Contact Position Estimation in the Event of Simultaneous Multiple Contacts in Vision-based Tactile Sensors

Senarath W. A. T. N, Fernando S. A. W
Department of Mechanical Engineering, Sri Lanka Institute of Information technology
New Kandy Road, Malabe, 10115, Sri Lanka

Rajakaruna R. M. T.P
Department of Mechanical Engineering, Sri Lanka Institute of Information Technology
New Kandy Road, Malabe, 10115, Sri Lanka
thilini.ra@sliit.lk

ABSTRACT

Tactile sensors are used to detect physical contact or pressure. They provide feedback about the physical environment and allow more natural and intuitive interaction with machines. Tactile sensors have many applications in the fields of agriculture, space exploration, health and automotive. Capacitive, resistive, as well as vision (optical) based tactile sensors have been proposed in the literature. This paper proposes a novel approach to solving the problem of estimating the contact locations in the event of simultaneous multiple contacts in vision-based tactile sensors. The relationship between the contact force and the resulting physical deformation of the sensor material of a large-scale tactile sensor was studied with the aid of a custom-built hardware unit. Hardware architecture consists of a custom-designed flat rectangular sensor surface coupled with a mono-vision camera to capture the surface deformation. This method can capture detailed information on the resulting deformation for multiple simultaneous contacts. A software -based deformation estimation algorithm is proposed, where the grid array of marker positions was estimated with a tracking algorithm, an estimation algorithm, and a graphical representation algorithm. Moreover, separate analyses have been carried out to find the best suitable method to observe the deformation of the sensor material. In this study, the approach that was taken to find the contact position and deformation, produced results with an accuracy of more than 97%. Consequently, these results show that this method outperforms existing state-of-the-art techniques in terms of accuracy in the detection of the contact position.

KEYWORDS: *Vision-based tactile sensors, Surface deformation, marker-based localization, Contact point estimation*

1 INTRODUCTION

Human body contains many organs that help humans work, learn and adapt to given environments and applications. Skin is one of the most extensive and complicated organs in the human body responsible for sensations like touch, temperature, and pressure. These sensations help humans to work smoothly and flawlessly. The application of tactile sensors in the fields of agriculture, space exploration and automotive are the use of robotic grippers for plucking harvest, texture, and material hardness monitoring of soil via remote space rovers and touch controls in human machine interfaces.(Wan et al., 2017). In the health care field of application, tactile sensors are mainly used for breath monitoring, wound healing, and pulse monitoring applications. Moreover, tactile sensors are used for voice monitoring and knee motion monitoring applications as well (Liu et al., 2020). Tactile sensors provide additional information during a robot's movements, like gripping and touching. This information is helpful because objects have different physical properties for surface hardness and surface textures. Therefore, they need to be handled at different pressure levels. Therefore, the tactile sensor can help provide that additional information. Figure 1 shows the difference between human tactile and artificial tactile sensations.

ISSN 2961 - 5410 (Online) | https://doi.org/10.54389/HHZM8357

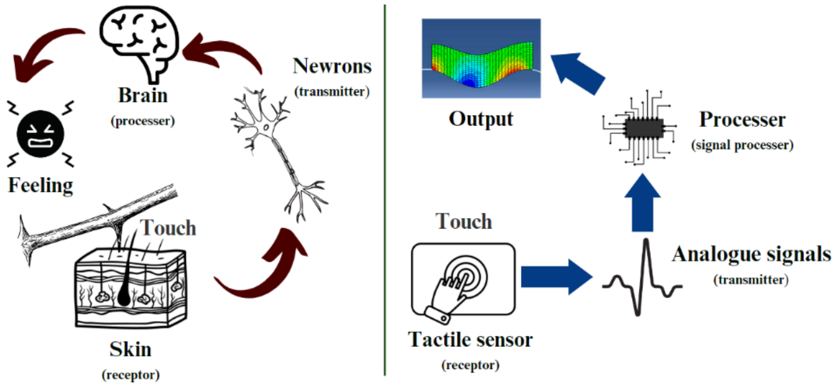


Figure 1: Human tactile sensation and artificial tactile sensation

Tactile sensors can detect deformation when they come into contact with an object, and a suitable technique is used to convert the resulting physical deformation of the contact to an electrical signal. Then these data or the signals are fed into a computer or a microcontroller to carry out the preprocessing and extraction of data. Thereafter, data features like resistance, capacitance, or pixel coordinates, are subjected to a mathematical model to obtain the necessary contact information like coordinates of the contact and the depth of the contact. Currently, vision-based tactile sensors can be seen in industry. Nevertheless, the availability of commercial vision-based tactile sensors is low due to the high cost.

