Characterization of contact interaction with an inflatable link robot

Margarida Fernandes Martinho
margarida.f.martinho@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

July 2021

Abstract

This work comprises a virtual simulation study of position and contact estimation on an inflatable structure, using computer vision and machine learning tools. The project developed is meant to be applied and validated on a prototype that will function as an additional link implemented on a commercial rigid robot manipulator. It builds on previous work, upgrading the internal pattern and the optical sensor used but not the structure geometry. The simulation environment was built employing the Blender software, with two different optical sensors: a single camera and stereo camera set. The images retrieved were submitted to an image segmentation algorithm to extract relevant features to be used as inputs to an artificial neural network that returns the bending curve and tip displacement. For the contact estimation, after the image segmentation step, an empirical algorithm, using the principal inertia moments information, computes the contact point and the direction. The performance of the bending estimation algorithm was very satisfactory for both single view and stereo view. As for the contact estimation, initial tests were promising but further study is required to fully establish the method.

Keywords: Soft Robotics, Proprioception, Computer Vision, Artificial Neural Networks

1. Introduction

Soft robots are intrinsically safe to interact with people [1] or fragile objects due to their lightweight, compliant and flexible nature. Accordingly, there is a lot of research to improve their control and to integrate them in traditional robotic manipulators.

This project builds on previous work with inflatable links [2], in which an inflatable link prototype was created, with a centered camera at the base, using the interior images to estimate, recurring to neural networks, the force applied at the distal end of the link, with and without rotation.

This work aims to improve the interior pattern and the image segmentation method and expand the link proprioception, estimating the bending curve and contact point and direction. There is also the intent of changing the sensor to a stereo camera set with a wider field of view and higher frame rate, so the algorithms must function for stereo images. The work developed is meant to be applied and validated on a prototype to be integrated as an additional link in a traditional robot.

Soft Robots allow for more freedom of movement, compared to traditional robots, adding degrees of freedom, which increases the complexity of control. The applications of soft robotics in engineering are very vast, since flexibility, freedom of movement and volume variance are extremely useful features in this context.

Inflatable links use internal pressure to maintain structural integrity [3] and have been used in the robotic domain to lower the contact stiffness of traditional robots, as inflatable sleeves for a rigid robotic skeleton [4], or as replacement robotic links [5][6].

As mentioned before, the significant disadvantage the soft robots have is the difficulty in control. As such, they rely on sensors [7], to get information on the system and to develop control strategies.

Tactile sensing is evolving to recreate the human touch sensibility in soft robots. There are very interesting investigation works on this topic, such as the TACTIP device, which is uses internal pins to assess the surface deformation [8]. This technology has been transferred to other geometries, to accommodate different needs [9]. In [10], there is a detailed explanation of how to use markers on a cylindrical surface (TacCylinder) to characterize the objects that come into contact with it. A camera was place by one of the ends of the prototype and aligned with its central axis.

The knowledge of the position and deformation of the soft surface allows recreating the environment of interaction [11]. In that way, it is important to create and improve the tools to comprehend the internal state of the object.

Proprioception is the understanding of the body of its internal state. It is crucial to have reliable information from the robot itself if it is to be used outside of monitored environments. Research on proprioception might aim to track the position of a single point, usually the extremity [12], or to be aware of the deformation of the entire surface [13].

The authors of [14] present a method to measure the tip position of an inflatable link and binary contact detection, also using a built-in camera. The internal pattern consists of ellipse-shaped markers and the position and contact detection are measure using blob analysis and geometry.

In Werner et al. [12], the process of estimation of the three dimensional tip position of an inflatable actuator is described, using images from an integrated centered camera. The pattern applied uses scattered dots to gather bending information according to the light density of the image. This work continues in [15], using the interior pattern for three actuators. The rotational degrees of freedom of the arm are estimated using deep neural networks.

2. Background

2.1. Computational Vision

Image Segmentation

Segmentation methods were used in this project to identify and label areas of binary images, to get information on their geometry.

In order to have the binary image, the original was transformed into a grayscale image and then an intensity threshold, chosen through the Otsu method, was applied to the entire matrix, above which every pixel will take the value of one and below it, zero. The grayscale conversion is made through a weighted sum of the three RGB channels, following equation (1).

$$I = 0.2989R + 0.5870G + 0.1140B \tag{1}$$

Often in experimental environment the picture needs to be corrected in contrast so that important features are not lost in the process of binarization.

Blob Analysis

In a binary image, a blob is a connected group of white pixels, as the black pixels are perceived as background.

