2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Attention-based Markerless Pose Estimation of the Assistant-Port Trocar in Robot-Assisted Surgery with a Head-Mounted Display

Nicholas Greene† Aoqi Long† Peter Kazanzides

***Abstract*— In robotic-assisted minimally invasive surgery, an assistant surgeon stands at the bedside to insert and manipulate instruments while the primary surgeon operates the robot. Augmented reality (AR) head-mounted displays (HMDs) may improve the assistant's spatial awareness, but require tracking of surgical tools (both robotic and hand-held) for accurate overlay. In this work, we propose a markerless method to estimate the 6-DoF trocar pose for the assistant port, which can convey the insertion trajectory of any handheld instrument to the assistant surgeon. The method is based on a deep U-Net architecture with cross-attention and Atrous Spatial Pyramid Pooling (ASPP) to predict 2D keypoints on the trocar, which are then used by a Perspective-n-Point (PnP) method to estimate the trocar's pose. From the predicted trocar pose, we can also directly find the 4-DoF shaft-line of the handheld instrument using a multi-view method; this enables correction for misalignment of the trocar and instrument shaft. The trocar tracking runs in real-time (66 Hz) and can be integrated into an AR-assisted workflow. Experimental results with a phantom show an accuracy of ∼5.5 mm and angle error of ∼1.9 degrees, which is sufficient to guide instrument insertion into the endoscope field of view.**

I. Introduction

Robot-assisted minimally-invasive surgery couples a tele-operated robot with a human surgeon at a console, offering benefits like enhanced dexterity and depth perception [1]. A sterile *assistant surgeon* at the patient's bedside is required to perform tasks such as instrument insertion, retraction, and suction [2], often through a dedicated, non-robotic port (trocar). The assistant's performance is critical to surgical outcomes [3], [4], yet the assistant lacks the immersive visual feedback of the console surgeon and often works under suboptimal viewing conditions [5]. For example, instrument insertion is typically performed without direct endoscopic visualization, and has been shown to generate forces sufficient to damage intra-abdominal organs [6].

We previously proposed *ARssist* [7], an augmented reality (AR) application intended to improve the assistant's situational awareness. *ARssist* uses an optical see-through HMD to provide *in situ* augmented reality overlays of the robot and handheld instrument shafts in the assistant's field of view. Fiducial markers attached to the handheld instrument and to the surgical robot's cannulas enabled tracking of the instrument poses relative to the HMD. The *ARssist* AR overlays demonstrated improved instrument navigation for novices compared to no AR [8].

A major limitation of *ARssist* is the requirement for attached markers, which are not straightforward to implement in a surgical setting. Sterilization, placement, and registration of markers may disrupt established surgical workflows and complicate or prolong procedures. An alternative to sterilizing and placing markers would be to introduce new sterile devices which would add significant overhead for design, regulatory approval, and inventory. This limitation motivates the development of markerless tracking methods for localizing the instruments in the surgical scene with respect to the HMD.

Dept. of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA (email: {ngreen29, along51, pkaz}@jhu.edu)
†Nicholas Greene and Aoqi Long contributed equally to this work.

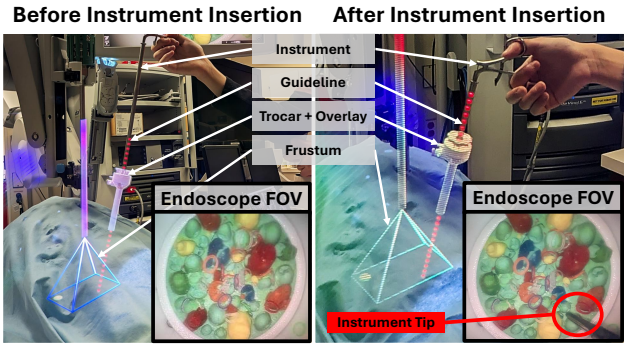

Fig. 1. Images captured through the HoloLens 2 display showing an AR guidance application based on our markerless trocar tracking method. The robotic endoscope is tracked via an attached ArUco marker (not shown) and has an overlay showing its shaft and camera frustum. The trocar is tracked using our markerless method (without shaft detection). A dashed 3D guideline along the trocar's center shows the direction of insertion in relation to the endoscope frustum. In the right image, the instrument is inserted until just visible in the endoscope FOV, as seen on the monitor.

Recently, we achieved markerless tracking of *robotic instruments* by detecting feature points in the HMD camera images of the robotic instrument housing and using Perspective-n-Point (PnP) to predict the pose [9]. The pose was refined by finding the instrument shaft in a multi-view geometry error minimization. This method attained sub-5 mm tip accuracy, which was comparable to a marker-based baseline method. However, this work focused on the robotic instruments and not handheld instruments.

In this paper, we tackle the challenge of markerless tracking of the assistant surgeon's trocar in real-time in order to estimate the insertion trajectory for any handheld instrument, as shown in Fig. 1. The advantage of this approach is that we need only track one object (the trocar), rather than any possible instrument that could be inserted through that trocar.

The contributions of this work are:

1) A **markerless 6-DoF pose estimation method** for tracking a laparoscopic trocar using a HoloLens 2 (HL2) HMD.
2) A **real-time 2D object keypoint detection network architecture** based on U-Net with cross-attention and ASPP and a task-specific composite loss for robust single-image keypoint predictions.

3) A **three-stage training strategy** combining extensive synthetic data and real annotated data to achieve high accuracy despite imperfectly labeled real data.
4) Experimental evaluation of the accuracy of our method for its intended use of providing a shaft direction that can be used for AR guidance in robot-assisted surgery.

Our method lays the groundwork for a clinically realistic, comprehensive AR system to assist the surgical assistant without disrupting the current robotic surgery workflow.

II. Related Work

Early augmented reality systems for the surgical assistant in robot-assisted surgery, such as ARssist [7], [8], relied on fiducial markers to track instruments. Other works have explored using other types of markers for HMD tracking. For example, Martin-Gomez et al. developed STTAR [10], which uses the HL2's IR camera to track passive markers. They achieved sub-millimeter tool tracking accuracy in a K-wire insertion phantom experiment, and demonstrated that built-in HMD sensors can rival optical tracking for small tools.

