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1898–1903, 2019.

[24] Bo Li, Lionel Heng, Kevin Köser, and Marc Pollefeys. A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1301–1307, 2013.

[25] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[27] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[28] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenović, and Jiří Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4826–4837, 2017.

[29] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

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

[31] Mikael Persson and Klas Nordberg. Lambda twist: An accurate fast robust perspective three point (P3P) solver. In *European Conference on Computer Vision (ECCV)*, pages 318–332, 2018.

[32] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[33] James Pritts, Felix Seegräber, and Kevin Köser. RANSAC scoring done right. *arXiv preprint arXiv:2606.27385*, 2026.

[34] Francisco J. Romero-Ramírez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76: 38–47, 2018.

[35] Gerald Schweighofer and Axel Pinz. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2024–2030, 2006.

[36] Mili Shah. Solving the robot-world/hand-eye calibration problem using the Kronecker product. *Journal of Mechanisms and Robotics*, 5(3):031007, 2013.

[37] Xuesong Shi, Dongjiang Li, Pengpeng Zhao, et al. Are we ready for service robots? The OpenLORIS-Scene datasets for lifelong SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3139–3145, 2020.

[38] Hideaki Uchiyama and Hideo Saito. Random dot markers. In *IEEE Virtual Reality Conference (VR)*, pages 271–272, 2011.

[39] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198, 2016.

[40] Guoshen Yu and Jean-Michel Morel. ASIFT: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011.

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

Supplementary Material

The supplementary material of the paper follows in full. Its sections, figures, tables, equations and algorithms carry the S prefix under which the paper cites them.

S1 Board Generation

Algorithm S1 states the placement procedure. Blobs are placed largest-first by rejection sampling with disjoint supports; the coarse-to-fine order matters, because placing small blobs first leaves no room for large ones and yields a substantially sparser board.

Algorithm S1 Greedy coarse-to-fine board generation: blobs are placed largest-first by rejection sampling with disjoint supports, then rasterised in a single pass.

Require: board size $W \times H$; scale schedule $\sigma_1 > \sigma_2 > \cdots > \sigma_n$ with target counts m_i; seed
Ensure: pattern image P; blob set $\mathcal{B} = \{(\mathbf{c}_j, \sigma_j)\}$

$\mathcal{T} \leftarrow$ empty R-tree of placed supports; $\mathcal{B} \leftarrow \varnothing$
for $i = 1, \ldots, n$ **(coarse-to-fine)** and $k = 1, \ldots, m_i$ **do** ▷ largest blobs first
  repeat
    sample centre $\mathbf{c}$ uniformly within the inset board
    if support disc of $(\mathbf{c}, \sigma_i)$ overlaps no blob in $\mathcal{T}$ **then**
      insert $(\mathbf{c}, \sigma_i)$ into $\mathcal{T}$ and $\mathcal{B}$
    end if
  until placed **or** trial budget exhausted ▷ skip if no free spot found
end for
$P \leftarrow \mathbf{1}_{H \times W}$ ▷ white background: the $\|\mathbf{x}\| \to c\sigma$ limit of the blob profile
rasterise every blob in $\mathcal{B}$ over its support ▷ disjoint supports $\Rightarrow$ single pass
return P, $\mathcal{B}$

S2 Board Generation Settings

Table S1 lists the pattern parameters used to generate every BlobBoard in the evaluation. All three physical sizes share one parameter set; only the pattern extent differs, so the number of blobs a board carries grows with its area. A tag of the same physical size contributes four corners at every size.

	small	medium	large
pattern extent (mm)	40	60	120
blobs per board (median)	213	375	1027
shared across all sizes			
minimum blob scale $s_{\min}$		0.2 mm	
maximum diameter fraction d_f		0.5	
support cutoff c		2	
scale-sampling distribution α		1.6	
scales per octave		3	
print density		1200 dpi	
polarity		dark	

Table S1: Pattern-generation settings for the evaluation boards (18 small, 8 medium, 4 large). Each board is generated from its own random seed, so blob counts vary slightly within a size class; the ranges are 202–226, 365–390, and 1001–1032.