After the blobs were labeled, the information of area, centroid (2), principal moments of inertia and respective orientation(4) were obtained.

$$C_x = \frac{\sum_{i=1}^{n} X_i}{n} \tag{2}$$

$$C_y = \frac{\sum_{i=1}^{n} Y_i}{n} \tag{3}$$

Where n is the number of pixels in the blob, and X_i and Y_i are the x and y components of pixel i, respectively.

$$I_{\substack{max\\min}} = \frac{I_x + I_y}{2} \pm \sqrt{\frac{I_x - I_y}{2}^{2} + I_{xy}^2} \tag{4}$$

$$tan(2\theta_p) = \frac{-I_{xy}}{(I_x - I_y)/2} \tag{5}$$

Alternatively, it is possible to take the linear algebra approach and compute I_{min} and I_{max} using the eigenvalues of the matrix M depicted in equation 6.

$$\boldsymbol{M} = \begin{bmatrix} I_{xx} & -I_{xy} \\ -I_{xy} & I_{yy} \end{bmatrix} \tag{6}$$

The value of θ can be calculated through the eigenvectors of matrix M, by applying the inverse of the tangent.

Boundary Analysis

Alternatively, to avoid evaluating the blob in its entirety, boundary analysis was also tested.

The boundary was defined through the Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria[16].

Once the boundary is identified, it is possible to find the same geometric properties as for the blob analysis. The boundary approach does not correspond exactly to the geometry of the blob, but it might be an adequate alternative.

After image processing, the data was used as input for developed algorithms for bending estimation, using artificial neural networks, and for contact point and direction evaluation.

2.2. Artificial Neural Networks

The configured NN was a completely connected feedforward network typically used in curve fitting problems and regressions, created through the MATLAB® function *fitnet*. The NN was defined as having one hidden layer and a linear output layer.

The NN trains under Supervised Learning, modifying the weights of the node connections based on examples of input-output pairs, the training set, using the error as feedback information for the performance status.

The data for training is split into three categories: training - to update the weights of the NN -, validation - to avoid overfitting of the data - and test - to evaluate the NN performance.

The training function chosen was the Levenberg-Marquardt backpropagation, a method for detecting the minimum value of the error. It is a very fast algorithm which is an advantage in a repetitive process like training. The algorithm is considered to have attained convergence when the gradient of the performance function is smaller than a user-predefined number.

3. Simulation Setup

The software chosen for the simulations was Blender, which is more suitable than a finite element analysis software since it must produces a visually realistic behaviour of an inflatable link and enables customization of camera properties.

The virtual link created is exhibited in figure 1, with the different components and measurements discriminated. The overall appearance, size and geometry were kept from the previous work[2], only simplified, as there is no need for an overlapping of the surfaces to close the sleeve. The changes will occur mainly at the internal pattern and the optical sensor.

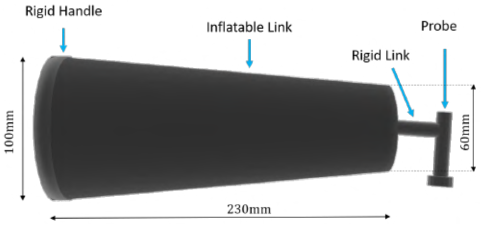


Figure 1: Components and Dimensions of the Simulation Link

The virtual link was modelled as a cloth object with the material rubber and some values were then adjusted, such as weight and internal pressure. The internal pressure feature was not used for the bending simulation, only for the contact simulation, since it involves collision with other objects and how the link responds to collisions will heavily depend on how pressurized it is.

3.1. Bending Simulation

The bending motion was achieved applying a displacement at the extremity of the link, keeping the base at the same place - a fixed beam with a free end. The position of the other parts is defined by the software itself, through the use of inverse kinematics.

The virtual link is controlled by a structure (armature) that encloses its physical properties. It is at the armature that the displacements are directly applied, and it is according to the movement of the structure that the virtual link will then move. The armature will correspond to the centerline of the soft link, from where the reference position values will be taken.

The centerline was divided into six parts, to guarantee a smooth bending curve, blocking the rotation of the last segment relatively to the two of the rigid body.

3.2. Contact Simulation

The goal of the contact simulation was to analyse the effects of surface interaction at the internal view. Some cylindrical objects were created to collide with the virtual link while it followed a predetermined trajectory. For this simulation, it was important to recognise how the interior markers deform with the contact with other objects and if it was possible to identify the location and direction of the contact.

The reference values of the contact location were retrieved computing the location of the object relative to the base of the virtual link.

3.3. Patterns

The interior pattern of the link must provide enough information to reconstruct the bending curve. An easy way to do this is to use ring-like markers, using their centroids to estimate the curve.

Three patterns were evaluated: the Original Work Pattern - rings covering approximately 40% of the link (used mostly for comparison purposes) -, the Striped Pattern - twelve rings covering the entire link, equally spaced and with the same thickness - and the Checkered Pattern- checkerboard-like pattern with the same amount of rows as the striped one. The selected pattern was the Striped Pattern, portrayed in figure 2.