A successful use of STTAR in an AR application was to track a dental drill for root canal therapy [11]. The tracked drill pose updated live overlays of cone-beam-CT slices on the dentist's HMD. In phantom trials with six novices, the HMD-only method achieved a mean positional error of 1.3 mm on the crown plane and an angular deviation of 1.8°.

As another example, Gadwe et al. [12] introduced a marker-based method that tracks a cylindrical instrument's 6-DoF pose using a fiducial pattern of squares and circles wrapped around the instrument shaft, achieving mean translation and rotation errors of 1.3 mm and 1.5°.

While fiducial markers can lead to effective AR guidance, the challenge of introducing markers into the sterile field has led to research on markerless pose estimation. Doughty et al. proposed HMD-EgoPose [13], a single-shot CNN for markerless 6-DoF pose estimation of a surgical drill and the user's hand from monocular RGB video. On a real benchmark dataset, the network achieved a drill tip error of 28.1 mm and a direction error of 3.9°. A separate feasibility experiment demonstrated sending video from a HoloLens 2 to a remote workstation for inference, then returning the estimated pose for a total round-trip latency of approximately 199 ms.

Dehghani et al. [14] developed Colibri5, which tracks a trocar's 5-DoF pose monocularly in real time for robot-assisted vitreoretinal surgery. On a phantom eye dataset, Colibri5 achieved average orientation errors of 3° at 15 fps, with translation errors of 4.6 mm, 3.2 mm, and 1.5 mm along X, Y, and Z. While not HMD-based, this demonstrated the feasibility of markerless pose tracking of a trocar.

FoundationPose [15] is a transformer-based foundation model trained on large-scale (∼1 million) synthetic images for unified 6D pose estimation and tracking. It can generalize to novel objects without per-object training (given a CAD model or a few reference images) and significantly outperformed specialized methods on standard benchmarks. This underscores the power of massive synthetic data and attention-based architectures, which we adopted in our method. However,

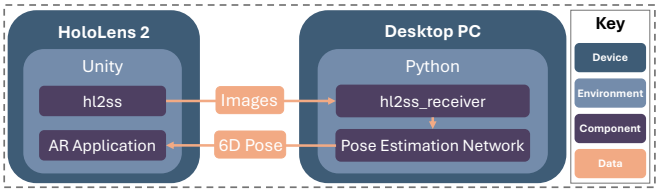

Fig. 2. Data flow for AR-Based trocar tracking: HoloLens 2 streams RGB and stereo grayscale via hl2ss to a workstation, the pose-estimation network predicts 6-DoF trocar pose in real time, and poses are sent back to the HMD for overlay.

FoundationPose requires about 1.3 s for pose estimation before it switches to a 32 Hz tracking mode. If tracking is lost, the full 1.3 s pose-estimation is required for reinitialization, which may be disruptive in time-sensitive applications.

Another notable approach is SC6D [16], a symmetry-agnostic and correspondence-free framework for 6D pose estimation. It predicts an object segmentation mask alongside a visual embedding that is matched against pre-computed SO(3) rotation embeddings to estimate orientation. We similarly employ object mask prediction to aid our network.

While the above works represent significant accomplishments toward tracking for AR, to the best of our knowledge, no prior work has demonstrated real-time monocular markerless tracking of a laparoscopic trocar using an HMD.

III. Method

Our goal is to estimate the pose of the assistant surgeon's trocar relative to the HMD camera in real time. We first describe the devices and physical setup in Section III-A, then our markerless pose-estimation network in Section III-B, training strategy in Section III-C, and shaft detection method in Section III-D.

A. System Overview

As shown in Fig. 2, a Microsoft HoloLens 2 serves as the image source for all experiments as well as the augmented-reality display. RGB frames are streamed at 720×1280 and 30 fps, and the left- and right-front visible-light (grayscale) cameras provide 640×480 at 30 fps via HoloLens 2 Research Mode [17]. Image streams are transmitted using `hl2ss` [18] to an off-device workstation for real-time processing. Factory camera calibrations supplied by the specific HoloLens 2 device are used. Estimated poses are returned to the HoloLens 2 via UDP. The workstation is equipped with an Intel Core i9-10850K (10 cores, 2020) and an NVIDIA RTX 3090 (2020).

B. Markerless Pose Estimation

Our markerless pose estimation method uses a network to predict 2D keypoints of the trocar in the image, which correspond to known 3D keypoints on the trocar model. We then estimate the pose using PnP.

1) Trocar Keypoint Selection: We manually chose 11 keypoints on the trocar. The keypoints were chosen based on the features of the model, and were chosen to be asymmetrical. Not all points are directly visible in every view (e.g., the far-side points might be self-occluded by the trocar depending on the view). However, we chose points that, when projected onto the image plane, will most likely lie within the trocar