There are mainly three types of tactile sensors based on the working principle: capacitive, resistive, or vision (optical) based tactile sensors. The basic principle behind the capacitive sensors is that the distance between the two plates decreases when there is a force on the sensor. As a result, capacitance changes, and the capacitance level is calibrated to represent a given pressure value. In resistive type tactile sensors, instead of dielectric material, there is a resistive material. When there is a force on the sensor distance between the two terminals decreases, the resistivity of the sensor changes accordingly. However, these sensors are difficult to implement on a large scale in complex geometries. Moreover, capacitive and resistive type sensor surfaces are vulnerable to damage; this is mainly because they come into direct contact with the surface; in vision-based tactile sensors, the electrical components are at a significant distance from the surface, as a result when an excessive force is applied, it is more prone to damage than vision-based tactile sensors. As a result, vision-based tactile sensors can be used in such scenarios.

Vision-based tactile sensor systems can be developed by tracking the movement of maker points. When a force is applied to the soft material of the sensor surface, deformation of the soft material can be seen. Then, using vision-based algorithms, the deformation is estimated with the aid of close-range high-resolution camera capture. Each method has its advantages and disadvantages. The light conductive membrane method can measure contact position and force on soft material; the reflective membrane method can measure the surface texture of the contact object and position, and the marker displacement method can be used to measure contact position shear and torque on the soft material (Van Duong & Ho, 2021a). In this study, a low-cost large-scale vision-based tactile sensor is to be developed to express and enhance the true versatility and potential of the tactile sensation. Thus, this study on vision-based large-scale tactile sensors is mainly focused on the marker displacement approach. The main reason behind this approach is that surface geometry can be changed as required and also depending on material availability. Therefore, at the end of this study, a low-cost large-scale tactile sensor can be introduced to improve the applications of tactile sensors and the sustainability of engineering environments.

In brief, this paper will explore the specific area of engineering related to vision-based tactile sensors, with a focus on a novel approach to solve the problem of estimating the contact locations in the event of simultaneous multiple contacts. The paper will also provide a detailed analysis of the current state of research in this field, as well as new perceptions and impacts. Subsequently, results obtained from the research will be analyzed and discussed. Finally, by examining these important matters, this research aims to make a significant contribution to the field of engineering and provide valuable insights for researchers and engineers to further research on how tactile sensors significantly impact the development of robotics and other applications.

2 LITERATURE

Recent studies have explored various approaches to develop vision-based tactile sensors. Depending on the way deformation is observed and the purpose of the tactile sensor, mainly three types of approaches can be observed in all the past approaches. One method is the reflective membrane approach, which was used by Elliott Donlon et al. in their compact version of the GelSight sensor. By using a semi-specular gel coating on the sensor surface, they obtained the surface texture of the grasping surface as output (Donlon et al., 2018). Another approach, taken by Xinghao Zhu et al., involves using a deformable elastomer as the sensor surface and a Generative Neural Network (GNN) model to reconstruct the deformation of the contact. The ground truth data was generated in a simulation environment using the Finite Element Method (FEM) (Zhu et al., 2022). Gomes et al. used the light conductive method and a 2D convolution neural network (CNN) to obtain the heat map of the deformation, which enabled them to find the contact position on the sensor surface (Gomes et al., 2020). Duong Van Lac et al. used the Inverse Finite Element Method (IFEM) to estimate both the deformation and force on the sensor surface, with the marker tracking method used to observe the deformation (Van Duong & Ho, 2021b). Vijay Kakani et al. also used the marker tracking approach to observe the deformation of the sensor surface, but their main focus was to estimate the contact force (Kakani et al., 2021). According to the above recent studies, many approaches have been taken to observe the deformation and estimate the deformation.