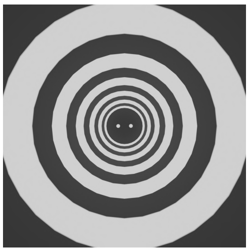
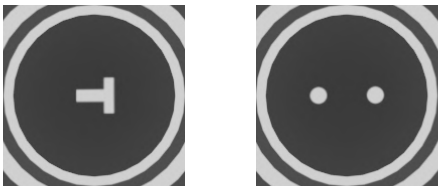


Figure 2: Striped Pattern

3.4. Distal Extremity Marker

The Distal Extremity Markers have two purposes: identify the position and the rotation of the end piece of the link. The markers have to be determined in a way that allows that specific information to be retrieved.

There were two distinct types of markers used: T-shaped marker - where the centroid and the moment of area indicate the position and the rotation of the centre of the piece, respectively 3(a) - Two circular markers - where the combined centroid of both circles shows the position and the centroids of each present the rotation, 3(b).



(a) Original Cover (b) Dotted Cover

Figure 3: Inside View Cover

When the simulation goal is solely to find the position (in a pure bending motion), the type of cover marks does not affect the result, as the centroid of the T and the combined centroid of the circles are designed to be at the centre of the cover.

In the end, the markers chosen were the two circles, since they allow for an implementation using only the centroids of each marker as opposed to needing the entirety of the boundary.

3.5. Camera Properties

For this project, two camera sets were used: a single camera centred at the base of the virtual link (equivalent to the previous work configuration, with a Trust SpotLight Pro Webcam[1]) and a stereo set, mimicking the Leap Motion Controller™ sensor that will be used when recreating the simulation in an experimental environment.

The relevant properties used for the simulation of this sensors are presented in tables 1 and 2.

$Camera\ Property$	$Value$
Maximum Resolution	640 × 480
Focal Length / Field of View	120^o
F-stop	$f/11$

Table 1: Camera Properties - Single Camera

$Camera\ Property$	$Value$
Maximum Resolution	640 × 240
Focal Length / Field of View	150^o
F-stop	$f/11$

Table 2: Camera Properties - Stereo Set

3.6. Lighting Conditions

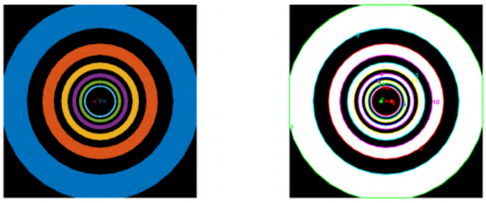
It was decided to even out the lighting in the simulation, avoiding shadows and other obstacles to image processing. This was achieved by removing all light sources from the simulation environment and making the background white. That way, it is as if the entire background is the light source and the virtual link is receiving light from every direction homogeneously.

4. Image Processing

After the binarization of the images, image segmentation follows. Two different image segmentation methods were defined at this phase to be compared on the impact of the bending curve estimation performance: Blob and Boundary Detection.

Blob Detection finds and labels all the connected components in the image. Boundary Detection returns the contour of the blobs. Both methods then proceed to compute geometric features, such as centroid, area and moment of inertia.

Figure 4 portrays both of the methods, for the Striped Pattern. In Blob Detection, both binary and complementary images must be segmented, as only the white blobs, with value 1, are identified as objects.



(a) Blob Detection (b) Boundary Detection

Figure 4: Segmentation Methods - Striped Pattern

In table 3, the average computation time required for each method is presented, with and without display of the image. The boundary segmentation is significantly faster, specifically when not showing the image assessed.

$Method$	$ImageDisplay$	$Time(s)$
Blob	Yes	1.2299
Blob	No	0.8290
Boundary	Yes	0.5375
Boundary	No	0.04632

Table 3: Computation Time - Striped Pattern

Both methods provide the area within the boundary or the area of the blob, and the location of their centroids. This is the information that will be used to compute the deflection of the inflatable link. For the contact simulation the moments of inertia in the principal axis of each blob or boundary are computed, as well as their direction.

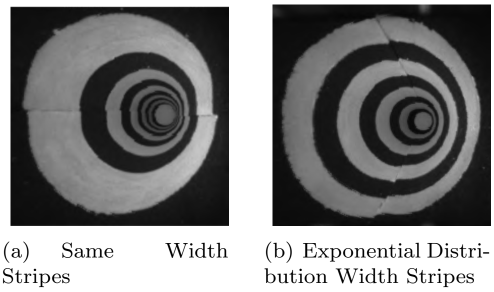
4.1. Experimental Work

When transitioning to the experimental work, a new pattern was created, the Exponential Distributed Striped Pattern, with stripes of different thickness, growing thicker with the distance from the base, so

[1] https://www.trust.com/en/product/16428-spotlight-pro-webcam-with-led-lights

they are perceived to have the same width by the camera.

Figure 5 shows the comparison of the two patterns.