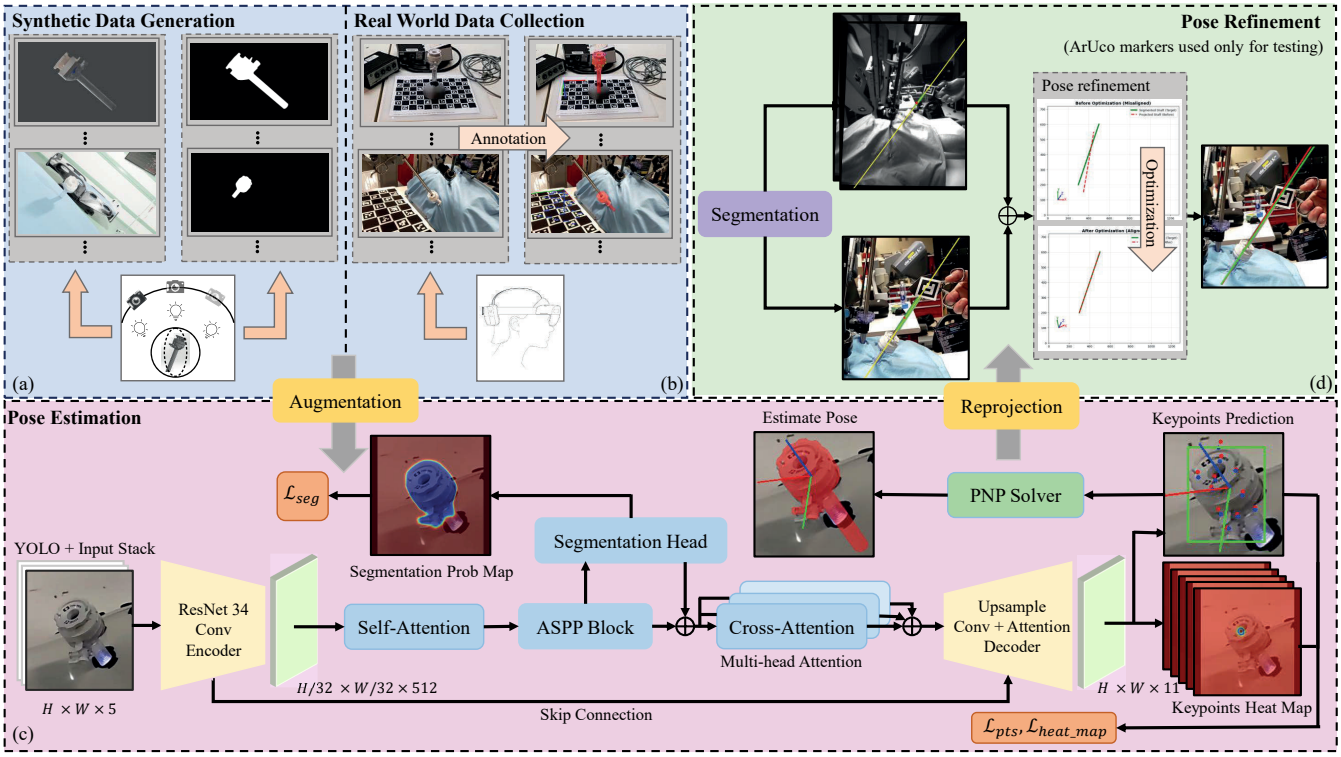

Fig. 3. An overview of our training data, keypoint prediction network, pose estimation via PnP and multi-view refinement pipeline. (a) Synthetic photorealistic data generation of the trocar using randomized pose/lighting via BlenderProc. We generate a pose, an RGB image, and a mask. (b) Real data collection via HL2 videos of the trocar inserted in different phantoms. We get 6-DoF ground truth (and a mask) from a hand-annotated registration to a tracked ChArUco board. (c) U-Net-inspired architecture for keypoint detection using YOLO detection as the input. Our architecture uses a ResNet-34 encoder. The bottleneck fuses self-attention, ASPP, and a segmentation head via cross-attention on resized mask logits. In the up-sample decoder, skip connections from corresponding encoder levels are used for multi-scale fusion, and there is a self-attention layer in the middle of the decoder layers. The output is 11 keypoint logit heatmaps, converted to sub-pixel coordinates via soft-argmax, then fed to RANSAC-PnP for 6-DoF pose. (d) Our Multi-view shaft refinement. We seed SAMURAI with a projected point, fit PCA lines in RGB + stereo, then solve a 4-DoF line fit with keypoint regularization. Red Line: before refinement, Green Line: after refinement. Note: The visible ArUco markers were not used for tracking, only for evaluation.

mask in the image. For example, a point near the tip, when projected onto the image, would likely lie somewhere on the phantom, outside the visible part of the trocar.

We chose 11 keypoints informed by the ablation study in PVNet [19], which compared different numbers of keypoints in a similar scenario. They found best results with 8, though 12 also performed well. Using more than 8 gives room for outlier rejection during the PnP step.

2) Network Architecture: The trocar is a difficult object to track due to partial radial symmetry, textureless surfaces, and specular reflections caused by its glossy finish under overhead surgical lighting.

The input to our network is a cropped image of the trocar provided by YOLOv11 [20], which was fine-tuned on bounding boxes derived from the real dataset (described in Section III-C.2). Bounding boxes were generated by projecting a shortened trocar model (in order to exclude the occluded portion inside the phantom) onto the image plane using the 6-DoF pose annotations.

To handle the challenges of the trocar appearance, our network architecture (Fig. 3) adopts an encoder-decoder paradigm inspired by U-Net [21], augmented with attention mechanisms [22] and multi-scale feature aggregation. The network jointly predicts keypoint heatmaps and a binary segmentation mask of the trocar. Input RGB frames are augmented with two coordinate channels which linearly encode the x and y coordinates in normalized coordinates $[-1, 1]$. This explicit spatial representation aids in localizing keypoints relative to the trocar's geometry and helps to mitigate visual ambiguities. The encoder employs a lightweight ResNet-34 backbone [23], initialized from scratch and inspired by the structure from [24].

At the bottleneck, a self-attention block builds a global summary of the scene and reweights feature channels accordingly. The subsequent ASPP layer [25] applies parallel dilated convolutions at multiple rates to merge local features with broader spatial context without downsampling. Together, these operations promote a globally consistent representation and reduce sensitivity to context-inconsistent distractors such as background shafts or cables.

A dedicated segmentation head branches from the bottleneck features, progressively upsampling through convolutional layers with ReLU activations and batch normalization to output a logit map for binary segmentation, inspired by SC6D [16]. This mask prediction not only regularizes the network by encouraging coherent keypoint placements within the instrument boundary, but also serves as a soft prior for subsequent cross-attention fusion. The predicted mask is resized to match the bottleneck resolution and then mapped to the channel dimension via a 1×1 convolution. The mapped

mask and the RGB bottleneck features are fused via a multi-head cross-attention layer, and the output residuals are added to the bottleneck features. Such integration draws from multi-head cross-attention designs in pose estimation frameworks like [15].