3 METHODS

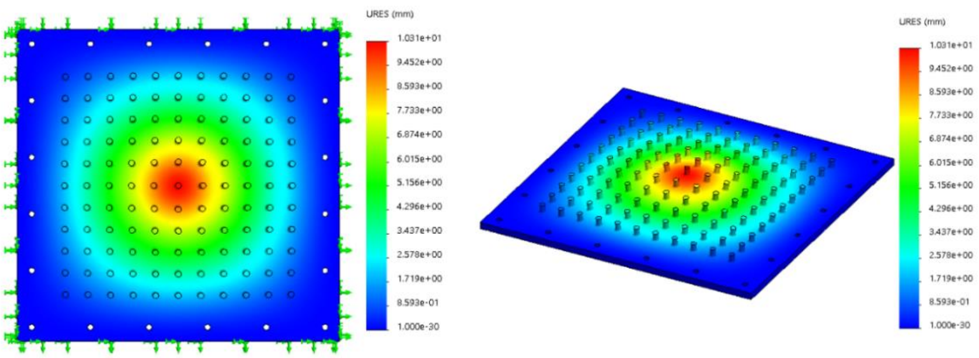
3.1 Sensor surface design

The design of the sensor surface is one of the critical steps of this study, mainly because the material properties and dimensions of the sensor surface will directly affect the robustness and sensitivity of the sensor surface. Therefore, throughout the sensor surface design process, some design selections and analyses were carried out to ensure the sensitivity, durability, and effectiveness of the sensor surface for this study. Some of the main steps that were taken during this phase were technique selection, design selection, and dimension selection.

In the technique selection step, a suitable method was selected to observe the deformation of the sensor surface. During this design section, factors like the effectiveness of the method and the complexity of the design were mainly considered. The light conductive method is an effective method to observe the contact position, but it is a slightly complex approach when compared with the refractive membrane method. The refractive membrane method is more effective when compared with the marker displacement method, but the complexity is greater than the marker displacement method. Although the effectiveness of the marker displacement method is less than the reflective membrane method, it can be further improved by increasing the density of marker points and emerging the marker points away from the sensor surface. Thus, the marker displacement method is a useful method to observe the deformation of the sensor surface effectively.

In the design selection step, several physical factors related to the sensor surface and fabricating methods were discussed. The main aim of this selection is to find the best suitable material, shape, dimensions, and manufacturing process for the sensor surface. The material will determine the robustness and sensitivity of the sensor surface. The manufacturing process will determine the surface finish and the uniformity of the sensor surface. Thus, it is necessary to consider both the material and manufacturing process of the sensor surface at once. When considering those factors, 3D printing using Thermoplastic Polyurethanes (TPU) is the most suitable option for manufacturing the sensor surface; a flexible material that can be used as a filament in 3D printing. The shape of the sensor surface will determine its practicality of the sensor surface. The convex surface can be used in grippers, and cylindrical surfaces can be used in robotic arms. In this study, a flat rectangular surface has been selected to demonstrate the working principle of the sensor surface.

The physical design of the sensor surface mainly depends on specific design parameters. These parameters will have a direct relationship with the accuracy and sensitivity and the manufacturing complexity of the design. The main design parameters are the number of marker points on the sensor surface, the marker point radius, and the thickness of the sensor surface. Since marker points can describe the deformation of the sensor surface, increasing marker point density in a unit area will increase the resolution of the observed deformation. This will increase the accuracy of the contact point estimation. However, some limitations are observed when increasing the marker point density. For example, excessively increasing the marker point density will negatively affect the detection of marker points under complex deformations. Therefore, to measure the deformation effectively, it is necessary to determine the most suitable marker point density. The marker point area will also affect the detection of marker points. When the area of the marker points increases, it will have a lesser possibility of disappearing from the camera view. Furthermore, the height of the marker points will improve the sensitivity of the sensor surface, but excessively increasing the height of marker points will cluster the marker points under minor forces. The thickness of the sensor material has a significant impact on the sensitivity of the sensor surface. Thus, the amount of deformation for a given force depends on the thickness of the sensor surface. Figure 2 shows a simulation carried out in the Solidworks 2019 environment to find the most suitable material and the thickness of the material for the sensor surface. (*Downloads | Support | SOLIDWORKS*, n.d.)



(a) (b)

Figure 2: Deformation simulation of the sensor material; (a)- Front view, (b)- Isometric view

Table 1: Simulation and material parameters

	Parameter	**Value / Description**
Material (Rubber)	Elastic Modulus	0.01 N/mm^2
	Poisson's Ratio	0.45
	Mass Density	960 kg/m^3
	Tensile Strength	20 N/mm^2
Simulation	Mesh density	medium
	Mesh parameters	Standard mesh
	Boundaries	Fixed boundaries

According to the above simulation, the maximum deformation experienced on the sensor surface due to a force of 10N on a 1cm^2 surface on the sensor is 1.01cm. The thickness of the sensor surface is 3mm. The maximum angle of deformation of the sensor surface with the horizontal is necessary to determine the parameters such as minimum marker point radius, marker point height, and the minimum distance between the marker points. A mathematical representation can be modeled to develop a relationship between those factors, as shown in Figure 3.