(a) Same Width Stripes

(b) Exponential Distribution Width Stripes

Figure 5: Comparison of Patterns

The process of treating the images retrieved from the experimental setup is slightly different mainly due to the lighting conditions. There is a need to even out the lighting of the image before making it binary, at the risk of losing the markers distant from the camera. The method used was an adaptive histogram equalization (CLAHE) with a single threshold. This method divides the image into 8 by 8 tiles and applies an histogram equalization to each, while smoothing the transitions between neighbouring tiles, with a bilinear interpolation - shown in figure 6.

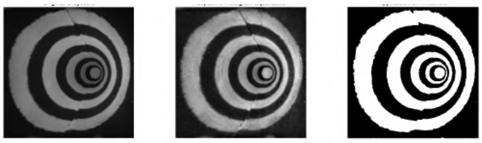


Figure 6: Binarization Process Chosen

After having found a suitable binarization method for the lighting conditions, the rest of the segmentation and estimation process follows the same steps as the other pattern.

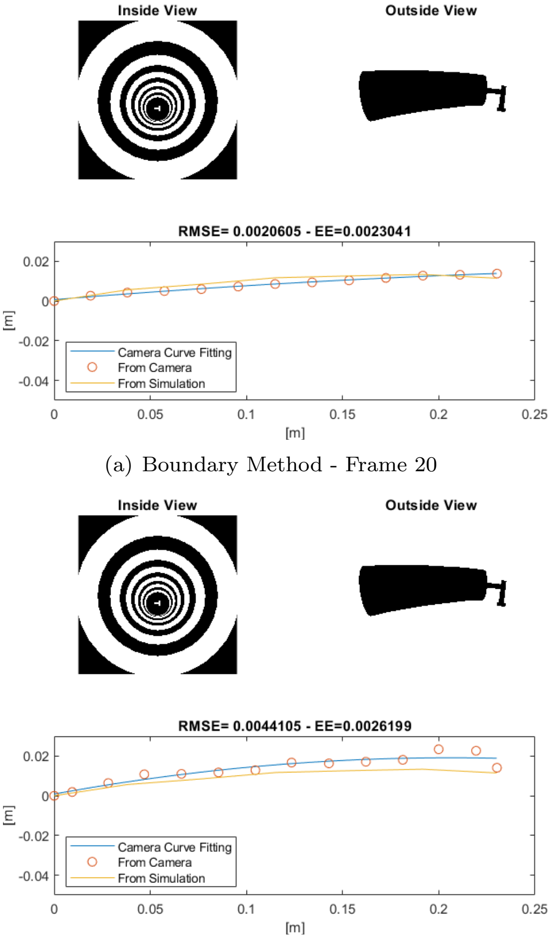
5. Results

5.1. Bending Problem

The algorithm developed for bending estimation is presented in algorithm 1.

This process was repeated for both segmentation methods and the results can be examined in figure 7. The deflection curve is a second order polynomial curve fitting, with the estimated points identified as markers, over the curve from the simulation software.

```
Algorithm 1: Bending Estimation Algorithm
Input: Position of Centroids ($C_i$), in pixel,
       and angle of camera ($\alpha_i$)
Output: Deflection
1. Compute displacement of centroids,
   considering the initial position as resting
   pose:
   $\Delta y = (C_i - C_1)$
2. Apply pixel to meter conversion
   previously found using a known reference:
   $\Delta Y = 1.7920 \times 10^{-4} \times \Delta y$
3. Change to global reference frame:
   $Deflection = R_\alpha \times \Delta Y_i$
```



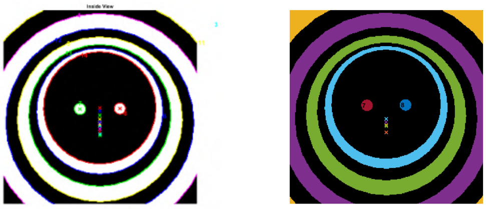
(a) Boundary Method - Frame 20

(b) Blob Method - Frame 20

Figure 7: Boundary vs Blob Performances

The performance of the algorithm will be evaluated by 3 factors: Root Mean Squared Error, RMSE, (measured using the second order polynomial curve adaptation of the points), Extremity Error, EE, (measured using only the last value) and computation time. This algorithm is very susceptible to errors when occlusion of markers exist as it loses information to work with and the information it does receive is distorted.

Boundary Detection offers a better result than Blob Detection. The reason for it is likely the distortion of the rings - as one side grows thicker and the opposite grows thinner, the blob centroid will shift towards the thicker side, 8(b). The distortion of the rings does not affect as much the boundary method, 8(a), since all the edges remain to be circles.



(a) Boundary Method (b) Blob Method

Figure 8: Close up on Centroids

In comparison with the previous work pattern, it is noticeable that the error at the tip was not improved with the new pattern as it still receives the same information. What is significantly better is the error along the bending curve, since the number of markers as increased.

