# Robotic Ultrasound Needle Placement and Tracking: Robot-to-Robot Calibration

Christopher Hunt  Matthew Walmer
Mentors:
Bernhard Fuerst  Risto Kojcev  Javad Fotouhi  Nassir Navab

**Technical Summary**

*Introduction and Background*

In medicine and industry[1], there is a need for flexible multi-robot platforms that allow for independent placement of multiple robotic manipulators. The Computer Aided Medical Procedures research group at Johns Hopkins University has developed a multi-robot surgical platform with two KUKA iiwa robotic manipulators. Each KUKA system is built into a mobile workstation, which makes them easy to transport and less obtrusive in an operating room environment. Instead of an externally mounted tracking camera, each KUKA is mounted with an Intel RealSense RGB-D camera.

One surgical application which has been explored with this platform is ultrasound guided dual-robotic needle placement[2]. In this surgery, one robotic manipulator controls an ultrasound probe to capture patient anatomy, while the other arm precisely inserts a needle. In such a procedure, precise robot to robot coordination is key. For platforms with rigidly connected robotic arms, the transformation between the arms' reference frames can be assumed constant, and need only be calibrated once. For this flexible platform, because each robot base can be positioned independently, base to base calibration must be performed frequently. Thus, there is a need for an efficient method to precisely calibrate multiple robots.

The objective of this research is to explore a variety of robot-to-robot calibration methods and validate their efficacy for use in dual-robotic surgeries and experiments.


*Figure 1:* CAMP lab's dual robotic platform with two KUKA iiwa robotic manipulators.

*Hardware*

CAMP lab's dual-robotic platform has two KUKA iiwa manipulators. Each robot has an Intel F200 RealSense RGB-D camera, capable of capturing depth information through infrared structured light as well as high definition video. For these experiments, the two KUKA systems are positioned facing each other approximately 1 meter apart. In this configuration, the workspace is the half meter platform between the two KUKAs, which is meant to simulate the operating room table.

*Setup*

There are two calibration steps which must be performed before the robot-to-robot calibration. First, a monocular camera calibration must be done to obtain intrinsic parameters and to undistort images. Second, it is necessary to determine the hand-eye transformation that relates a robot's camera space and the robot end-effector frame of reference. In figure 2 below, this is represented by ${}^{G}H_{C}$.

In the envisioned workflow for this dual-robotic system, it is assumed that the cameras and camera mounts remain constant, such that repositioning the robot bases does not change any of these parameters. Thus, camera calibration and hand-eye calibration can be performed in advance and do not disrupt the operation's workflow. For each of the robot-to-robot calibration methods, it is assumed that the camera parameters and hand-eye transformations are already known.

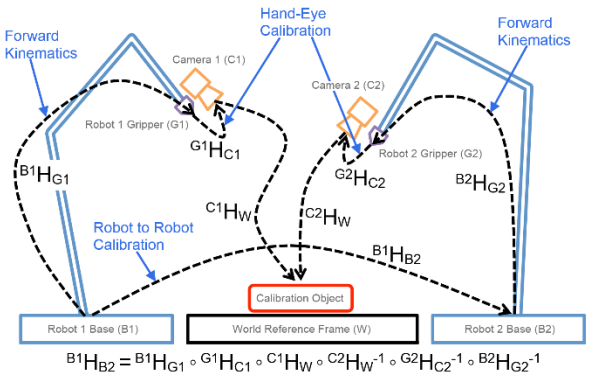


*Figure 2:* Schematic overview of the dual robotic platform and the calibrations used to determine each transformation.

*Monocular Camera Calibration*

This procedure uses a regular pattern of known dimensions to determine the intrinsic parameters of the camera[5]. For these experiments, a checkerboard is used as the calibration object. Multiple images of the calibration object are taken from different viewpoints, and a maximum-likelihood estimator is applied to solve for the camera intrinsics, extrinsics and distortion coefficients. Intrinsic parameters are defined as the principal point offset, skew, and focal length. Extrinsic parameters are the three-dimensional poses of the camera with respect to the calibration object reference frame for each image. These are later utilized in the hand-eye calibration. This project uses the camera calibration implementation found in MATLAB's Computer Vision Toolbox[6].