The decoder mirrors the encoder's hierarchy, employing bilinear upsampling followed by convolutions and skip connections from the corresponding encoder levels to recover detail. A single self-attention layer in the mid-decoder further refines representations. The final output layer produces a logit map for each keypoint. These logits represent unnormalized heatmaps, from which sub-pixel keypoint coordinates are extracted via a soft-argmax operation.

3) 6D Pose prediction: Once the 2D keypoints are predicted, we recover the 6-DoF pose by using OpenCV's `solvePnPRansac` with the EPnP method [26]–[28]. This finds the pose that minimizes reprojection error of the known 3D model points, with RANSAC outlier rejection to handle occasional detection failures.

For our handheld instrument shaft tracking, our predicted trocar pose seeds a multi-view shaft-line estimation which is described in Section III-D.

4) Loss Construction: The network outputs keypoint logits and one segmentation logit. Our training data contains three supervision sources: a binary segmentation mask, labeled keypoint coordinates, and keypoint heatmaps which are generated as normalized Gaussians ($\sigma = 5$) centered at the ground-truth keypoints.

We construct our loss as a weighted sum:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{heatmap}} + \lambda_{\text{coord}}\mathcal{L}_{\text{coord}} + \lambda_{\text{seg}}\mathcal{L}_{\text{seg}}, \tag{1}$$

where λ_{coord} and λ_{seg} balance the terms. Heatmap loss $\mathcal{L}_{\text{heatmap}}$ applies KL-divergence between the spatial softmax of the predicted logits and the ground-truth Gaussian heatmaps. Coordinate loss $\mathcal{L}_{\text{coord}}$ uses Huber (smooth L1) on extracted soft-argmax points. The choice of Huber provides robustness to outliers. Segmentation $\mathcal{L}_{\text{seg}}$ applies binary cross-entropy on logits. The composite loss $\mathcal{L}_{\text{total}}$ is optimized via Adam with cosine annealing.

C. Training with Synthetic and Real Data

A significant contribution of our work is an effective strategy for training our network with synthetic data and real data, which overcame frequent tracking failures when the network was trained with only real data. In the following subsections, we describe our synthetic data generation, our method for collecting real data, and our training strategy.

1) Highly Realistic Synthetic Data Generation: Training with photorealistic synthetic data has been shown to consistently improve accuracy on real-image 6D pose benchmarks. In 2020, the Benchmark for 6D Object Pose Estimation (BOP) challenge [29] introduced BlenderProc [30], a tool to procedurally generate ray-traced images using Blender's cycles engine. The top entries leveraged these renders during training [31]. In reported ablations across these challenges, every method that augmented real data with synthetic images outperformed its real-only counterpart [32].

We generated two synthetic datasets using BlenderProc [30] with the same camera intrinsics as our HL2 camera. We took care to use realistic materials (e.g., white glossy plastic or semi-transparent plastic) for our trocar model.

Our first dataset, Syn-Empty (SE), contains the trocar on an empty background with two point light sources. For each image, the location and intensity of the light sources are randomized, and the camera position is uniformly sampled in a spherical shell centered on the trocar. The inner (0.2 m) and outer (1.0 m) radii of the shell are chosen based on how far the assistant surgeon would be from the trocar during a procedure. The camera orientation is directed at the trocar with random orientation offsets applied using Euler angles, $\pm 90°$ for roll and $\pm 45°$ for pitch and yaw. In addition to the RGB image, we also generate a segmentation mask.

Our second dataset, Syn-Hospital (SH), contains renders of the trocar model in a hospital scene. The scene consists of a room with various objects such as a surgery table, a medical robot, a computer monitor, a hanging blood bag, and a hospital bed. Each object has realistic materials for rendering. On the surgery table is a mock "patient" underneath surgical drapes. The trocar orientation and position are randomized such that it may be placed in various poses which intersect the "patient". In this scene, there are three point light sources which are randomized in position and intensity, but they always remain above the surgical table to represent overhead lights. The camera position is uniformly sampled in a hemispherical shell centered on the patient and normal to the surgical table with the same random radius as before. The camera orientation is chosen to point towards the object but such that the camera roll is "up" (i.e., as if someone were wearing an HMD). The camera then has a random rotation applied in the same manner as before. Notably, in this dataset, the trocar is always partially occluded by the "patient" and the generated segmentation mask reflects this.

2) Real Data Collection: Our real data collection required images from the HoloLens 2 camera in realistic conditions, without any visible markers attached to the trocar, and with a fast collection process.

We tracked an OpenCV ChArUco board [26], [33]–[35], which is robust to partial occlusion. In one set of data, we created a small hemispherical phantom (diameter 80 mm) which sits at the center of a standard letter sized (21.6×27.9 cm) ChArUco board. The trocar was inserted into the phantom and we recorded videos. In each frame, we found the ChArUco board pose, and computed the pose of the trocar relative to the camera given a registration. We manually registered the model of the trocar to the tracked ChArUco coordinate frame with an interactive GUI. The GUI shows the projection of the trocar model onto calibrated RGB, left-grayscale, and right-grayscale images from different video frames. The trocar registration is "nudged," using a keyboard, until it looks visually correct in the different images and frames.

In another set, we created a very large ChArUco board (50×90 cm) which was placed alongside a full-sized training phantom. We then inserted the trocar into the phantom. We also inserted an instrument through the trocar in approximately

half of the images. We used the same method to manually register the trocar model to the board. However, we also corrected for error in the ChArUco tracking caused by motion artifacts (such as rolling shutter distortion) by interactively micro-adjusting the trocar to camera pose in each frame using a similar GUI. Usually, we only adjusted the position, as the orientation was more stable. We used the previous video frame's adjustment as the starting point for the next frame to make this process go quickly. Our adjustment speed exceeded 1000 images per hour per person. The two phantoms and ChArUco boards are visible in Fig. 3b.