5.2. Neural Networks

The purpose of using NN is to establish if it can compensate the occlusion of markers or small deviations from the segmentation and, overall, if the curve error can be reduced. The inputs chosen for the NN were the marker's Area, Centroid Position (vertical only), Position along length and Angle of camera.

The creation of the NN happens inside a training loop that increases the number of neurons of the hidden layer and compares the performance of each created network, always keeping the best one. 10% of the data is used for test and the rest is divided into training and validation sets, in a ratio of 85 to 15, respectively. The network trains with 113 input-output pairs out of 147. Since the training values are chosen randomly within the set and they greatly influence the performance, the NN trains several times with the same hidden layer size, only changing the data distribution for training and validation. This will guarantee the network several chances to find a better training set.

The best performance of the NN was 3.87×10^{-9} for 8 neurons. The mean performance of the NN for each hidden layer size is displayed in figure 9.

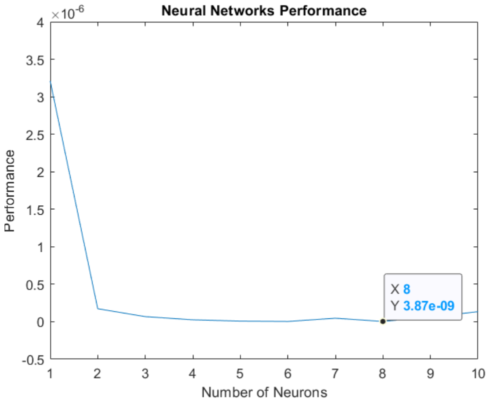
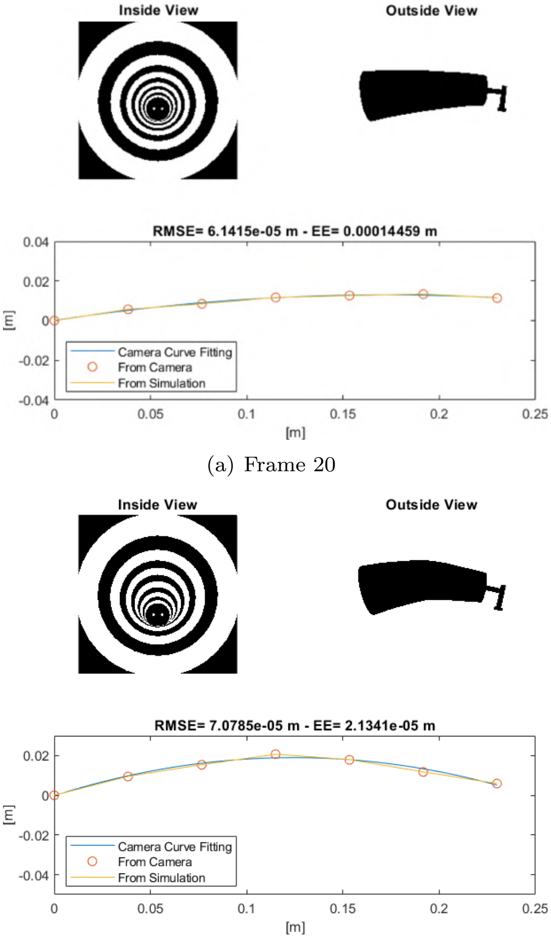


Figure 9: NN Performance Mean with Number of Neurons 1 to 10

The performance for this NN is in figure 10, particularly improving the estimation of the curve (RMSE). The deflection results shows that the NN approach can provide precise estimations even if there is occlusion of markers10(b).



(a) Frame 20

(b) Frame50

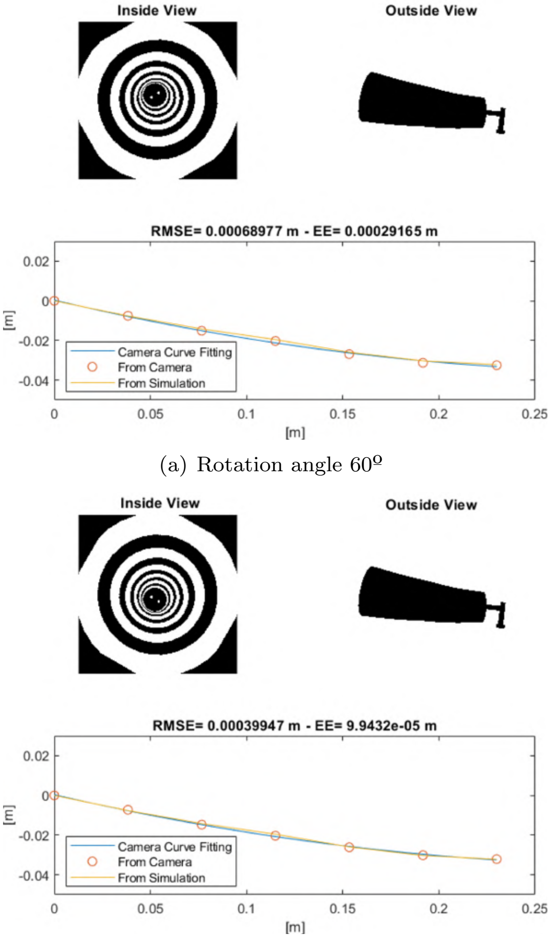
Figure 10: NN Performance - Bending in One Direction