S3 Finite-Support Scale Correction

Truncating the Gaussian at c and removing the pedestal $p = e^{-c^2/2}$ perturbs the scale-space response. Writing $\rho = \sigma/\sigma_0$, $\kappa = c^2(1+\rho^2)/2\rho^2$ and $q = e^{-\kappa}$, the centre response of the truncated profile is

$$R(\mathbf{0},\sigma) \;=\; \frac{A}{1-p}\left[\frac{2\big(1-(1+\kappa)q\big)}{(1+\rho^2)^2} \;-\; \frac{2(1-q)}{1+\rho^2} \;+\; \frac{c^2 q}{\rho^2}\right], \tag{S1}$$

which recovers the ideal Laplacian-of-Gaussian response $-2A\sigma_0^2\sigma^2/(\sigma_0^2+\sigma^2)^2$ as $c \to \infty$. Its extremum lies at $\beta(c)\,\sigma_0$, where $\beta(c)$ is the pedestal scale bias of the main paper, $\beta(c) = \arg\max_\rho |R(\mathbf{0},\rho\sigma_0)|$. Because β depends only on c, the truncated response is still scale-covariant with a peak magnitude independent of σ_0, so dividing each detected scale by $\beta(c)$ recovers σ_0 exactly under this continuous model.

S4 Descriptor Training: Reference and Observation Patches

At test time a query patch cropped from an image is matched against a descriptor computed from the reference pattern, so the training objective must contain that pairing explicitly. It does. For every board, reference patches are rendered from the reference pattern and the known blob geometry, and are written into the same patch tensor as the real observations, keyed by the same (board hash, blob index) identity. A reference patch is therefore an ordinary member of its supervised-contrastive class: it serves as an anchor whose positives are the real observations of that blob, and as a positive when a real observation is the anchor.

The batch sampler ensures that reference patches are available for real patches by concatenating the patches for all tracks sharing a blob id including the reference patch and treating them as a unit. Unmatched detections enter each batch as confusers with unique singleton labels: they never act as anchors and never supply positives, appearing only in other anchors' negative denominators. A batch is made up of $1/2$ continuous sequences, and $1/2$ confusers.

S5 Detection, Matching and Verification Settings

Table S2 lists every operational parameter of the pipeline as run for the results in this paper. These are transcribed from the run configuration shipped with the released dataset record, which each driver copies beside its output, so a run's artifacts always carry the configuration that produced them.

stage	parameter	value
detect	affine warps	IMAS-25: identity, plus tilt 2.62 at 8 angles $\pi k/8$ and tilt 5.18 at 16 angles $\pi k/16$
	pre-upsampling	$2\times$ (scale space starts at octave -1)
	DoG peak threshold	0.01
	DoG edge threshold	5.0
	cross-warp NMS	elliptical, Mahalanobis cutoff 2.0
	detections kept	unbounded (no response-rank cap)
adapt	scan window	$2.0\,\sigma$
	max IRLS iterations	20
	convergence tolerance	10^{-6} relative
	storage / accumulation	`Float32` / `Float64`
	validity guards	scale $\leq 0.1\sqrt{WH}$, scale ratio ≤ 4, clip ratio ≤ 0.2, aspect ≤ 10
canonicalize	log-polar grid	64×64
	inner radius	$c = 2.0$
	outer radius c_{out}	96 (in units of σ)
match	descriptor similarity threshold	0.6 (cosine)
	first pass	mutual nearest neighbour
	second pass	k-nearest neighbour, $k = 3$
verify	geometric model	P3P (calibrated)
	N_{min} inliers	9 (inclusive)
	inlier half-width a	5 px
	RANSAC iterations	100 minimum, 10000 maximum
	RANSAC confidence	0.99
	local optimization	Student-t/uniform IRLS, seeded per board
	appearance certification	$\geq 50\%$ of inliers at cosine similarity ≥ 0.6
	mirror branch test	enabled (sign of the score difference only)

Table S2: Operational settings for the reported runs. The RANSAC seed is fixed and each board's stream is derived from it, so the fit stage is bit-reproducible given the extracted features.

S6 Pipeline

The pipeline of the main paper in full, in the order the stages run. Stages (1)–(4) are computed once per image and shared across all candidate boards; (5)–(8) run per board and compete for the same detections.

Algorithm S2 Detection, identification and pose for a gallery of boards. Stages (1)–(4) run once per image; (5)–(8) run per board and compete for the same detections, so committing a board claims its detections before the next fit.