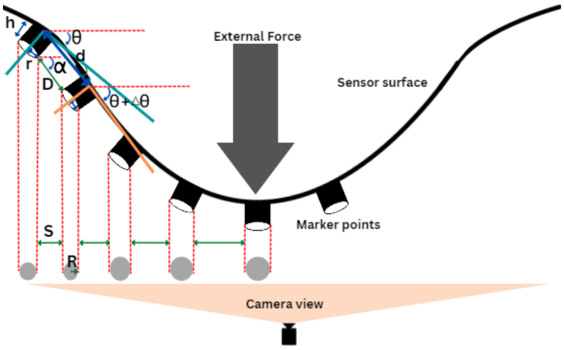


Figure 3: Schematic view of the dimensions of the sensor

$$S = d(\cos(\theta + \Delta\theta)) + hsin\theta - rcos\theta - h\cos(\theta + \Delta\theta) - r\,(\cos(\theta + \Delta\theta)) \qquad (1)$$

In equation (1), the θ stands for the angle made by the base of the marker point with horizontal, r stands for the radius of the marker point, h stands for the height of the marker point, the horizontal distance between two marker points when deformed is shown by the s and d represents the initial distance between the two marker points. Solving the above-modeled equation, solutions to some design parameters like the appropriate radius of the marker point, the height of the marker point number of marker points, and the thickness of the sensor surface. By considering those factors, the following design is modeled using solid works as shown below in Figure 4.

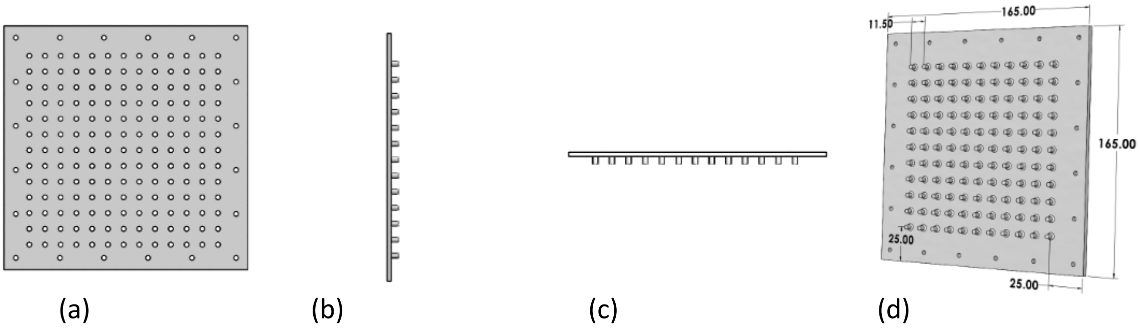


Figure 4: Large scale 3D design of the sensor surface; (a)- Top view of the sensor, (b)- Bottom view, (c)- Side view, (d)- Isometric view with dimensions

3.2 Prototype setup design

The setup consists of components like the outer casing, light source, and camera. Although the main purpose of the casing is to mount and position the sensor surface at the correct distance, it also acts as insulation from the background light. Inconsistency of illumination in a closed environment may increase the noise in the environment. This can be reduced to some extent by covering the reflective

