

# Camera-based Shape Sensing and Motion Capturing of Tendon-driven Continuum Robots

Yasmeen Hmaidan, Priyanka Rao, Jessica Burgner-Kahrs

*Continuum Robotics Laboratory*

*University of Toronto, Mississauga*

**Abstract**—Continuous real-time estimation and recognition of a Tendon-driven Continuum Robot’s (TDCR) backbone whilst performing minuscule surgical procedures requires robust sensing technology. Unlike traditional robots, continuum robots do not depend on rigid joint-link composition and instead are inspired by biologically compliant structures like elephant trunks and worms. However, by offering more organic movement in an environment, their deformability and continuous curve morphology presents several recognition challenges in a surgical task space. As such, this research project aims to overcome this obstacle by relying on an external multi-camera image-based sensing approach to continuously recognize and estimate a TDCR’s shape in real-time.

**Index Terms**—real-time estimation, shape sensing, motion capturing, medical robotics, tendon-driven continuum robot (TDCR)

## I. INTRODUCTION

Burgner-Kahrs et al. [1] define continuum robots as “an actuatable structure, whose constitutive material forms curves with continuous tangent vectors”. Unlike traditional robots, they lack joints and rigid-links along their continuous backbone. This inherent flexibility feature enables continuum robots, as shown in fig. 1, to be efficiently used for performing delicate tasks in unstructured and complex environments like airplane engine inspections and microscopic surgery. Their environmental compliance, miniature size, and obstacle maneuver locomotion distinguishes them from traditional robots who have much more limited DOFs (degrees of freedom) in a target task space [10].

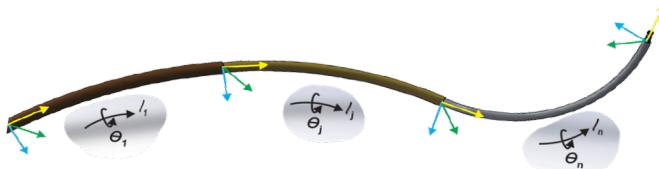


Fig. 1. A Continuum Robot (CSC476, 2020)

There are several different types of continuum robots that are extrinsically and intrinsically actuated. Tendon/wire-actuated as shown in fig. 2 and telescoping pre-curved tubes in fig. 3 are extrinsically actuated [1].

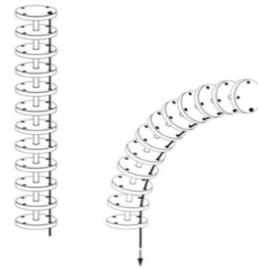


Fig. 2. A Tendon-driven Continuum Robot (CSC476, 2020)

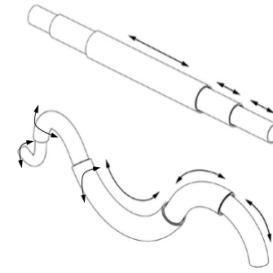


Fig. 3. A Telescoping Pre-curved Tube Continuum Robot (CSC476, 2020)

Flexible Pneumatic Actuators (FPAs) as shown in fig. 4 and Fluidic elastomer actuators (FEAs) in fig. 5 are intrinsically actuated. Functionally, FPAs expand, contract, or extend as a function of change in their internal pressure depending on the type’s material. Whereas, FEAs are composed of a pneumatic channel network in which an elastomer expands when filled with a pressurized fluid [1].

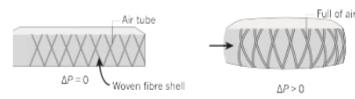


Fig. 4. A Flexible Pneumatic Actuator (FPA) (CSC476, 2020)

The focus of this research project is intrinsically-actuated, Tendon-driven Continuum Robots (TDCRs) and their shape-prediction necessity for a variety of applications like path planning, control, etc. Current TDCRs used in surgery need

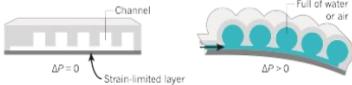


Fig. 5. A Fluidic Elastomer Actuator (FEA) (CSC476, 2020)

real-time estimation of their shape and location in 3D space, especially when navigating tight spaces like brain vessels in neurosurgery. However, existing shape sensing solutions are insufficient because they are either too large, time consuming, not contactless, or expensive. Moreover, current robotic models are not reliable due to unmodelled effects of friction, assembly and manufacturing errors [10]. Therefore, it is essential to estimate the backbone shape in real time using sensors. Some of those options include, embedded sensors (ie. fibre-optic or electromagnetic sensing) and external sensors (ie. mechanical or laser probes).