Require: image I; gallery $\mathcal{G}$ of boards with reference descriptors and geometry $\mathcal{B}$; camera (optional)
Ensure: accepted (board id, pose) pairs

$\mathcal{F} \leftarrow$ DETECTADAPTDESCRIBE(I) ▷ (1)–(4), once per image
$C \leftarrow \varnothing$ ▷ claimed detections
for all $(\text{strategy}, s_{\min}) \in \{(\text{mutual NN}, s),\ (k\text{NN}, s')\}$ **do** ▷ (5) strict, then permissive
  $M \leftarrow$ MATCH($\mathcal{G},\ \mathcal{F} \setminus C,\ \text{strategy},\ s_{\min}$)
  while $\mathcal{G} \neq \varnothing$ **do**
    $A \leftarrow \varnothing$ ▷ this round's admissible candidates
    for all $g \in \mathcal{G}$ **do**
      $M_g \leftarrow$ RESOLVEONETOONE($M_g \setminus C$) ▷ re-resolved against the current claims
      if $|M_g| < N_{\min}$ **then continue** ▷ match floor
      $(\boldsymbol{\theta}, I_g) \leftarrow$ LORANSAC(M_g) ▷ (6) spatial verification
      $\boldsymbol{\theta} \leftarrow$ RESOLVEMIRROR($\boldsymbol{\theta}, I_g$) ▷ planar pose ambiguity
      if $|I_g| < N_{\min}$ **then continue**
      if APPEARANCESUPPORT(I_g) $< \frac{1}{2}$ **then continue** ▷ (7) certification
      $A \leftarrow A \cup \{(g, \boldsymbol{\theta}, I_g)\}$
    end for
    if $A = \varnothing$ **then break**
    $(g, \boldsymbol{\theta}, I_g) \leftarrow \arg\max_A (|I_g|,\ \text{score})$ ▷ commit the best-supported board
    $C \leftarrow C \cup \{\text{detections inside the quad of } g \text{ under } \boldsymbol{\theta}\}$ ▷ (8) claim
    accept $(g, \boldsymbol{\theta})$; $\mathcal{G} \leftarrow \mathcal{G} \setminus \{g\}$
  end while
end for
return accepted pairs

S7 Experimental Setup

Figure S1 shows three further captures per marker type, beyond the single capture per type in the main paper. All three panels carry 30 markers at three physical scales and are observed from the same stations, so the rows are directly comparable; estimated board frames are overlaid, and for BlobBoards the reference pattern is additionally projected under the recovered pose.

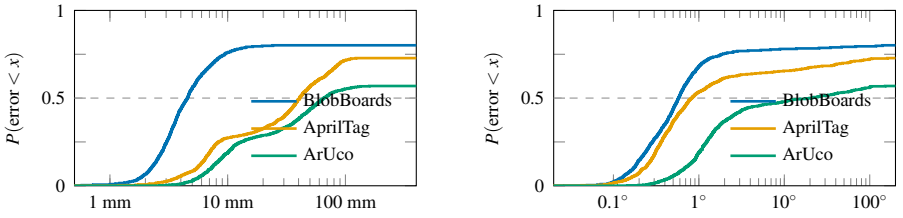
Figure S1: Experimental setup. BlobBoards, AprilTag, and ArUco are mounted on matched rigid panels containing 30 markers at three physical scales: 18 small, 8 medium, and 4 large. The panel is observed at varying distances and obliquities while its 6-DoF pose is tracked by motion capture. Estimated board coordinate frames are overlaid; for BlobBoards, the reference pattern is additionally projected under the recovered pose.

S8 Hand–Eye Calibration Diagnostics

system	frames	recovered	resid. mm	resid. deg
BlobBoards	41	37	0.68	0.270
AprilTag	22	0	2.54	0.432
ArUco	15	0	2.81	1.885

Table S3: Hand–eye calibration diagnostics, one fit per marker system over all 30 boards. *Residual* is the noise scale of the pose residual the staged linear fit reports, in millimetres and degrees; *recovered* counts calibration frames whose untracked camera pose was substituted. The camera is static; its rig was untracked in 37 of the 41 BlobBoard calibration frames, which use a substituted constant camera pose admitted only under an explicit constancy gate on the motion-capture reconstruction and the image background.