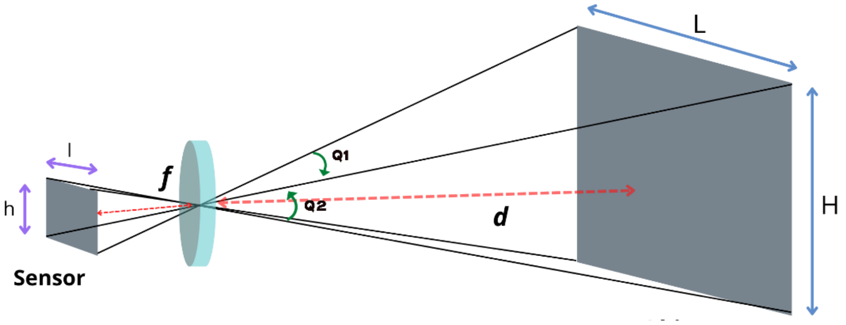


Figure 5: Optical path between object (sensor surface) and camera sensor

surfaces with dark mate material. Consistency and contrast are the two main properties that should be considered when selecting an improved method for insulation. The camera is the main component of the vision system. The supported resolution, frame rate, minimum focus range, and field of view (FOV) are some important parameters that should be considered when designing the vision system. The design of the vision system mainly deals with the estimation of the minimum distance that the sensor surface should be mounted to have a full view of the sensor surface while in the focus range. Sufficient light intensity should be supplied to reduce the noise in the capture frame. Moreover, the light source should be able to disperse the light through the sensor surface uniformly. To derive and model the relationship between the mounting position of the sensor surface from the camera, the optical path of the light rays from the sensor surface should be examined. Figure 5 shows the optical path of the light rays from the sensor surface falling on the sensor surface.

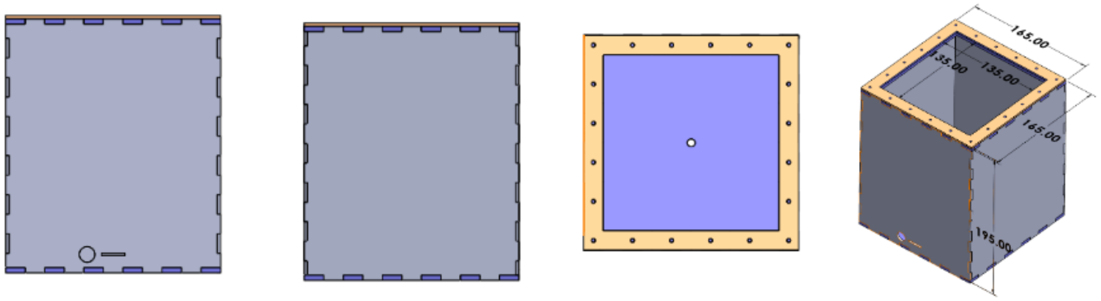
Mainly two factors should be used to determine the minimum distance between the camera and the sensor surface. The camera should be able to view the area of interest, and the distance between the sensor and the surface should be greater than the minimum focusing distance. The required relationship can be modeled considering basic geometric relationships. The equation below shows the relationship between the distance between the lens of the camera with the field of view. In the equations below, the vertical field of view (VFOV) of the camera is represented by *Q2*, while the horizontal field (HFOV) of view is represented by *Q1*. The distance between the camera lens and the sensor surface is generally represented as *d*. During these calculations, the VFOV and the HFOV should be separately counted as two components. Then the distance obtained for *d1* and *d2* will be compared, and the greater value will be selected as *d*. The value obtained for *d* should be compared with the minimum focus range (*F*) distance. To obtain a clear view of the sensor surface, it is necessary to have a *d* value greater than the minimum focus distance of the camera.

$$tan\left(\frac{\mathrm{Q2}}{2}\right) = \frac{(H/2)}{\mathrm{d2}} \tag{2}$$

$$tan\left(\frac{Q1}{2}\right) = \frac{(H/2)}{d1} \tag{3}$$

$$d < F \tag{4}$$

The focus range of the camera is from 10cm to infinity, and the HFOV and VFOV of the camera are obtained directly from camera specifications. According to the above calculations, the distance camera from the sensor surface is obtained as 11.63cm. Since this value is large than the focus range, it can be taken as the minimum distance from the sensor to the camera. Figure 6 shows the finalized design of the casing.



(a) (b) (c) (d)

Figure 6: 3D design of the sensor casing; (a)- Top view of the casing, (b)- Bottom view, (c)- Side view, (d)- Isometric view with dimensions

3.3 Contact position estimation

Estimation of contact position is carried out using the object tracking algorithm. The tracking algorithm observes the movement of marker points in each consecutive frame. The tracking algorithm is based on pixel-wise calculations. Hence, the distortion caused by the camera should be corrected before applying the image processing operations. Radial distortion and tangential distortion are the main distortion types that may happen due to the fisheye effect of the camera. This can be corrected using the camera calibration algorithm. During camera calibration, the chessboard pattern is viewed from different angles, and several frames are saved. Then using these sample photos, calibration parameters like $k1, k2, p1, p2,$ *and* $k3$ and the new camera matrix were obtained. Eqs. (5) - (6) shows a relationship modeled for radial distortion and Eqs. (7) - (8) shows a mathematical relationship modeled for tangential distortion. The distance between the distortion center and the distorted image point is represented by r. The focal length is represented by f_x,f_y, and the optical center is expressed as c_x and c_y. This calibration was performed using OpenCV library functions.