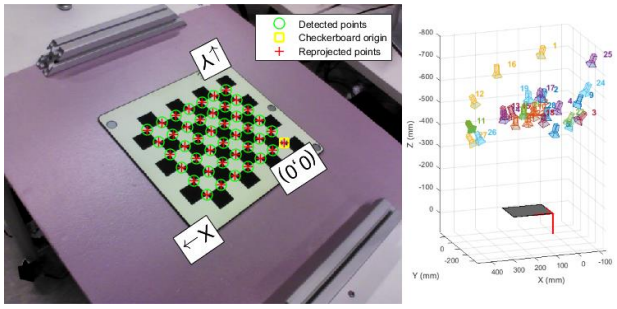


*Figure 3:* Matlab camera calibration application detecting corners (left) and extrinsic camera data, i.e. camera poses with respect to the checkerboard based frame (right).

*Hand-Eye Calibration*

The hand-eye calibration finds the transformation between the robot end-effector and the camera frame. For each checkerboard image, the board pose in the camera frame (${}^{c}H_{cb,i}$) is matched with the corresponding end-effector pose in the robot's base frame (${}^{b}H_{g,i}$). Using these transformations, an AX=XB problem can be set up to solve for the constant transformation between the camera space and the end-effector space (${}^{g}H_{c}$)[8]. The method used in this project to solve this set of equations is a nonlinear least-squares optimization[9]. The implementation of this algorithm is available open-source from the California Institute of Technology's Computer Vision Toolbox[7].

*Robot-Robot Calibration*

Three separate calibration algorithms were developed and explored throughout the course of this research: checkerboard, ARToolKit markers, and RGB-D image feature and depth information.

*A. Checkerboard*

In *Monocular Camera Calibration*, it is outlined how to compute the pose of a checkerboard with respect to a camera (the extrinsic parameters). Using this, along with the hand-eye transformation and the forward kinematics of the robot, the transformation from the robot base frame to the checkerboard frame can be computed. Once this is done for both robots, the base to base transformation can be computed through the checkerboard frame. Multiple pictures of the checkerboard are taken from a wide range of angles to improve the accuracy of the calibration.

*Figure 4:* A selection of checkerboard calibration images.

*B. ARToolKit Markers*

ARToolKit is a computer vision framework which provides a host of visually distinctive markers as well as software that can recognize and determine their 3D pose with respect to the camera. This calibration method determines the robot-to-robot transformation through a commonly observed ARToolKit marker, following a similar procedure to the checkerboard calibration. ARToolKit supports a host of different markers including single markers and multimarkers. A multimarker is an array of single markers arranged in a fixed, known orientation to one another. ARToolKit identifies all visible markers in an image and uses any observed markers to estimate the pose of the multimarker array as a whole. Multimarkers provide more accurate pose estimation because of the increased number of feature points.
Multimarkers need not be planar, and ARToolKit provides cubic templates. A multimarker with orthogonally oriented markers could provide better rotation estimation than a simple planar multimarker or checkerboard. For this robot-to-robot calibration method, three marker modalities were tested: a single marker, a 4x3 planar multimarker, and a non-planar orthogonal multimarker.

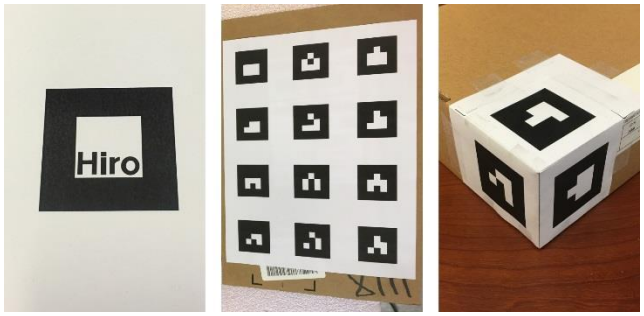


*Figure 5:* Three ARToolKit markers used for calibration: a single "Hiro" marker, a 4x3 multimarker array, and a non-planar multimarker with three orthogonal markers.

ARToolKit applications using multimarkers require a .dat file specifying the orientation of the markers with respect to each other. This is easy to determine for planar multimarkers, but difficult for non-planar ones. For simple augmented reality applications, minor inaccuracies in this .dat file will not matter, but for precision calibrations they certainly will. This makes ARToolKit's foldable paper cubes undesirable, as they will easily warp and deform. A great deal of care was put into the manufacture of this test's orthogonal multimarker so it would be structurally stable and accurate to the .dat file. Furthermore, rather than use a cube, the robots were restricted to viewing the same three markers on one corner. This ensured that any errors from inaccuracy in the .dat file would be uniform for both robots.