5.3. Bending in Multiple Directions

To evaluate if the NN can predict the bending in various directions, different rotations are applied to the collection of images. The goal of this stage will be to verify if the NN can estimate the deflection in the main bending direction, when it has to combine the displacements in the orthonormal axes. The estimated deflection will only be defined in the main direction.

The inputs chosen for the NN were the marker's Area, Centroid Position (vertical and horizontal), Position along length and Angle of camera. The training process and determination of the best network created follows the same method described for the previous NN.

The evaluation of the results occurs with the comparison of the curves for the same image with different rotations applied. An example of this process is presented below, in figure 11.



(a) Rotation angle 60º

(b) Rotation angle 150º

Figure 11: NN Performance - Bending in Different Directions

In table 4, the performance for each of the methods can be analysed. The computation times are, all at the same order of magnitude, 10^{-5}s, so, it is fair to choose the best method based on the accuracy alone, using the average error values.

Method	*RMSE* (*m*)	*EE* (*m*)
Bound	0.0013	7.5314×10^{-4}
BlobComp	0.0017	7.4768×10^{-4}
BlobBin	0.0015	8.8917×10^{-4}
BlobAll	0.0019	8.7730×10^{-4}
NN (1D)	6.6082×10^{-5}	1.1047×10^{-4}
NN (MD)	2.8817×10^{-4}	4.0691×10^{-4}

Table 4: Performance - Bending Curve

The bending algorithm does not achieve an adequate performance when compared to the NN approach. That way, all further simulations will be assessed using NN. The following step is to test with two cameras, using the structure of the Leap Motion Controller™ sensor.

5.4. Stereo Vision

A new NN was created and trained using six inputs: Area and position of the centroid (vertical and horizontal) for both images. Since the NN takes the data from both pictures, even when there is occlusion of markers from one view, the NN is capable of getting information from the other.

The performance of the NN application in the stereo images can be seen in figure 12.

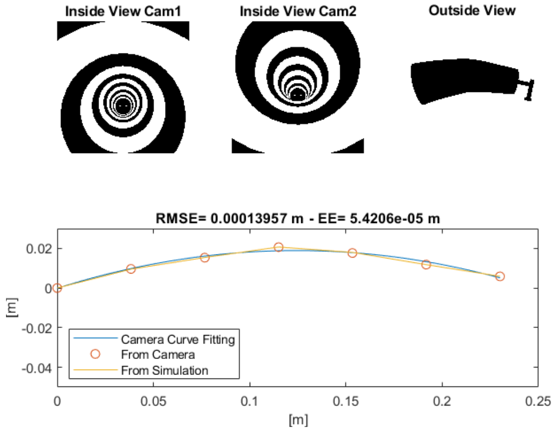


Figure 12: Stereo Set - NN Performance

Below, in table 5, the average performance of the NN is exhibited.

Method	*RMSE* (*m*)	*EE* (*m*)
NN Stereo	3.4895×10^{-4}	4.4119×10^{-4}

Table 5: Stereo Set Performance - Bending Curve

At this point, the method developed is adequately examined and presented a successful performance.

The next step is to validate it in an experimental setting.

5.5. Contact Problem

If there is no contact, all of the boundaries have a circular shape, approximately, which means that the ratio of moments of inertia will be close to 1. If the contact exists, then the circular shape is deformed, which will lead to an alteration of the ratio of inertia moments.

According to this notion, the algorithm presented in algorithm 2 was developed.

Algorithm 2: Contact Estimation Algorithm

```
Input: Position of Centroids ($C_x$, $C_y$),
       Principal Inertia Moments ($I_1$, $I_2$)
       and respective direction ($\theta_1$,$\theta_2$)
Output: Contact point ($CPoint$) and
        Contact direction ($CDirection$)
1. Compute ratio of moments of inertia:
   $Ratio = I_2/I_1$
2. Find peak value and respective index of
   boundary.
3. Compute Contact point:
   $CPoint = max_{index} \times width_{marker}$
4. Get the directions of the minor axis ($\theta_2$)
   and the centroid trajectory($\theta_c$) for the
   boundary:
   $CDirection = \theta_2$
   $\theta_c = arctan(C_{y_i} - C_{y_1}, C_{x_i} - C_{x_1})$
5. Correct contact direction with a $\pi$
   deviation - only if centroid trajectory
   direction is in the same quadrant:
   $CDirection = CDirection + \pi$
```

The direction of the contact is retrieved from the minor axis angle (θ_2) and a decision mechanism was applied after this, using the neutral position of the centroids of the link (C_1). The angle of contact is opposite to the direction the centroid moves (θ_c). This effect is shown in figure 13.