Composed of geometric fibres and a sensor array, fibre-optic sensing (shown in fig. 6) does output real-time shape information, but it is very expensive and not a scalable option for affordable robotics-based solutions.

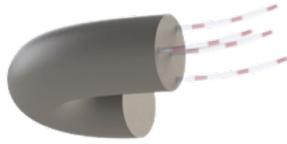


Fig. 6. Fibre-Optic Sensor (CSC476, 2020)

The other embedded sensor type, electromagnetic sensing (shown in fig. 7), also offers real-time pose tracking with sensor coils. However, despite not introducing any line-of-sight restrictions to estimating the shape of the robot, it introduces rigidity to the continuous structure of a TDCR.



Fig. 7. Electromagnetic Sensor (CSC476, 2020)

Thus, despite providing real-time backbone estimation, embedded sensor options like these are rendered invalid because they are unaffordable or restrictive to a continuum robot's organic movements.

Besides embedding sensors onto the robot's directly, external ones can be explored. For instance, a mechanical probe (shown in fig. 8) can measure multiple distinct locations along a TDCR's backbone. However, it must make contact the robot to measure its tip position or shape each time, making it not a contactless solution. Thus, this property is not realistic for

miniature continuum robots in tight vessels only millimeters thick and should be avoided.



Fig. 8. Mechanical Probe Sensor (CSC476, 2020)

On the other hand, laser probes can emit laser lines that use images from the camera along the robot to get object distance without direct contact. This technique can then generate a dense point cloud of the robot's shape but is time-consuming. Whilst it may seem promising, time and efficiency are one of the key components in using medical robotics for complex procedures, so a better solution is warranted for. Fig. 9 shows a laser probe in action.



Fig. 9. Laser Probe Sensor (CSC476, 2020)

After considering all these shape-sensing methods, the ideal solution was concluded to be external image-based sensing for its overall efficiency and safety. As such, this research project implements calibration and computer-vision algorithms for an existing multi-camera system to capture the motion and sense the shape of Tendon-driven Continuum Robots (TDCRs) in real-time.

## II. MULTI-CAMERA SYSTEM

To investigate this research project, a TDCR placed in the middle of a metal camera apparatus was used with a cardboard plaque to limit light reflection from the glossy green screen. The 19.5cm long TDCR was composed of 14 discs, 2 mm each. Four, 5mm ArUco markers were also placed on every fourth disc along the robot's backbone as shown in fig. 10(b). Initially, each disc contained an ArUco marker but the miniature scale of the robot proved too crowded for realistic marker detection so four were chosen. The TDCR was mechanically configured to an upright position with a slight bend forward to emulate an organic shape in medical application. Simultaneous image capture was conducted from three FLIR BlackFly S cameras horizontally positioned 4cm apart, with no differences in vertical translation. The entire setup is shown in fig.10.



Fig. 10. The Multi-Camera System

Each camera was connected to the Spinnaker SDK's GUI, SpinView, where images were taken and labelled for future processing and keyframe selection. Several image datasets were curated in this multi-camera system to output the most optimal camera calibration and ArUco marker detection scores. As such, out of 10 image datasets, composed of 50 images each with three camera angles, only one was sufficient. Furthermore, several techniques were discovered to reduce distortions in this sensitive camera system. For instance, to avoid camera discolouration, a blank piece of paper was placed in front of each camera before a photo was taken.

### III. METHODOLOGY

To model a TDCR's 3D world coordinate space, the image processing pipeline, indicated in fig. 11, is composed of a multi-tiered workflow based in Python and OpenCV. Namely, after image collection from the three cameras, an algorithm outputs the extrinsic and intrinsic camera parameters for each, as well as undistorts them. Next, distinctive image features, mainly the robot's backbone via distinctive ArUco markers, are highlighted using SIFT (Scale-Invariant Feature Transform) detection. Then, the most overlapping key point matches in an image between the three camera angles are drawn using the FLANN (Fast Library for Approximate Nearest Neighbors) feature matching algorithm. Shortly after, those keypoints are matched on the same horizontal epipolar lines to establish 3D relationship pairs. Then, the three camera images are rectified to a common plane with matching key points. The ArUco markers along the TDCR's backbone are then detected. Lastly, the 2D ArUco coordinates are converted to 3D world coordinates using a robust triangulation algorithm.