To improve robustness, we intentionally captured some images where the trocar was partially occluded. We varied the HoloLens 2 camera ISO gain and exposure time across recording sessions to help make our method robust to various lighting conditions. We further applied standard practice data augmentation during training.

3) Three Stage Training Strategy: We generated/annotated 20,000 images for each dataset. All three datasets provide an RGB image with a binary segmentation mask of the trocar, and the 2D projections of the 11 keypoints as training labels. We did not use any pre-trained weights. The first stage of training was with the SE dataset for 400 epochs and with a high learning rate of 0.001. This forced the model to learn the structure of the keypoints. The second stage involved training on a 30% SE and 70% SH mix with a 0.0001 learning rate for 600 epochs. The final stage was 30% SH and 70% real data with a 0.0001 learning rate for 400 epochs. The total training time was approximately 60 hours on the workstation described in Section III-A.

D. Shaft Detection for Pose Refinement

Given the predicted 6-DoF trocar pose, we are able to provide some directional information to the assistant surgeon for guiding the handheld surgical instrument. One caveat, however, is that the inner diameter of the trocar is often larger than the outer diameter of the handheld instrument, leading to significant play between the trocar and the instrument shaft. This can cause a less accurate shaft direction prediction. In our case, we used a 10 mm diameter instrument with a 5-12 mm trocar (the inner diameter was measured at 13.3 mm). The maximum angle error between this instrument and trocar is calculated to be 2.1°. For a 5 mm instrument, the maximum error is calculated as 5.5°. We address this by using the trocar pose as a hint to find the instrument shaft directly in the RGB and grayscale images.

We detect the shaft by employing SAMURAI [36], a real-time segmentation tracker. To initialize tracking, a fixed 3D point on the z-axis (along the shaft) of the trocar's coordinate system is projected to 2D in all images using the estimated trocar pose. The projected point serves as a "point prompt" for the segmentation network. For each image (RGB, left-grayscale, and right-grayscale), SAMURAI provides a segmentation prediction of the handheld instrument's shaft. We clean the predicted binary mask by retaining the largest connected component and pruning the bottom 20% of points along the principal axis. In each image, a line is fitted to the remaining pixels using principal component analysis: the mean $\boldsymbol{\mu} \in \mathbb{R}^2$ yields a point on the line, and the dominant eigenvector $\mathbf{v}_1 \in \mathbb{R}^2$ yields the direction. The resulting 2D lines $\ell_{\text{RGB}}, \ell_{\text{L}}, \ell_{\text{R}}$, lie along the shaft in each respective image.

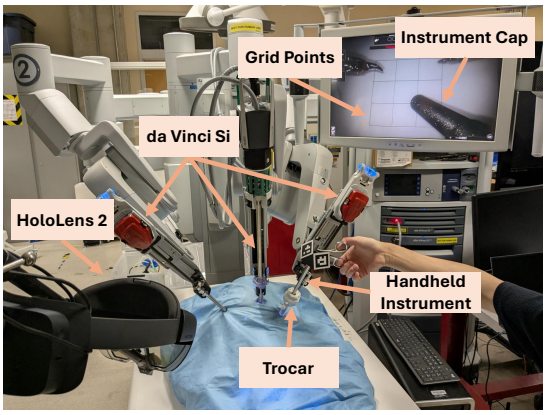

Fig. 4. da Vinci Si docked to a draped phantom with an internal 4×4 target grid (20 mm). A trocar enabled handheld instrument passage; the tip carried a centered 3D-printed cap and ArUco markers. Tip contact was monitored on the endoscope display. A tripod-mounted HoloLens 2 at an assistant-like vantage defined the world frame, independent of SLAM.

We take the initial handheld instrument pose $\mathbf{T}_0 \in SE(3)$ to be equal to the previously predicted trocar pose. We then refine this so that its axis better matches the true instrument axis. We project two canonical 3D shaft endpoints $\mathbf{q}_1$ and $\mathbf{q}_2$, on the z-axis of $\mathbf{T}_0$, onto each image plane. Using an iterative method, we find the pose, formulated as $\mathbf{T}_1 = [\mathbf{r}, \mathbf{t}] \in \mathbb{R}^6$ ($\mathbf{r}$ is the axis-angle representation of the rotation, and $\mathbf{t}$ the translation) to minimize the sum of squared residuals:

$$\min_{\mathbf{T}_1} \sum_i w_i \|\mathbf{e}_i(\mathbf{T}_1)\|^2 \tag{2}$$

where w_i are weights, and $\mathbf{e}_i$ are residuals which encode multi-view constraints.

For each image (RGB, L, R), there are two residuals that are the signed distances between each endpoint of the corresponding detected 2D line (ℓ_{RGB}, ℓ_{L}, or ℓ_{R}), and the projected shaft line (i.e., the 2D line through the projections of $\mathbf{q}_1$ and $\mathbf{q}_2$), yielding 6 residuals.

In the RGB image we obtain an additional 11 residuals by projecting the 3D trocar keypoints onto the RGB image from the starting pose $\mathbf{T}_0$ and finding the 2D distances to the new projections in the final pose $\mathbf{T}_1$. This has a regularizing effect to prevent the pose from shifting too much. We solve for $\mathbf{T}_1$ with an iterative least-squares solver.

In reality, we are only able to recover 4-DoF of information (a 3D infinite line) by looking at the shaft line in the images. We cannot recover the handheld instrument's insertion or rotation about the shaft. Despite this, it is convenient to solve for a 6-DoF pose even though there are two ambiguous degrees of freedom.

IV. Experiment

We performed an experiment to evaluate the accuracy of our method. Since the goal of the trocar tracking is to provide a direction to the assistant surgeon, we focus on the orientation accuracy of our markerless tracking method.

Fig. 5. RGB images from trial 1. The images show one frame of the handheld instrument while it is touching the four outermost corners of the grid. The predicted trocar pose is shown in each image as a red overlay and a coordinate frame.