*C. RGB-D Image Feature and Depth*

The Intel RGB-D cameras used in this system provide traditional RGB images along with depth information using structured light deformation. This last calibration method explores the efficacy of using this feature of the cameras to drive the calibration. To do this, interest points are first determined in the 2D image space using the SURF algorithm[10]. This provides a set of discriminatory features in the calibration scene. In this implementation, these interest points can be extrapolated into a 3D point cloud through inverse ray-casting using the corresponding depth image value to recover the scale.

Furthermore, because of the high discriminatory power of SURF features, it is possible to track an interest point through different frames of data. This makes it possible to build a point-to-point correspondence from different vantage points. Using a Procrustes method, it is possible to determine ${}^{c1}H_{c2}$, the transformation between camera frames for each image pair. Then, by using the corresponding robot poses, it is possible to determine the overall robot-to-robot transform by closing the transformation loop. This is done for each image pair with sufficient feature overlap and, the final base transform is the average of these pairwise transforms.

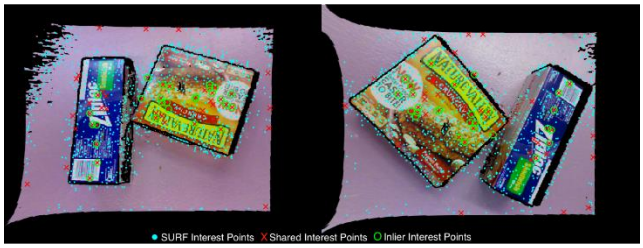


*Figure 6:* Two images of the same calibration scene taken from the two robots. General SURF interest points, interest points shared between the two images, and inliers of those shared interest points are marked as such.

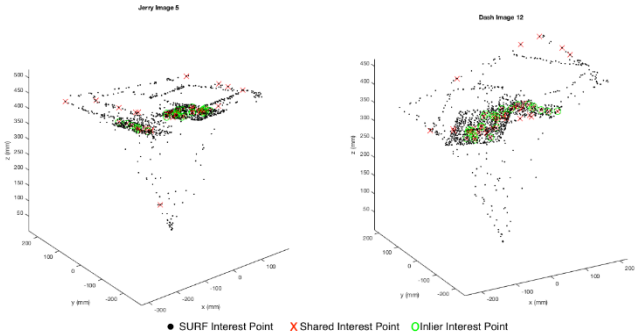


*Figure 7:* The extrapolated 3D interest point clouds of the calibration scene taken from two robot vantage points. General SURF interest points, interest points shared between the two images, and inliers of those shared interest points are marked as such.

## Experiments and Results

*Description of Data Sets*

Checkerboard calibration was tested with two different data sets. Both sets had images taken at 30 positions for each robot. The first set had many images taken at low angles around the board. The second had fewer low angle shots. For both checkerboard calibrations, hand-eye calibration was performed using the same image sets.

For each of the three ARToolKit marker configurations, marker and robot pose data was recorded at 10 different poses for each robot. In the case of multimarker tests, data was only captured at positions where all markers in the multimarker were successfully detected. In the case of the non-planar multimarker, this restricted the range of robot positions where data could be recorded. For the single marker and 4x3 multimarker array, data was captured at positions distributed all around the marker just as was done for the checkerboard. Checkerboard dataset B was used to generate the hand-eye transformation used in these trials. However, additional hand-eye transformations were generated using the pose data from the ARToolKit data sets themselves. The results of these calibrations with both hand-eye methods are listed below.

Two boxes with colorful writing and designs were used to give distinct features for the RGB-D Depth and Feature calibration. For each robot, 16 RGB and depth images were taken at different angles towards the scene. These images were captured with the Intel RealSense camera's color and depth overlay mode, which merges RGB and depth information. For this reason, a separate set of checkerboard images had to be captured to compute a hand-eye transformation.