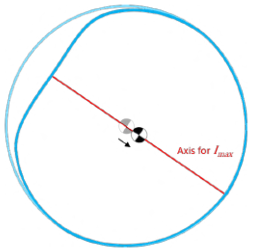
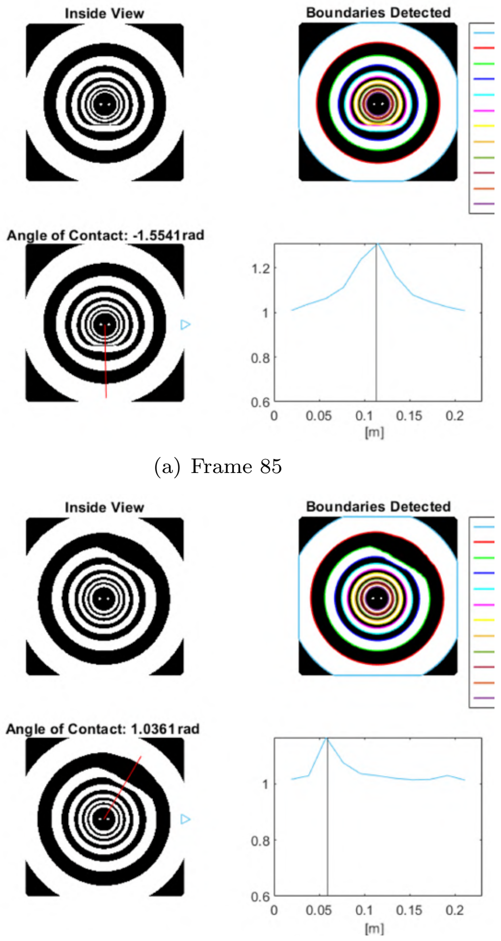


Figure 13: Centroid movement along principal axis

Figure 14 portrays the different elements involved in the contact detection. The contact position received from the simulation is plotted as a vertical line for a straightforward comparison with the peak value.



(a) Frame 85

(b) Frame 163

Figure 14: Contact Evaluation

In table 6, the performance of the developed algorithm for contact detection is exhibited.

Method	*Time (s)*	*Image Display*
Contact Detection	0.22144	Yes
Contact Detection	2.756×10^{-5}	No

Table 6: Table caption

This algorithm achieves a location error of 1.69 *cm*, which places the contact in the right marker, considering their width is about 1.9 *cm*.

For the case that an occlusion of the markers exists, the comparison is not as straightforward for the occlusion might alter the shape of the markers beyond it. A more appropriate definition could be to specify a limit value for each boundary above which the ratio should be disregarded, as it corresponds to an incomplete marker.

Like for the bending problem, the final step is to apply the same type of method to the stereo vision case.

5.6. Stereo Vision

The stereo images do not capture the first marker completely, so it is disregarded, as it is not applicable for the designed method. Therefore, any contact happening in the first marker or before it will not be recognised. This problem has already been solved in the new prototype, with the exponential distribution pattern, that guarantees the first marker is fully captured by the camera.

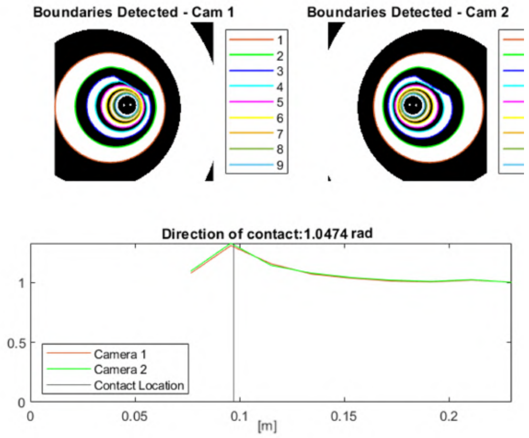


Figure 15: Contact Evaluation - Stereo Vision

This method would be a good candidate for a NN approach, as it would establish rules for the occlusion of markers.

The developed process should not be regarded as a fully finalized method but as the first step of problem-solving and evidence that the second moment of area of the markers provides relevant information for contact characterization.

6. Conclusions

The animation software Blender was a suitable choice, allowing to retrieve realistic enough images of the movement of the link as well as the centerline and contact point positions. This was confirmed by comparing the simulation generated images with images obtained from the experimental prototype in preliminary trials.

Boundary analysis is very advantageous compared to Blob analysis, because of its speed and non-susceptibility to ring distortion, allowing for better results when tracking in real time.