### IV. KEYFRAME SELECTION

When using a computer vision-based solution to contactless image sensing, data collection and key frame selection becomes pivotal in computing real-time pose estimation results close to the ground truths. As such, there are key guidelines to follow when deciding the location of the cameras and curating an image dataset (akin to a live video feed) to obtain the best image-sensing results. Those considerations include the baseline distance of the multi-camera system and the area overlap between the captured image task space. Namely, the baseline distance in 3D between the camera frames should be

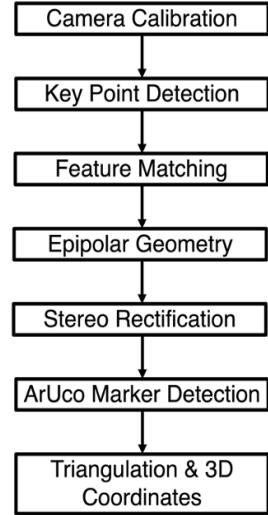


Fig. 11. The Image Processing Pipeline

at least 4 cm. As such, the three FLIR Cameras were measured with a ruler and separated by 4cm on the same horizontal plane [5]. Vertical translation was targeted to be consistent but note, there were errors with the camera's holder. So it introduced some inaccuracy, despite the image's orientation being corrected for in the stereo rectification process later on. The result is that the three sets of image frames appear as if they have been taken only with a horizontal displacement. This small, but important, experimental apparatus consideration simplifies calculating the disparities of each pixel later on. Next, the 3D image's depth and coordinates can only be computed for areas that are overlapped in all images. With trial and error, the minimum seems to be a 40% overlap for results to converge in pixel accuracy. Although, a larger overlap percentage is most ideal [5]. The camera angle images were tested in an online image overlapping website and were found to have image similarity scores of: 45.75% (cameras 1 & 3), 52.01% (cameras 1 & 2), and 57.97% (cameras 2 & 3) respectively. Thus, it is important to note the aforementioned guidelines of an established baseline distance and visual task space overlap when selecting for the most optimal key frame orientation in a multi-camera system.

### V. CAMERA CALIBRATION

When trying to make sense of a 3D coordinate system based in the real-world, it is key to consider that cameras distort images when not parallel to the imaging plane. These distortions heighten when several cameras are involved. Most commonly, radial and tangential distortions occur where straight lines in the image appear curved and some areas in the image look nearer respectively. This phenomenon can be observed fig. 12. Thus, OpenCV's chessboard camera calibration, show in fig. 16, and undistortion algorithms were used alongside the `calibrateCamera()` function to output the optimal camera parameters in the multi-camera system image's pre-processing pipeline. The output is shown in fig. 13.

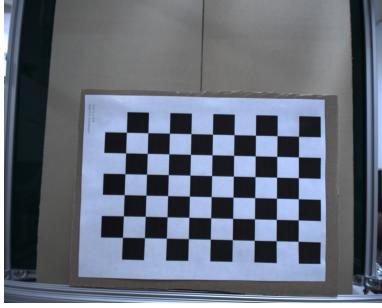


Fig. 12. Distorted Camera Image

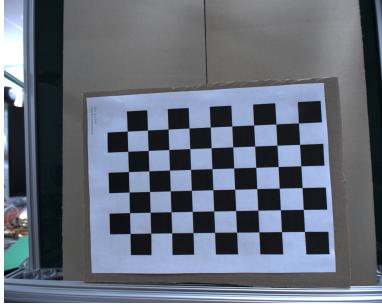


Fig. 13. Undistorted Camera Image

More specifically, in intrinsic calibration, given a point  $p = (x, y, z)$  in a camera frame, the image coordinates were calculated from the top of the frame with respect to the camera center (principal point) and distance to image plane (focal length). Then, a Perspective Projection Model (PPM) maps  $p$  to image coordinates  $(u, v)$  outputting the intrinsic camera parameters, shown in fig. 14.

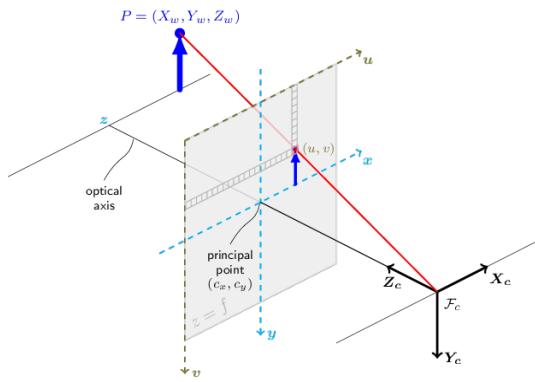


Fig. 14. A Perspective Projection Model (PPM) (OpenCV, 2021)

For extrinsic calibration, the algorithm maps where the objects are in the camera frame's task space and then maps it to the TDCR's frame with rotational translation like fig. 15. This process outputs the extrinsic camera parameters and as such, one knows where the several cameras are relative to the TDCR in real-time.