A. Experimental Setup

Our experimental setup, as shown in Fig. 4, consists of a visually realistic robot-assisted surgical scene with a training phantom and a da Vinci Si. The da Vinci is docked to the phantom. A 4×4 paper grid with 20 mm spacing, placed inside the phantom, serves as a ground truth. The phantom is covered with a blue surgical drape to better mimic a clinical scenario. We simulate the role of the assistant surgeon by touching the grid points with a handheld instrument that is inserted into the phantom through the trocar. We use the robotic endoscope video monitor to view the grid and instrument tip when touching the points.

The instrument does not have a well-defined tip, so we attached a 3D-printed cap with a sharp tip centered on the instrument axis. We also attached a pair of ArUco markers to the end of the instrument shaft, which avoids interference with our markerless tracking method.

During the experiment, the HoloLens 2 is placed on a tripod at a realistic distance and orientation from the phantom. We keep the HoloLens fixed for each trial of our experiment to avoid relying on the device's SLAM to relate different camera poses throughout a trial.

B. Experimental Procedure

Before our experiment, we performed a pivot calibration to find the instrument tip with respect to each ArUco marker using the RGB images. We also calibrated the shaft direction using a jig which constrained the instrument to only rotate about the shaft. We placed the trocar at four distinct locations within the phantom for diversity. Two trials were recorded per location (8 trials total).

For each trial, we touched the instrument tip to each of the 16 grid points on the planar target (20 mm spacing). An example of the collected data can be seen in Fig. 5. We recorded 30 video frames (1 second) while maintaining contact at each point, yielding a total of 480 frames captured with the HoloLens 2 for offline analysis. Each frame contains the RGB, left grayscale, and right grayscale images. The RGB images were processed using our markerless trocar pose estimation method to compute the 6-DoF trocar pose. The instrument shaft is assumed to be coaxial with the trocar centerline. For each prediction, we subsequently used our multi-view shaft refinement with all three images to get a refined shaft direction. Additionally, we recorded the instrument tip position and shaft directions given by the ArUco markers.

For each trial, we fitted a plane in HL2 coordinates using the measured tip points from the ArUco markers. Next, we computed errors for the shaft direction predictions for three cases: our tracking method without refinement, our method with multi-view refinement, and the ArUco marker (shaft directions obtained from the jig calibration). For each trial and method, we registered the grid points to the measured shaft lines by finding a grid-pose (3 DoF) on the trial's fitted plane which minimizes the distance of each grid point to its corresponding shaft line. We also found the remote center of motion (RCM) point as the point which minimizes the distance to each recorded shaft line.

Finally, for each predicted shaft direction we computed the in-plane distance error in mm, and the orientation error as the angle between the recorded shaft direction and the direction from the RCM point to the corresponding grid-point

C. Results

Table I reports in-plane position error (mm) and angular error (deg) over eight trials (3,840 frames). We evaluated the estimated instrument shaft line in two cases: (i) the direct markerless trocar pose prediction ("before refinement") and (ii) the multi-view shaft refinement result using case (i) as the initialization ("after refinement"). Before refinement, the mean planar position error was 7.10 mm (median 6.70 mm, std. 3.60 mm), with a maximum of 19.99 mm and 22 outliers. The mean angular error was 2.40° (median 2.14°, std. 1.39°), with a maximum of 7.82°. After refinement, errors improved substantially: mean planar position error dropped to 5.50 mm (median 4.60 mm, std. 3.50 mm), with no outliers and a maximum of 19.20 mm. The mean angular error reduced to 1.87° (median 1.51°, std. 1.32°), with a maximum of 6.74° and zero outliers.

Multi-view refinement improved results in every trial, with the largest relative improvements in trials with higher pre-refinement error (e.g., Trial 7: mean planar error from 9.00 mm to 6.70 mm; angular from 2.77° to 2.13°). Across all trials, refinement eliminated all outliers. We set the outlier threshold to 20 mm, which is the size of the grid.

Our full pipeline (YOLO + pose prediction network + PnP) runs at 66.38 Hz before refinement (tracking is limited to 30 Hz by the HoloLens 2 camera capture rate), enabling real-time AR overlays. We measured latency by filming videos with a 30 Hz camera through the HoloLens display and counting the frames between the start of a sudden motion of the real trocar and the start of motion of the AR overlay. In all 10 video clips considered, the latency was exactly 6 frames, which is nominally 200 ms. The multi-view refinement reduced error, but the downside is that the inference time for three SAMURAI segmentation instances is too slow for real-time use.

TABLE I. Position (mm) and angular (deg) errors from our experiment. The "Network Prediction" block corresponds to the raw trocar pose predictions from our markerless method (Section III-B). The "Multi-View Refinement" block shows the results after refining the instrument shaft line (Section III-D).

Trial	Error Type	Network Prediction						Multi-View Refinement					
		Mean	Median	Std	Min	Max	Outliers	Mean	Median	Std	Min	Max	Outliers
Trial 1	Planar (mm)	6.20	5.60	3.30	0.70	15.70	0	4.70	3.70	3.30	0.20	15.10	0
	Angle (deg)	2.26	1.88	1.41	0.25	6.47		1.78	1.40	1.34	0.05	6.17	
Trial 2	Planar (mm)	6.70	6.30	3.10	0.30	16.50	0	4.00	3.30	2.40	0.90	11.90	0
	Angle (deg)	2.38	2.15	1.27	0.12	6.36		1.40	1.09	0.87	0.24	4.00	
Trial 3	Planar (mm)	8.00	8.00	3.33	0.80	16.80	0	7.20	6.60	3.30	0.00	15.20	0
	Angle (deg)	3.14	2.90	1.51	0.30	7.82		2.74	2.51	1.42	0.02	6.51	
Trial 4	Planar (mm)	8.10	8.20	3.70	0.40	16.90	0	7.70	7.90	4.00	0.02	16.60	0
	Angle (deg)	3.29	3.30	1.58	0.15	7.13		2.92	2.88	1.55	0.08	6.42	
Trial 5	Planar (mm)	5.80	5.70	2.90	0.60	18.9	11	4.00	3.60	1.90	0.20	9.80	0
	Angle (deg)	1.58	1.57	0.72	0.15	3.72		1.15	1.03	0.60	0.08	3.36	
Trial 6	Planar (mm)	5.40	5.30	2.70	0.40	18.80	1	3.70	3.33	2.00	0.50	11.50	0
	Angle (deg)	1.52	1.40	0.74	0.09	3.68		1.12	0.97	0.66	0.11	3.50	
Trial 7	Planar (mm)	9.00	9.00	4.30	0.30	19.90	10	6.70	5.70	4.00	0.06	19.20	0
	Angle (deg)	2.77	2.92	1.32	0.10	7.23		2.13	1.76	1.25	0.15	6.74	
Trial 8	Planar (mm)	7.60	7.30	3.90	0.20	19.70	3	5.70	5.40	3.60	0.30	16.20	0
	Angle (deg)	2.25	2.27	1.11	0.07	6.00		1.71	1.61	1.11	0.09	5.78	
All Trials	**Planar (mm)**	**7.10**	**6.70**	**3.60**	**0.20**	**19.99**	**22**	**5.50**	**4.60**	**3.50**	**0.00**	**19.20**	**0**
	Angle (deg)	**2.40**	**2.14**	**1.39**	**0.07**	**7.82**		**1.87**	**1.51**	**1.32**	**0.02**	**6.74**	