For the bending simulation, the markers of the chosen pattern are sufficient to get a smooth and accurate bending curve. The process of training a NN is worthwhile since the deflection it measures is much more reliable than the empirical algorithm results. The NN approach is also promising when evaluating bending in multiple directions, although these results should be confirmed through further simulation work.

The stereo camera set does not hinder the bending simulation, with the new NN estimating the deflection curve taking information from both images retrieved. The stereo set also offers an advantage when dealing with occlusion of markers, when the deformation occurs in the plane of the two cameras, perpendicular to the image plane, with one of the images probably being occlusion-free, or, at least, less affected by it.

The ratio of inertia moments is a relevant information for contact estimation, but might not be enough to confidently establish the contact point, the average error being $1.69cm$. The results may be enhanced by defining more rules, to avoid choosing an occluded marker or using another approach to compute the contact location, for example NN. As for the direction of the contact, the method chosen returns accurate results, as long as the resting pose is the same and the inflatable link remains with the same internal pressure.

As a final conclusion, it was demonstrated that the designed pattern can in fact provide relevant information for both bending and contact situations, and that a approach merging computer vision and neural networks is a suitable solution for tracking the internal state of the inflatable link.

References

[1] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in Biotechnology*, 31(5):287–294, 2013.

[2] António Franco. Estimation of contact force on an inflatable structure using vision sensors. Master's thesis, Instituto Superior Técnico, Lisbon, 11 2019.

[3] Siddharth Sanan. *Soft Inflatable Robots for Safe Physical Human Interaction*. PhD thesis, Robotics Institute Carnegie Mellon University, Pittsburg, Pensilvania, 8 2013.

[4] Preston Ohta, Luis Valle, Jonathan King, Kevin Low, Jaehyun Yi, Christopher G. Atkeson, and Yong Lae Park. Design of a lightweight soft robotic arm using pneumatic artificial muscles and inflatable sleeves. *Soft Robotics*, 5, 2018.

[5] Siddharth Sanan, Michael H. Ornstein, and Christopher G. Atkeson. Physical human interaction for an inflatable manipulator. In *2011*

Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 7401–7404, 2011.

[6] Agostino Stilli, Helge A. Wurdemann, and Kaspar Althoefer. A novel concept for safe, stiffness-controllable robot links. *Soft Robotics*, 4, 3 2017.

[7] Javier Tapia, Espen Knoop, Mojmir Mutný, Miguel A. Otaduy, and Moritz Bächer. Makesense: Automated sensor design for proprioceptive soft robots. *Soft Robotics*, 7, 2020.

[8] Benjamin Winstone, Gareth Griffiths, Tony Pipe, Chris Melhuish, and Jonathon Rossiter. Tactip - tactile fingertip device, texture analysis through optical tracking of skin features. In Nathan F. Lepora, Anna Mura, Holger G. Krapp, Paul F. M. J. Verschure, and Tony J. Prescott, editors, *Biomimetic and Biohybrid Systems*, pages 323–334. Springer Berlin Heidelberg, 2013.

[9] Benjamin Ward-Cherrier, Nicholas Pestell, Luke Cramphorn, Benjamin Winstone, Maria Elena Giannaccini, Jonathan Rossiter, and Nathan F. Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft Robotics*, 5, 2018.

[10] Benjamin Winstone, Chris Melhuish, Tony Pipe, Mark Callaway, and Sanja Dogramadzi. Toward bio-inspired tactile sensing capsule endoscopy for detection of submucosal tumors. *IEEE Sensors Journal*, 17, 2017.

[11] Hongbo Wang, Massimo Totaro, and Lucia Beccai. Toward perceptive soft robots: Progress and challenges. *Advanced Science*, 5, 2018.

[12] Peter Werner, Matthias Hofer, Carmelo Sferrazza, and Raffaello D'Andrea. Vision-based proprioceptive sensing: Tip position estimation for a soft inflatable bellow actuator. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8889–8896, 2020.

[13] R Wang, S Wang, S Du, E Xiao, W Yuan, and C Feng. Real-time soft body 3d proprioception via deep vision-based sensing. *IEEE Robotics and Automation Letters*, 5:3382–3389, 4 2020.

[14] João Oliveira, Afonso Ferreira, and João C.P. Reis. Design and experiments on an inflatable link robot with a built-in vision sensor. *Mechatronics*, 65, 2020.

[15] Matthias Hofer, Carmelo Sferrazza, and Raffaello D'Andrea. A vision-based sensing approach for a spherical soft robotic arm. *Frontiers in Robotics and AI*, 8:8, 2021.

[16] P. Rajashekar Reddy, V. Amarnadh, and Mekala Bhaskar. Evaluation of stopping criterion in contour tracing algorithms. *International Journal of Computer Science and Information Technologies*, 3, 2012.