Considering all these factors, effective camera calibration was found to output accurate estimates of objects in the 3D

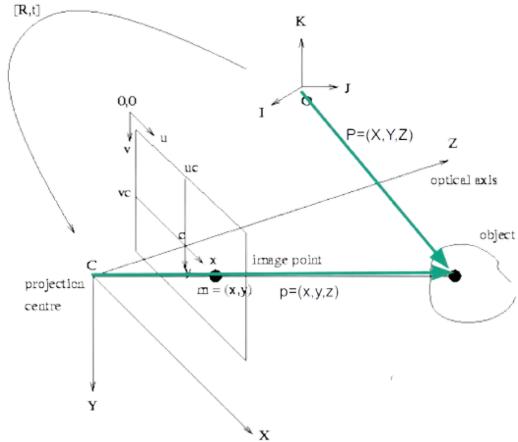


Fig. 15. Extrinsic Camera Frame Locality (OpenCV, 2020)

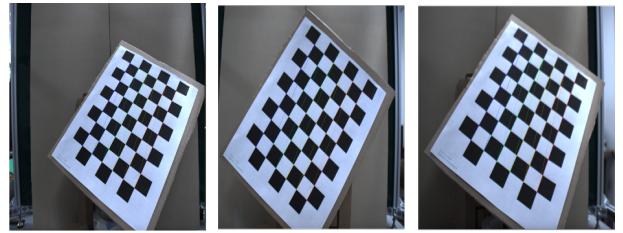


Fig. 16. Multi-Camera System Chessboard Calibration

world task space and where the TDCR is in that environment with no blind spots and image distortion discrepancies.

## VI. PREPROCESSING

### A. Camera Image Comparisons

By capturing the 2D projection of the TDCR's 3D task space via three cameras, discrepancies are bound to happen. Therefore, despite taking a picture of the same TDCR task space, the three different viewpoints' captured pixels are not only translated to the left or right. Instead, the image pixels transform in several ways, as well as scale and rotate differently. These differences are important to note when preprocessing images for the future purpose of authentic key point detection. For instance, the TDCR's features (ie. the ArUco marker) may appear to be placed higher on a disc in one camera angle versus another when outlined by a horizontal line. This difference is later rectified but establishes an informed outline of the original camera images' keypoint relationships and orientation differences as shown in fig. 17.

### B. Key Point Detection

Now, to detect key features in the cameras' TDCR images, one can consider several key point detection algorithms commonly used, like Harris Corner Detection. However, Harris and others are rotation-invariant, meaning if an image is rotated, one cannot detect the same corners. This property can be very troublesome in a surgical context where vessels and the TDCR are continuously moving. Especially since corners

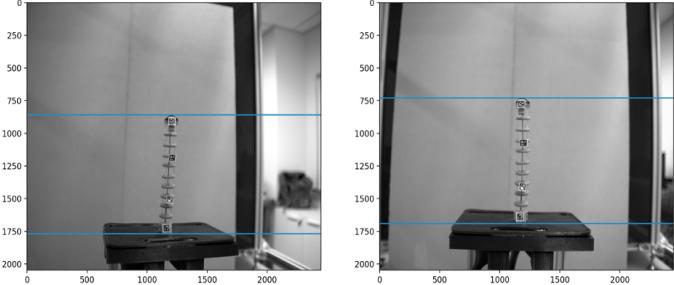


Fig. 17. Original Image Comparisons between cameras 2 and 3

remain corners in rotated images as well. In addition, an ArUco marker's corner may not be a corner if the image is scaled. A corner in a small image within a small window is flat when it is zoomed in the same window. Thus, Harris corner detection is not scale invariant and should be avoided [6]. Thus, SIFT (Scale-Invariant Feature Transform) detection becomes ideal in this context because the multi-camera system images' varying scale and rotation are accounted for in the target task space. Fig. 18 shows the detected key points in the three camera images.