D. Analysis and Discussion

We quantified pose accuracy for our trocar tracking method both before and after multi-view shaft refinement against the ArUco marker as a baseline. We additionally compared our results to other relevant works.

To establish the baseline, we followed the identical procedure for computing position and angular errors, but used the shaft lines recorded by the ArUco markers attached to the handheld instrument shaft. Of the 3,840 frames, we removed 105 outliers, which yielded a mean planar position error of 5.3 ± 3.7 mm and a median of 4.4 mm, where $\pm$ denotes standard deviation. The mean angular error was $1.71 \pm 1.32°$ with $1.38°$ as the median. From both positional and angular perspectives, our pose-refinement results are comparable to this baseline, with an overall mean position error of 5.5 ± 3.5 mm and angular error of $1.87 \pm 1.32°$ (no outliers removed). This indicates that the refined method achieves similar accuracy to marker-based tracking while offering greater stability (i.e., zero outliers). Pre-refinement performance is slightly inferior to the baseline, with a position error of 7.1 ± 3.6 mm and an angular error of $2.4 \pm 1.4°$ (22 outliers), though the standard deviations are nearly identical. Nonetheless, even without refinement, the errors remain within a viable range relative to the baseline—just 2 mm and $0.7°$ worse on average—while offering real-time tracking and greater stability with fewer outliers. Our accuracy is within the approximate 10 mm tolerance that a surgeon subjectively estimated for instrument insertion when aiming at a safe zone in the endoscope frustum [9].

To contextualize our results within the broader landscape of trocar and instrument pose estimation, we reference two related works cited earlier: a marker-based method for shaft detection [12] and a markerless neural network-based approach (Colibri5) [14] for trocar tracking. We note that direct comparison is not possible as each method was evaluated on different objects, clinical contexts, and test sets. Since our application focuses on the insertion direction, we focus on the rotation error reported in these works. Both Colibri5 and our method are markerless, relying on deep learning for real-time inference from monocular RGB images, but each work targets a distinct clinical context: Colibri5 focuses on smaller trocars in vitreoretinal procedures, where sub-millimeter precision is paramount due to the eye's confined workspace. Their system achieves an average orientation error of $3.06 \pm 2.36°$ at 15 fps. With larger laparoscopic trocars, we obtain a post-refinement angular error of $1.87 \pm 1.32°$. Our direct network prediction (no refinement) still achieves reasonable error at $2.4 \pm 1.4°$ and is more than four times faster than Colibri5 at 66 fps. Our result is also comparable to the marker-based approach in [12], which achieved an angular error of $1.50 \pm 0.87°$ with a 6.7 mm diameter cylindrical tool.

E. Limitations

Our method estimates the insertion trajectory (orientation) but does not provide depth along the shaft, so the instrument tip position is not visualized. However, tip localization is not the intended use case: our system guides the instrument *toward* the endoscope field of view, at which point the surgeon switches to direct visual feedback from the endoscope video. Additionally, our evaluation was conducted with the HoloLens 2 fixed on a tripod to isolate pose-estimation accuracy from head motion effects; the ArUco marker baseline was evaluated under identical conditions. In clinical use, a head-worn HMD would introduce additional challenges such as motion blur and temporary trocar occlusion. Qualitatively, our method works under these conditions, but future work would include measuring tracking accuracy.

V. Conclusion

In this work, we present a markerless tracking method for the assistant-port trocar in robot-assisted minimally invasive surgery. Our method enables AR overlays for head-mounted displays without disrupting sterile workflows. Our deep learning-based approach leverages a customized U-Net architecture to predict 11 asymmetrical 2D keypoints followed by RANSAC-PnP for robust 6-DoF pose recovery. A novel three-stage training paradigm achieves high generalization despite labeling imperfections, yielding real-time performance at 66 Hz on a consumer-grade GPU. Experimental validation

in a realistic da Vinci-docked phantom setup demonstrates in-plane tip errors ($\sim 5.5\,$mm) and angular errors ($\sim 1.9°$) for trocar tracking. Accuracy can be improved with multi-view shaft-line refinement that accounts for trocar-instrument diameter mismatches. These metrics are well within the 10 mm approximate clinical threshold for AR-guided instrument insertion estimated by a surgeon [9]. This trocar tracking method is a foundation for comprehensive AR assistance, enhancing the bedside surgeon's spatial awareness and reducing cognitive load. Future work will integrate a faster segmentation pipeline to enable real-time pose-refinement. Ultimately, our method advances toward clinically viable, marker-free AR systems that provide immersive surgical guidance.

ACKNOWLEDGMENT

This study was supported in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government.

REFERENCES

[1] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Ann. Surg.*, vol. 239, no. 1, p. 14, 2004.