*Establishing a Ground Truth Transformation*

To assess the accuracy of each of these methods, a ground-truth robot-to-robot transformation was needed. This was determined using the following procedure, which is also described by Gan Yahui and Dai Xianzhong[1]. First, each

KUKA robotic manipulator was fitted with a rigid pointer. The position of the pointer tip was determined using KUKA's built in four point pivot calibration program. First, one robot was moved to an arbitrary point in the work space. Then the second robot was manually maneuvered so that both pointer tips were exactly touching. The three dimensional position of both pointer tips in each robot's reference frames was recorded. This was repeated many times at positions evenly distributed around the workspace. The pointer tip positions create two point clouds, one in each robot's base frame, with known point to point correspondence. The ground truth robot-to-robot transformation was then determined through a Procrustes alignment of the two clouds.

For each of the calibration methods above, the computed robot-to-robot transformation was compared to this ground truth transformation for both linear and angular error. Additionally, for each calibration transformation, the set of robot 2 pointer tip points were projected into the robot 1 base frame using said transformation. For a perfect calibration, these points should exactly coincide with the pointer tip positions recorded with robot 1. The mean linear area for the projected ground truth points was recorded as another metric for calibration accuracy.

*Analysis of Results*

The most accurate calibration was achieved using checkerboard dataset A. It is believed that the wide range of angles captured in the image set allowed for a more precise determination of the board's position with respect to each robot. Checkerboard dataset B had fewer low angle shots, and had greater error for the linear component of the transformation. However, the error for sampled points in the workspace was not significantly worse. While low angle shots may help determine the board pose, they do have some risks. The high projective distortion may make it difficult for computer vision applications to detect the board. For checkerboard dataset A, two images were rejected because the board was not detected. So a middle ground of camera angles is preferable.

In general, the mean linear error for points sampled in the workspace was typically much less than the transformation linear error. This is reasonable, as the transformations are based on observations of calibration object points in the workspace. Some component of the angular error in each transformation compensates for the linear error (or vice versa) so that the error is lower for points in the workspace.

The single ARToolKit marker gave a less accurate calibration than the checkerboard. This should not be surprising as the checkerboard pose is estimated using 42 feature points, each inner corner, while ARToolkit estimates the marker pose using only 4 points, the corners of the marker. However, the 4x3 multimarker's pose is estimated using 12 markers and thus 48 feature points. The 4x3 multimarker did give one of the best calibrations, but it was not as accurate as the checkerboard. Note that the checkerboard data sets used 30 positions, while the ARToolKit calibrations took only 10. A 30 position calibration with the 4x3 multimarker could be comparable to the checkerboard calibration.

For the orthogonal multimarker, the range of robot positions which could observe all three markers was severely restricted, and pose data could not be captured all around the marker. This resulted in a significant linear transformation error. However, as expected, the rotational error was lowest for the orthogonal marker. This demonstrates the importance of capturing data from poses all around the calibration object.

Interestingly, for both ARToolKit multimarkers, there was a significant improvement in transformation linear error when using a hand-eye transformation generated from the same dataset. This effect is particularly pronounced for the orthogonal multimarker, which had a threefold drop in linear error. This may simply be an experimental artifact, or it could be that the least squares estimator of the hand-eye calibration helps to compensate for error in the ARToolKit marker poses.

| Calibration Method [#images/bot] | Hand-eye Calibration [#images/bot] | Sampled Point Mean Linear Error (mm) | Transformation Linear Error (mm) | Transformation Angular Error (deg) |
|---|---|---|---|---|
| Checkerboard A [30] | Checkerboard A [30] | 2.7285 | 2.5994 | 2.5065 |
| Checkerboard B [30] | Checkerboard B [30] | 2.9312 | 5.3538 | 2.3211 |
| ATK single marker [10] | Checkerboard B [30] | 3.8115 | 6.2353 | 2.3956 |
| ATK single marker [10] | ATK single marker [10] | 4.9717 | 8.9781 | 1.8524 |
| ATK 3x4 multi [10] | Checkerboard B [30] | 4.4614 | 7.7350 | 2.5708 |
| ATK 3x4 multi [10] | ATK 3x4 multi [10] | 3.6976 | 3.5423 | 2.7076 |
| ATK non-planar [10] | Checkerboard B [30] | 8.1168 | 24.9807 | 1.7201 |
| ATK non-planar [10] | ATK non-planar [10] | 6.0413 | 8.0032 | 2.0756 |
| RGB-D Features [16] | Checkerboard C [40] | 34.7979 | 51.7487 | 8.4658 |

*Figure 8:* Table of calibration experiment results. For each calibration, the hand-eye transformation used to compute final base to base transform is also listed. Sampled point mean linear error is the mean displacement between ground-truth points once projected into the same base frame. Transformation linear error and transformation angular error are determined with respect to the ground truth transformation described in the section above.