Independence of the ground truth. Exactly two observation streams enter the calibration: the motion-capture rig poses $\mathbf{A}_i$ and the board poses $\hat{\mathbf{B}}_{im}$ reported by the marker system under test (notation of the main paper). Three properties bound what the fitted mounts can absorb. First, an explicit calibration set is used, disjoint from every evaluation frame (41, 22 and 15 usable calibration frames for BlobBoards, AprilTag and ArUco, against 56, 57 and 57 evaluation frames), so no evaluation frame contributes to the transforms it is later scored against. Second, the calibration is fitted once per marker system, from that system's own detections, and then held fixed: every system is scored against ground truth from its own



(a) Translation error (b) Rotation error

Figure S2: Recall as a function of the translation and rotation error threshold, pooled over all marker sizes. Every visible board constitutes an attempt; a missed detection never enters the numerator, so each curve saturates at the corresponding probability of detection. Dashed lines mark recall 0.5.

detector and never a competitor's, and since the three panels are physically distinct objects a shared set of mounts does not exist in any case. Third, the fit is heavily overdetermined and spans the same distances and obliquities as the evaluation (section S9), so a rigid mount can absorb only an error that is constant over that span. A constant per-system bias is absorbed for each system alike; scale-, distance- and obliquity-dependent degradation survives into the numbers the main paper reports. The residual noise scale of the BlobBoard fit is 0.68 mm and 0.27°, well below the pose differences the evaluation must resolve (table S3).

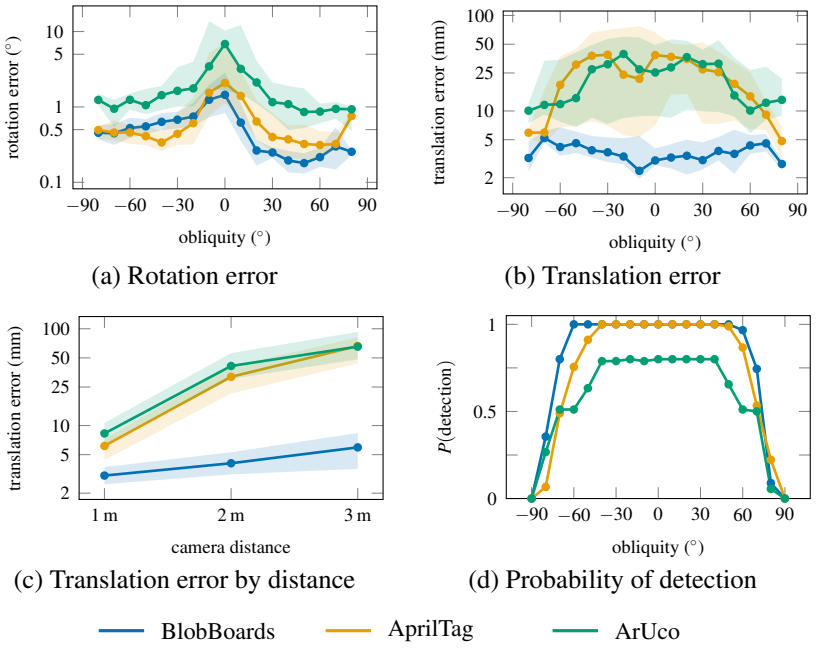
S9 Hand–Eye Estimator

Conditioning of the fit. The calibration is heavily overdetermined: up to 30 boards observed in each of 15–41 calibration frames constrain $6 + 6 \times 30$ parameters, and each board is seen over the same span of distances and obliquities as in the evaluation. Only a detector error that is constant over that span is indistinguishable from a rigid mount error and therefore absorbable; any scale-, distance- or obliquity-dependent component is not.

Staged hand–eye solve. Each of the three stages of the main paper is in closed form: the shared rotation of $\mathbf{X}$ by Kabsch alignment of the rotation axes of the relative motions pooled over all markers, each mount rotation as the chordal mean of its per-frame estimates, and all translations by one joint least-squares solve. The estimator is deliberately linear: a pose-residual nonlinear refinement was implemented and discarded, because it reduced in-sample cost by the amount its own quadratic model predicts yet extrapolated worse across the evaluation range, the signature of model misspecification absorbing an unmodelled systematic along a weakly constrained direction.

S10 Pooled Error Distributions

The main paper breaks recall down by marker size, because that is where the methods separate. Pooling the three sizes gives the aggregate view below, in which the tag curves flatten early where the two-fold planar pose ambiguity takes over.



(a) Rotation error

(b) Translation error

(c) Translation error by distance

(d) Probability of detection

Figure S3: Unoccluded performance binned by obliquity and camera distance. Curves are per-bin medians, bands the interquartile range. Bins with fewer than five detections are omitted.