Fig. 18. Key points detected by the SIFT algorithm

### C. Feature Matching

After detecting key points individually for each image outputted per camera, the matching pairs are computed and visually outlined by a green line. One should note that there will be slight differences in the visibility of the detected key points because of the the horizontal translation perspective differences. So before performing stereo rectification, the camera's key points are matched among one another. By introducing these direct comparisons, the key features present in both images will be outputted, with their position differences highlighted. The chosen algorithm was the FLANN (Fast Library for Approximate Nearest Neighbors) matcher [5]. It's efficient in feature matching because it classifies the best potential matches between similar key points in both camera image frames based on their distance using a K-Nearest-Neighbor (k-NN) search. In this context, the k-NN classification's output is ideally aimed at class membership of the ArUco markers. As such, a key point is classified by a plurality vote of its neighbors, ideally with the properties of an ArUco marker's features. If so, the key point will be classified as such to the most common among its  $k$  nearest

neighbors, where  $k$  is a small positive integer [7]. After using this algorithm, the image's contents provided more key point matches than the needed ArUco markers. Following that, a filtering image pixel slicing mask was added to decrease the number of unrelated matches (ie. camera apparatus rails, light beams, etc). At this stage, the 2D key point matching amongst the three camera angles has been completed as shown in fig. 19 and 20 and is ready for epipolar geometry compute.

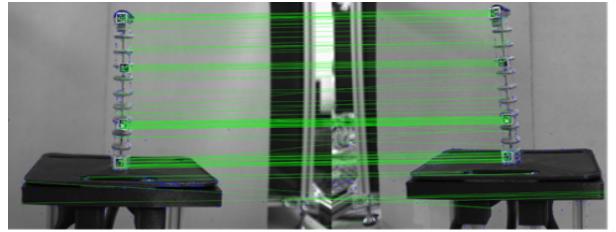


Fig. 19. FLANN Feature Matching between cameras 1 and 2



Fig. 20. FLANN Feature Matching between cameras 2 and 3

## VII. EPIPOLAR GEOMETRY

After 2D key point matching via feature matching, one must examine the cameras' images in a 3D relationship. This epiline (epipolar line) mapping can be accomplished with epipolar geometry and the fundamental matrix. This matrix describes the relationship between several images in the same TDCR scene. It can be used to map points of one image to lines in another [5]. By establishing these epipolar relationships, along a multi-camera system, one fundamentally corrects sensitive robot pose estimation. This feature becomes especially important when a robot is in a bent orientation and can have ambiguous disc orientation in a medical task space. The corresponding epipolar lines between cameras 1 and 2 and 2 and 3 are shown in fig. 21 and 22 accordingly.

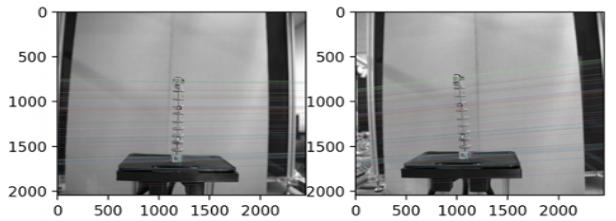


Fig. 21. Epipolar geometry in features between cameras 1 and 2

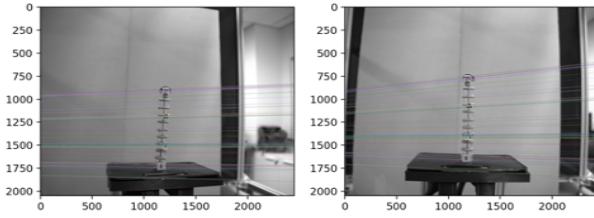


Fig. 22. Epipolar geometry in features between cameras 2 and 3

### VIII. STEREO RECTIFICATION

After outputting the epilines, matching them amongst the varying camera angles is challenging as they will be angled. As such, the stereo rectification process re-projects the multiple image planes onto a common plane parallel to the line between the camera centers. Methodically, the fundamental matrix allows one to compute the epipolar line in the second image given a point in the first image, enabling an undistortion feature as well [3]. This process is scaled to the three-camera system.

Using the built-in OpenCV *findFundamentalMat()* function, the fundamental matrix was calculated based on the matched ArUco marker key point pairs. The algorithm itself requires a minimum of 7 pairs and so the four arUco markers each outputted 4 pairs, resulting in 16 feature pair matches ideal for plentiful data use.

In addition, the RanSaC (Random Sample Consensus) method was used to consider that not all matched features are reliable, regardless of the feature mask.