Unfortunately, the RGB-D features and depth calibration method is not nearly accurate enough for medical purposes. With a mean linear error of 34.8 mm between corresponding points, the RGB-D features and depth calibration is by far the least accurate of the methodologies tested. This was surprising as the calibration contains multiple layers of outlier rejection, imposing thresholds for image feature overlap, feature inlier percentage, and sum of square distances in the camera to camera transformation. The most significant contributor to this point-to-point error is the linear transformation error, 51.7 mm, most of which is generated in the y-direction. In the calibration set tested, the y-translational difference between the ground truth transformation and the computed transformation is 53.3 mm. This is in comparison to the relatively low 3.39 mm and 8.61 mm in the x- and z-translational differences. While significant time was spent attempting to discover the source of this unbalanced error, currently, the source remains unknown.

Similarly, the rotational error, 8.5 degrees, is much higher than that of the other algorithms. That being said, this high rotational error may partially explain the higher linear distance errors. Like the checkerboard calibration, there is a noticeable increase in accuracy between two points in the workspace the closer the points are to the center of the calibration scene. The farther out a point is from this center, the more the rotational error affects the linear distance between two points. As such, it is feasible that this calibration can be improved by using multiple different calibration locations around the workspace.

*Conclusions*

In this paper, the efficacy of three distinct calibration methodologies has been explored. For medical procedures that require sub-millimeter accuracy, the checkerboard calibration has proven to provide a viable alternative to industry standards. However, this comes with the caveat that you are operating near the center of calibration. Furthermore, if a calibration multimarker is manufactured to have very precise relationships between each single marker in the grouping, the ARToolKit calibration method seems to have promise as a viable calibration alternative as well. Unfortunately, the RGB-D features and depth calibration is far too inaccurate for medical purposes, however, industrial procedures which require accuracy on the order of centimeters may find this method useful.

The calibration methods which best meet the need for quick but precise robot-to-robot calibration are the checkerboard calibration and the ARToolKit multimarker calibration. For both of these methods, it is recommended that 30 poses be recorded for each robot, which could be done automatically before the start of an operation. However, these methods cannot be applied to dynamically update the robot-to-robot transformation during the surgical procedure. To achieve that, external tracking hardware would still be necessary.

**Management Summary**

*Who Did What*

The initial setup calibrations (camera and hand-eye) were covered by both Christopher and Matthew. The checkerboard calibration was also jointly handled. Testing of ARToolKit calibration methods were performed by Matthew, while the RGB-D depth and features calibration was handled by Christopher. Ground truth point collection and error analysis was done by Matthew.

*Accomplished Versus Planned*

We succeeded in developing and testing three different methods of robot-to-robot calibration. Additionally, we created a comprehensive code package to perform and test the various calibration methods. This package is available on the course wiki under the “Other Resources and Project Files” section.

We were able to produce calibrations with accuracy on the order of millimeters. While a greater degree of accuracy is necessary for surgical applications, we are still satisfied with this result given how many opportunities there are for error to enter the system. To compute the base to base transformation, there are six intermediate transformations: two for robot kinematics, two for hand-eye transformations, and two for camera to calibration object transformations.

The accuracy of the RGB-D feature and depth calibration was disappointing, however, it gave good insights into the feasibility of this methodology in medicine and industry. While medical applications require more precision than what was provided, industrial applications may still be feasible. Processing these sparse and correlated point clouds is fast and provide accuracy on the order of centimeters. Therefore, this methodology may be of use in other real-time applications, such as autonomous mobile robot navigation.

We did not have time to develop a fourth planned calibration method, which would use only the depth cloud generated by the Intel RealSense camera. However, another group has done work relevant to this calibration. Project 06: Augmented Reality for Orthopedic and Trauma Surgeries has done some work with aligning dense partially overlapping point clouds using Fast Point Feature Histograms.

Early on, we planned to perform some needle tracking experiments as a demonstration of the dual-robotic platform. However, over time this became a low priority and was ultimately canceled.