S11 Unoccluded Performance by Angle and Distance

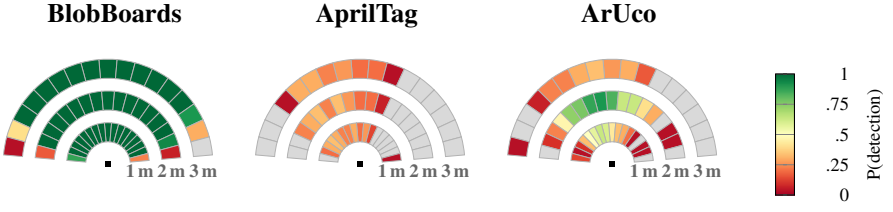
The floor-plan figure of the main paper reports the unoccluded result over the capture floor plan, with distance and viewing angle on the two axes of a single plan and the per-station median as colour. The figure below is the same measurements viewed as marginals: each quantity against obliquity, and translation additionally against camera distance. It adds what a per-station median colour cannot show — the interquartile band, and so the spread of the error at each station, not only its centre.

S12 Occlusion Examples

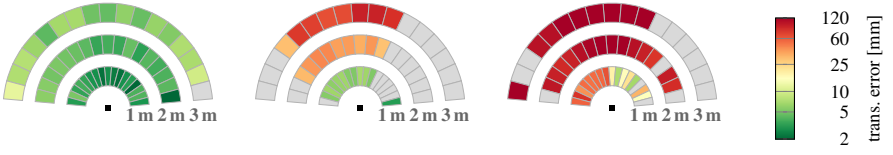
The synthetic occluder is one deterministic region per marker, constructed in the pattern's own coordinate frame and warped into the image through the ground-truth pose, so the same area fraction f is hidden on every marker type at every scale and viewpoint. Its boundary is a single 45° line anchored at the lower-left corner of the pattern. For $f \leq \frac{1}{2}$ the masked region is the right triangle with legs $\sqrt{2f}$ of the pattern side; a triangle cannot exceed half of a square, so for $f > \frac{1}{2}$ the masked region is instead the complement of the right triangle of area $(1-f)$ at the opposite corner. Either way the boundary crosses two edges of the marker, which is what breaks a quad-based detector. The realized covered fraction matches the target exactly at every level of the sweep. We use this one placement rather than averaging over occluder positions.

S13 Occlusion by Floor-Plan Level

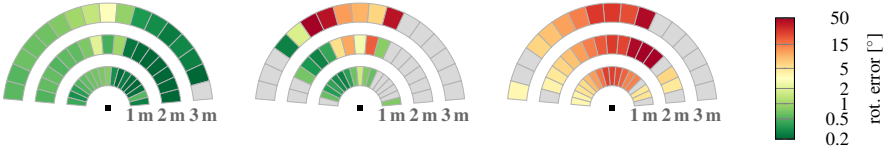
The main paper quotes the 50% level in prose and plots the sweep over all ten levels. The plans below repeat the unoccluded floor plan's encoding at 10%, 50% and 70%, so the loss of coverage can be read per station rather than only in aggregate. The 10% level locates where the two designs part: an occluder crossing the marker border breaks the single quad AprilTag and ArUco depend on, while BlobBoards lose only the blobs the occluder covers. By 50% neither tag baseline returns poses anywhere on the floor, and BlobBoard coverage is not confined to the near, frontal stations where the boards are largest.



(a) Probability of Detection

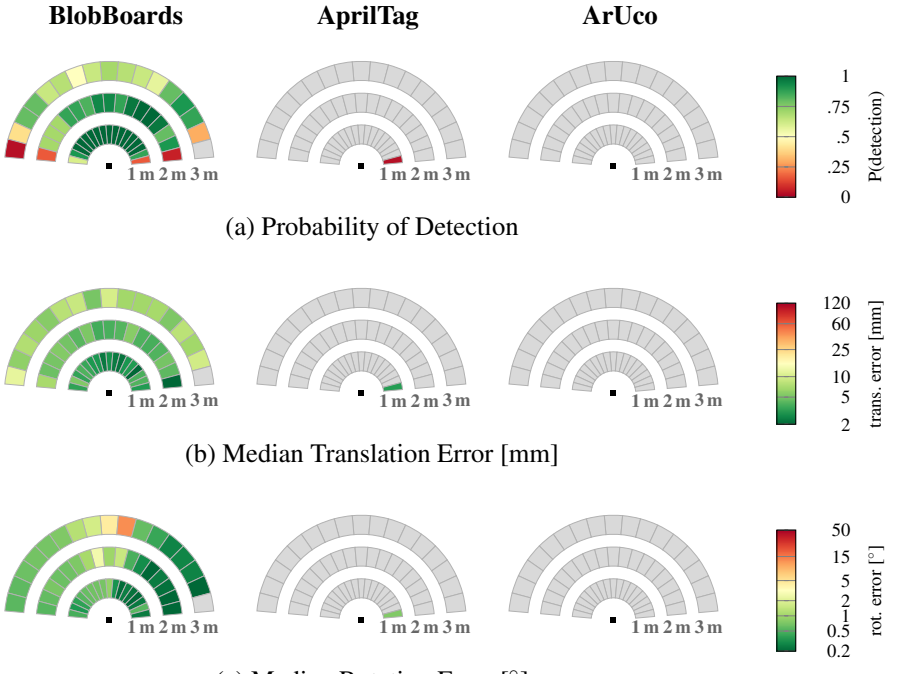


(b) Median Translation Error [mm]