[2] O. Sgarbura and C. Vasilescu, "The decisive role of the patient-side surgeon in robotic surgery," *Surg. Endosc.*, vol. 24, no. 12, pp. 3149–3155, 2010.

[3] A. M. Potretzke, B. A. Knight, J. A. Brockman, J. Vetter, R. S. Figenshau, S. B. Bhayani, and B. M. Benway, "The role of the assistant during robot-assisted partial nephrectomy: does experience matter?" *J. Robot. Surg.*, vol. 10, no. 2, pp. 129–134, 2016.

[4] R. Nayyar, S. Yadav, P. Singh, and P. N. Dogra, "Impact of assistant surgeon on outcomes in robotic surgery," *Indian J. Urol.*, vol. 32, no. 3, p. 204, 2016.

[5] L. Qian, A. Barthel, A. Johnson, G. Osgood, P. Kazanzides, N. Navab, and B. Fuerst, "Comparison of optical see-through head-mounted displays for surgical interventions with object-anchored 2D-display," in *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 6, 2017, pp. 901–910.

[6] K. Makiyama, K. Osaka, A. Araki, S. Ohtake, T. Tatenuma, M. Nagasaka, T. Yamada, and M. Yao, "How to reduce the risk of organ injuries during surgical instrument insertion in laparoscopic surgery: Pushing/pressing force analysis using forceps with sensors," *Asian J. Endosc. Surg.*, vol. 14, no. 3, pp. 504–510, 2021.

[7] L. Qian, A. Deguet, and P. Kazanzides, "ARssist: augmented reality on a head-mounted display for the first assistant in robotic surgery," *Healthcare Technol. Lett.*, vol. 5, no. 5, pp. 194–200, 2018.

[8] L. Qian, A. Deguet, Z. Wang, Y.-H. Liu, and P. Kazanzides, "Augmented reality assisted instrument insertion and tool manipulation for the first assistant in robotic surgery," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 5173–5179.

[9] N. Greene, A. Long, Y. Long, Z. Han, Q. Dou, and P. Kazanzides, "Markerless tracking of robotic surgical instruments with head mounted display for augmented reality applications," *Healthcare Technol. Lett.*, vol. 12, no. 1, p. e70044, 2025.

[10] A. Martin-Gomez, H. Li, T. Song, S. Yang, G. Wang, H. Ding, N. Navab, Z. Zhao, and M. Armand, "STTAR: surgical tool tracking using off-the-shelf augmented reality head-mounted displays," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 7, pp. 3578–3593, 2023.

[11] F. Li, Q. Gao, N. Wang, N. Greene, T. Song, O. Dianat, and E. Azimi, "Mixed reality guided root canal therapy," *Healthcare Technol. Lett.*, vol. 11, no. 2-3, pp. 167–178, 2024.

[12] A. Gadwe and H. Ren, "Real-time 6dof pose estimation of endoscopic instruments using printable markers," *IEEE Sensors J.*, vol. 19, no. 6, pp. 2338–2346, 2018.

[13] M. Doughty and N. R. Ghugre, "HMD-EgoPose: Head-mounted display-based egocentric marker-less tool and hand pose estimation for augmented surgical guidance," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 17, no. 12, pp. 2253–2262, 2022.

[14] S. Dehghani, M. Sommersperger, M. Saleh, A. Alikhani, B. Busam, P. Gehlbach, I. Iordachita, N. Navab, and M. A. Nasseri, "Colibri5: Real-time monocular 5-dof trocar pose tracking for robot-assisted vitreoretinal surgery," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024, pp. 4547–4554.

[15] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "FoundationPose: Unified 6d pose estimation and tracking of novel objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 17 868–17 879.

[16] D. Cai, J. Heikkilä, and E. Rahtu, "SC6D: Symmetry-agnostic and correspondence-free 6D object pose estimation," in *Int. Conf. 3D Vis. (3DV)*, 2022, pp. 536–546.

[17] D. Ungureanu, F. Bogo, S. Galliani, P. Sama, X. Duan, C. Meekhof, J. Stühmer, T. J. Cashman *et al.*, "Hololens 2 research mode as a tool for computer vision research," *arXiv preprint arXiv:2008.11239*, 2020.

[18] J. C. Dibene and E. Dunn, "Hololens 2 sensor streaming," *arXiv preprint arXiv:2211.02648*, 2022.

[19] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 4561–4570.

[20] G. Jocher and J. Qiu, "Ultralytics yolo11," https://github.com/ultralytics/ultralytics, 2024.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, 2015, pp. 234–241.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2961–2969.

[25] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[26] G. Bradski, A. Kaehler *et al.*, "OpenCV," *Dr. Dobb's J. Softw. Tools*, vol. 3, no. 2, 2000.

[27] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal *et al.*, "BOP: Benchmark for 6D object pose estimation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–34.

[30] M. Denninger, D. Winkelbauer, M. Sundermeyer, W. Boerdijk, M. Knauer, K. H. Strobl, M. Humt, and R. Triebel, "BlenderProc2: A procedural pipeline for photorealistic rendering," *J. Open Source Softw.*, vol. 8, no. 82, p. 4901, 2023.

[31] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP challenge 2020 on 6D object localization," in *Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 577–594.

[32] M. Sundermeyer, T. Hodaň, Y. Labbe, G. Wang, E. Brachmann, B. Drost, C. Rother, and J. Matas, "BOP challenge 2022 on detection, segmentation and pose estimation of specific rigid objects," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 2785–2794.

[33] F. J. Romero-Ramírez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, 2018.

[34] OpenCV Contributors, "Calibration with ArUco and ChArUco; detection of ChArUco boards; ArUco FAQ," https://docs.opencv.org/, accessed Sep 2025.

[35] 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 Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[36] C.-Y. Yang, H.-W. Huang, W. Chai, Z. Jiang, and J.-N. Hwang, "SAMURAI: Adapting segment anything model for zero-shot visual tracking with motion-aware memory," *arXiv:2411.11922*, 2024.