Thus, to add another level of robustness, it takes a random set of point correspondences to compute the fundamental matrix. Following that process, the matrix's performance is checked in contrast to other versions concurrently run for 8 other different random set variations. The best TDCR estimate amongst them is then outputted and chosen by the RanSaC algorithm.

Thus, after stereo rectification, all the epipolar lines in the multi-camera system's images are parallel to the horizontal axis of the image's base. This correction proves pivotal for the upcoming triangulation stage, where each pixel from one image is easily matched with the same pixel in the corresponding image because of the epipolar lines' guidance. The rectified image outputs are shown in fig. 23 and 24.

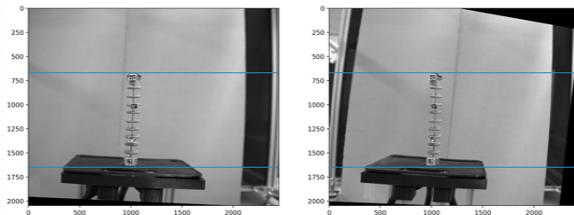


Fig. 23. Stereo rectified images from cameras 1 and 2

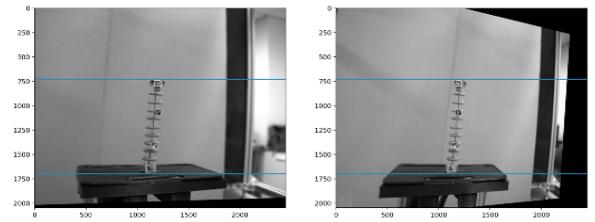


Fig. 24. Stereo rectified images from cameras 2 and 3

### IX. ARUCO MARKER DETECTION

ArUco markers are a type of synthetic QR code with an inner binary matrix for identification and black border for fast image detection. Using these markers will be necessary when evaluating real-time pose estimation for a TDCR because it will detect a collection of detected markers along the robot's backbone. The ArUco markers thus play a pivotal role in outputting the live coordinates of the TDCR, amongst all three camera angles. This flagging property plays an important environmental role in the upcoming triangulation requirements where projections points in each camera are necessary to retrieve. Before deciding to use this marker method, another commonly-used visual fiducial system in robotics was considered, April Tags. Serving the same property as ArUco markers, its main features included fewer tuning parameters, long distance compatibility, and less computation. However, its cons included limited opencv implementation, more steps to obtain April Tag markers, and a higher false detection rate. On the contrary, ArUco markers had more robust opensource OpenCV implementation guides, an available marker generator, and a better option for smaller scale object detection. However, detecting ArUco markers on each TDCR disc proved to be more computationally intensive and required tinkering tuning parameters beyond default mode [4]. Overall, ArUco markers were the most promising option for continuous pose estimation tracking. Though, it's important to note that the TDCR's small size posed considerable challenge for marker detection in contrast to larger, real-life sized items and should be factored accordingly. The detected markers have been shown in fig. 25 for the three camera images.

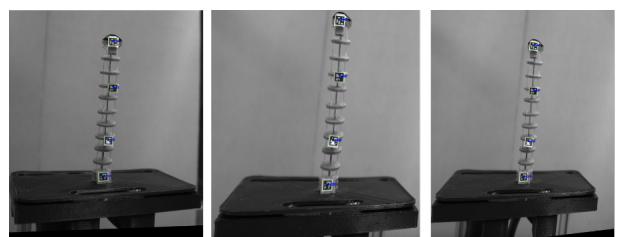


Fig. 25. Detected ArUco Markers along the TDCR's Backbone

### X. TRIANGULATION & 3D COORDINATES

In the last stages, the TDCR's detected 2D ArUco coordinates, existing in the cameras' local coordinate system,

should be converted to the 3D world coordinate system via triangulation. The OpenCV *triangulatePoints()* function reconstructs 3-dimensional points (in homogeneous coordinates) by using their observations with a stereo camera. Its input is composed of both cameras' projection matrices and points. Note that since the camera calibration function was used, the rotation matrix outputted is  $3 \times 1$  so it was converted to  $3 \times 3$  using OpenCV's Rodrigues method to meet the triangulation function's dimension requirements. Note that the Rodrigues transformation method was previously used to convert each camera's rotation matrix to a rotation vector in the shape transformation because it is the most convenient representation of the matrix as it is compact. This property is important when implementing global 3D geometry optimization operations like camera calibration [9]. The obtained 3D coordinates for a single marker are shown in fig. 26.