*Future Work*

One course for further research would be development of a depth only calibration method using the Intel RealSense cameras. Based on our results with the RGB-D depth and feature calibration, we do not expect that a depth only calibration would produce an accurate transformation, however, with a sophisticated point cloud alignment algorithm and the greater density of the full depth cloud, it

may be possible. Some avenues to consider may be using Fast Point Feature Histograms to thin the depth cloud while maintaining discriminatory integrity or an ICP variant that optimizes a global energy function.

Additionally, the RGB-D features and depth calibration may be further improved by using multiple micro-calibration scenes scattered across the workspace for base-to-base calibration. In theory, this should reduce the rotational error, which may limit its propagation through the linear distance error. That being said, a similar methodology would work for all of the algorithms provided and, as such, the RGB-D features and depth calibration would still be the least accurate option.

It would be good to explore ARToolKit multimarker configurations that combine the strengths of the markers tested. A non-planar multimarker with several markers on each surface could provide an excellent calibration. This could be built as a full cube or just a concave surface. It would be best if these multimarkers were manufactured by a more precise mechanism, to minimize errors in the configuration file which describes the multimarker geometry.

*What We Learned*

We have gained experience working with many calibration methods, including camera, hand-eye, and pivot calibrations. We've also learned many new things about computer vision, like SURF features and marker tracking with ARToolKit. We had to learn new things about projective geometry and camera models to create our calibration programs.

We learned a great deal from the challenges we faced during this project. For real robotic research, one of the most time consuming things is getting all the hardware working. Trouble shooting is inevitable. Our mentors were of immense help when we experienced these technical problems. Furthermore, for calibration tasks, explicitly stating the spatial relationship between reference frames is crucial. Wrong assumptions on frame orientations propagate to the final transformation, and it is difficult to track down where these errors began.

**References**


1. Y. Gan, and Xong Dai. "Base Frame Calibration for Coordinated Industrial Robots." Robotics and Autonomous Systems 59.7-8 (2011): 563-70. Web.

2. R. Kojcev, B. Fuerst, O. Zettinig, J.Fotouhi, C. Lee, R.Taylor, E. Sinibaldi, N. Navab, "Dual-Robot Ultrasound-Guided Needle Placement: Closing the Planning-Imaging-Action Loop," Unpublished Manuscript.

3. O. Zettinig, B. Fuerst, R. Kojcev, M. Esposito, M. Salehi, W. Wein, J. Rackerseder, B. Frisch, N. Navab, "Toward Real-time 3D Ultrasound Registration-based Visual Servoing for Interventional Navigation," IEEE International Conference on Robotics and Automation (ICRA), Stockholm, May 2015.

4. B. Fuerst, J. Fotouhi, and N. Navab. "Vision-Based Intraoperative Cone-Beam CT Stitching for Non-overlapping Volumes." Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015 (2015): 387-95. Web.

5. Z. Zhang. "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence IEEE Trans. Pattern Anal. Machine Intell.* 22.11 (2000): 1330-334. Web.

6. The MathWorks, Inc. "Find Camera Parameters with the Camera Calibrator." 1 Mar. 2013. Web. <http://www.mathworks.com/help/releases/R2013b/vision/ug/find-camera- parameters-with-the-camera-calibrator.html>

7. J. Bouguet. "Camera Calibration Toolbox for Matlab." *Camera Calibration Toolbox for Matlab*. 14 Oct. 2015. Web.

8. M. Shah, R. D. Eastman, and T. Hong. "An Overview of Robot-sensor Calibration Methods for Evaluation of Perception Systems." Proceedings of the Workshop on Performance Metrics for Intelligent Systems - PerMIS '12 (2012). Web.

9. R. Tsai, and R. Lenz. "A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/eye Calibration." IEEE Trans. Robot. Automat. IEEE Transactions on Robotics and Automation 5.3 (1989): 345-58. Web.

10. H. Bay, T. Tuytelaars, and L. Van Gool. "SURF: Speeded Up Robust Features." Computer Vision – ECCV 2006 Lecture Notes in Computer Science (2006): 404-17. Web.

11. X. Zhang, S. Fronz, and N. Navab. "Visual marker detection and decoding in AR systems: A comparative study." Proceedings of the 1st International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2002.


**Acknowledgements**


We thank our mentors Bernhard Fuerst, Risto Kojcev, and Javad Fotouhi for the guidance and insight they have provided us. We additionally thank Ashkan Khakzar, a master's student also working with the CAMP lab, who gave us invaluable assistance during this project.