(c) Median Rotation Error [°]

Figure S4: Pose accuracy over the capture floor plan at 10% occlusion, in the same layout and encoding as the floor-plan figure of the main paper. Rings are tripod distance, wedges are 10° rig stations; colour is (a) the fraction of the 30 boards detected, and the per-station median over detected boards of (b) translation and (c) rotation error. **Gray wedges are stations with zero detections.** At this level the AprilTag and ArUco plans are already largely gray while the BlobBoards plan is intact — the separation between the two designs is established well before the heavier levels shown below.

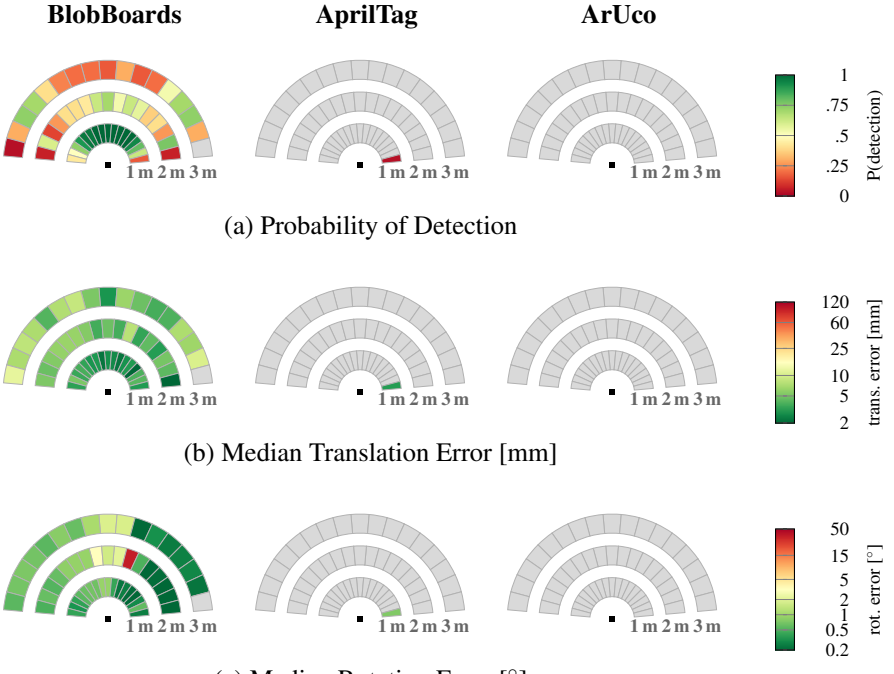


(a) Probability of Detection

(b) Median Translation Error [mm]

(c) Median Rotation Error [°]

Figure S5: Pose accuracy over the capture floor plan at 50% occlusion, in the same layout and encoding as the floor-plan figure of the main paper. Rings are tripod distance, wedges are 10° rig stations; colour is (a) the fraction of the 30 boards detected, and the per-station median over detected boards of (b) translation and (c) rotation error. **Gray wedges are stations with zero detections.** Both tag plans are uniformly gray; the BlobBoards plan retains coverage across the capture floor at per-station errors in the range it holds unoccluded.



(a) Probability of Detection

(b) Median Translation Error [mm]

(c) Median Rotation Error [°]

Figure S6: Pose accuracy over the capture floor plan at 70% occlusion, in the same layout and encoding as the floor-plan figure of the main paper. Rings are tripod distance, wedges are 10° rig stations; colour is (a) the fraction of the 30 boards detected, and the per-station median over detected boards of (b) translation and (c) rotation error. **Gray wedges are stations with zero detections.** The AprilTag and ArUco plans are gray everywhere: at this level neither returns a pose anywhere on the capture floor. BlobBoards retain coverage over a reduced but still substantial set of stations, at per-station errors in the same range as the unoccluded case.