After the outputted 3D points are computed from triangulation, the euclidean distance ( $E_d$ ) is calculated [8] amongst each ArUco marker's coordinates, in a clockwise manner (shown in fig. 27). Using the following equation:

$$E_d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

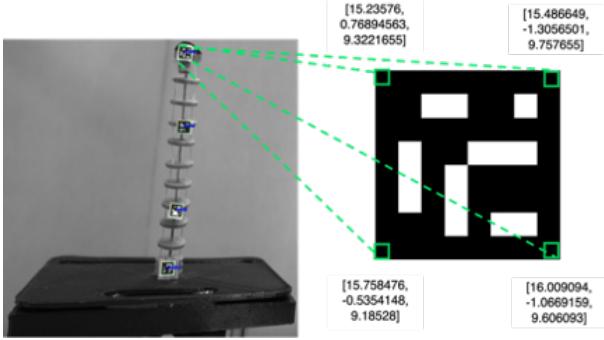


Fig. 26. The ArUco Marker Pixel Coordinates

## XI. RESULTS

Following the completion of the three-camera-system's triangulation, an accurate model of a TDCR is successively modeled in the 3D world coordinate space with an outputted matrix for each detected ArUco marker's coordinates. The (x,y,z) pixel points of each ArUco marker along the robot's backbone is outputted and matched with the ground truth measurements, outlined in the multi-camera system. In contrast to the 5x5mm ArUco marker, the computed Euclidean Distance ( $E_d$ ) from the outputted coordinates have a  $\pm 2$  mm error. This result was concluded after calculating each side of the four ArUco markers yielded unsimilar lengths. Noting that each calculated side length was not the same, especially if the ArUco marker in question was bent or on an angle. This quantitative indiscrepancy can be attributed to several error-inducing factors, which can be optimized, like incomplete camera calibration, environmental noise, and parameter tuning. Overall, the outputted coordinates were sufficient in deducing

the real-time pose and shape estimation of the TDCR via a multi-camera system.

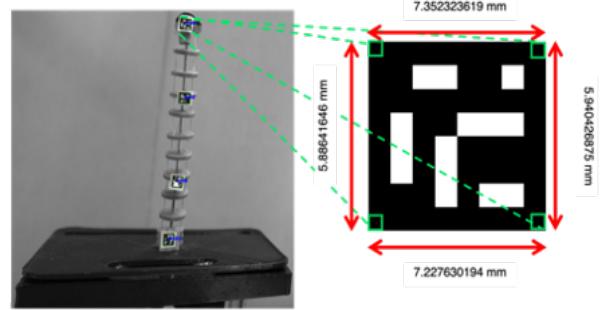


Fig. 27. The Distance (mm) between each ArUco Marker's Corner Pixel Coordinates

## XII. DISCUSSION

After completing this research project, an efficient use of calibration and computer-vision algorithms were implemented for an existing multi-camera system to capture the motion and sense the shape of Tendon-driven Continuum Robots (TDCRs) in real-time. This image sensing pipeline was composed of camera calibration, key point detection, feature matching, epipolar geometry, stereo rectification, ArUco marker detection, and 3D coordinate triangulation. Despite this robust workflow, several challenges were introduced in this research project's experimental apparatus and should be considered in future implementations. Those components included camera type, configuration, environmental control, and miniaturization. As a result, a stereo vision system was adapted at the end of compute because of a faulty camera. However, the aforementioned results can be achieved with a reliable, multi-camera system regardless.

As a future extension to this research project, the multi-camera system can be extended to  $n$  cameras by taking pairs of stereo image outputs and averaging them. If this configuration is considered, it would be ideal to use  $n/2$  stereo cameras instead of single cameras to reduce alignment errors. As without built-in support, the cameras moved throughout image collection and as such, following processes like camera calibration were hindered because of distortion. This small experimental apparatus feature later rendered future code involving undistortion algorithms ineffective because of image inconsistency. This issue can be observed in the images through radial and tangential distortion, where the experimental rail is bent and some areas of the TDCR are closer than they actually appear to the camera lenses. Having a binocular vision setup, similar to human eyes, via a stereo camera setup is ideal. Furthermore, camera specs should also be considered when running this experiment on a sensitive task space. Discolouration and light changes can introduce unwanted consequences like faulty detections by the key point finders and matching algorithms, despite outlining a specific region of interest. To control for these unwanted environmental

effects, executing this experiment in a photo box would also be recommended. Some improvements can also be made along the TDCR's pose estimating by placing an ArUco marker along each backbone disc. Alternatives to ArUco markers and meta-markers, like concentric markers with barcodes and QR codes, can also be explored for better recognition [2].

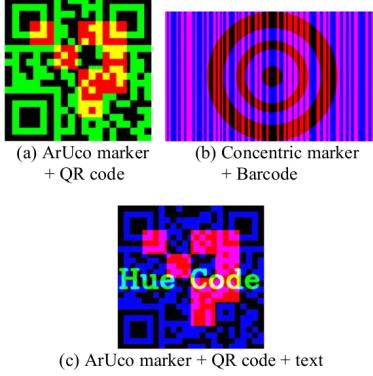


Fig. 28. Meta Marker Options (Okada et al., 2021)

In conclusion, these changes should help in reducing the small error margin of  $\pm 2$  mm in the measured ArUco markers' 3D coordinates. Thus, yielding the most optimal image-based sensing approach to continuously recognize and estimate a TDCR's shape in real-time with an external multi-camera system.

#### ACKNOWLEDGMENT

This Research Opportunity Project (ROP) was conducted under the Continuum Robotics Laboratory at the University of Toronto, Mississauga under the kind faculty supervision of Jessica Burgner-Kahrs and Priyanka Rao. A special thanks to Priyanka for being a phenomenal guiding force in this project's vision and execution. Additional thanks to Quentin Peyron, Reinhard Grassman, and Matthew Shamoon for helping in setting up the experimental apparatus at the lab, as well as lending a hand in active problem-solving. Finally, thanks to the other ROP and summer students for sharing this inventive experience with one another and completing it amidst a pandemic. Glad the future of continuum robotics is possible with all these phenomenal engineers.

#### REFERENCES

- [1] J. Burgner-Kahrs, D. C. Rucker and H. Choset, "Continuum Robots for Medical Applications: A Survey," in *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261-1280, Dec. 2015.
- [2] Y. Okada, D. Fujikura, Y. Ozawa, K. Tadakuma, K. Ohno, and S. Tadokoro, "HueCode: A Meta-marker Exposing Relative Pose and Additional Information in Different Colored Layers", in *IEEE International Conference on Robotics and Automation*, 2021 [preprint]
- [3] I. Gkioulekas, Computer Vision: Stereovision, The Carnegie Mellon Robotics Institute, Feb 26, 2020. Accessed on: Aug. 16, 2021. [Online]. Available: <https://www.cs.cmu.edu/~16385/lectures/lecture13.pdf>
- [4] A. Rufus, AprilTag vs Aruco markers, Robotics Stack Exchange, Jul 30, 2020. Accessed on: Aug. 16, 2021. [Online]. Available: <https://robotics.stackexchange.com/questions/19901/apriltag-vs-aruco-markers>
- [5] A. Jakl, Understand and Apply Stereo Rectification for Depth Maps, St. Pölten University of Applied Sciences, Dec 2, 2020. Accessed on: Aug. 16, 2021. [Online]. Available: <https://www.andreasjakl.com/understand-and-apply-stereo-rectification-for-depth-maps-part-2/>
- [6] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
- [7] OpenCV, Feature Matching with FLANN, Open Source Computer Vision, Mar 12, 2020. Accessed on: Aug. 16, 2021. [Online]. Available: [https://docs.opencv.org/3.4/d5/d6f/tutorial\\_feature\\_flann\\_matcher.html](https://docs.opencv.org/3.4/d5/d6f/tutorial_feature_flann_matcher.html)
- [8] B. Halabisky, Euclidean Distance In 'n'-Dimensional Space, Stanford University, Jan 19, 1999. Accessed on: Aug. 16, 2021. [Online]. Available: [https://hlab.stanford.edu/brian/euclidean\\_distance\\_in.html](https://hlab.stanford.edu/brian/euclidean_distance_in.html)
- [9] OpenCV, Triangulation, Open Source Computer Vision, Feb 3, 2022. Accessed on: Aug. 16, 2021. [Online]. Available: [https://docs.opencv.org/master/d0/dbd/group\\_\\_triangulation.html](https://docs.opencv.org/master/d0/dbd/group__triangulation.html)
- [10] Trivedi, D. et al., "Soft robotics: Biological inspiration, state of the art, and future research". *Applied Bionics and Biomechanics*, 5(3), pp.99–117. Oct. 2008
- [11] J. Burgner-Kahrs, Continuum Robot Types, Continuum Robotics Laboratory, Jan 19, 2021. Accessed on: Aug. 16, 2021. [Online]. Available: <https://crl.utm.utoronto.ca/crl/teaching/>