$$\mathrm{X}_{\mathrm{Radial\ distorted}} = \mathrm{x}(1 + \mathrm{k}_1\mathrm{r}^2 + \mathrm{k}_2\mathrm{r}^4 + \mathrm{k}_3\mathrm{r}^6) \qquad (5)$$

$$\mathrm{Y}_{\mathrm{Radial\ distorted}} = \mathrm{y}(1 + \mathrm{k}_1\mathrm{r}^2 + \mathrm{k}_2\mathrm{r}^4 + \mathrm{k}_3\mathrm{r}^6) \qquad (6)$$

$$\mathrm{Y}_{\mathrm{Tangential\ distorted}} = \mathrm{x} + (2\mathrm{p}_1xy + \mathrm{p}_2(\mathrm{r}^4 + 2\mathrm{x}^2)) \qquad (7)$$

$$\mathrm{Y}_{\mathrm{Tangential\ distorted}} = \mathrm{y} + (2\mathrm{p}_2xy + \mathrm{p}_1(\mathrm{r}^4 + 2\mathrm{y}^2)) \qquad (8)$$

$$Camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (9)$$

To carry out image tracking algorithms, the frame is treated as an image, and basic image processing techniques are applied. First, the image is cropped and aligned, and the necessary area of interest is selected. This is done to reduce unnecessary pixel computations that are out of the area of interest. This will reduce memory usage and lowers the computational delay. Furthermore, it will be easy to work with one channel and reduce memory usage. This grayscale conversion can be carried out by treating the image as an array. Then the grayscale cropped frames obtained after the above operations were passed through a series of image operations to distinguish marker points from the background. Then a thresholding algorithm is used to separate grey levels as binary black and white. Therefore, white markers will appear as white dots, and the background will appear as black. The exact margin of the threshold level for the separation of each marker from the rest of the background is obtained using the trial-and-error method**.** A certain level of noise will be visible as white patches in the frame. These patches can be treated using morphological operations like open and closing operations. In this scenario, the main objective is to eliminate small white patches. Thus, an opening morphological operation is conducted to remove the unnecessary noise in the image while keeping the area of markers as it is. The strength of the opening operation is dependent on the kernel size, and it has to be selected using a trial-and-error method. This can be implemented by passing the kernel array throughout the image array. The kernel operation will not be applied to the outermost pixel layer due to boundary conditions. This will erode the image size by (n-1) pixels if the kernel is (n*n) kernel. This can be overcome by using padding before the morphological operation.

After these basic image operations, marker points are visible and distinguished from the background. Thus, to detect the marker points, a contour detection algorithm must be used; this will detect the edges of the object where an intensity gradient happens. These objects will then be appended to an array for further reference. Then using, image operations the area of each closed contour is measured, and if the area is greater than a certain value, it will be considered as a marker point. Then the midpoint of each contour is taken to two variables to denote the x and y coordinates in the image plane. These x and y coordinates of each marker point are joined to have a tuple. Then at the end of each frame, these tuples are appended to form a list. Thereafter, using a conditional statement, length of the list is checked at the beginning of each frame to make sure all the marker points are available. Subsequently, the IDs are given to every marker point to examine them individually. This is mainly carried out to track each marker point throughout the frames. This is implemented by using a counter variable, and when each marker point is detected, the count value will increase. Furthermore, the array is updated to append the ID number of the marker with its coordinates. The ID number is based on the pixel coordinates of the marker points. Thus, the ID obtained by marker points is not in a cartesian order. This will make it difficult to track and do further operations because each time the camera is switched on, the same marker point will get a different ID number. Thus, this will mislead the algorithm and give wrong and invalid results. This can be resolved by implementing an algorithm that organizes the coordinates of the marker points according to the X and Y coordinates and gives the same ID each time when the camera is switched on. Then using the above function, the previously saved marker array will be sorted using the abovementioned function. This will append ID numbers to the marker points according to the coordinates. This can be implemented using nested "if conditions" and a comparison function inside a key function. The comparison function can accept two arguments and compare two arguments to return a value. Then to implement the tracking algorithm, the distance moved by the marker points in each successive frame is measured. If the distance moved by a marker point has changed by more than a certain number of pixels in the successive frame, it will have a new ID number. If not it will have the same ID number. The maximum or the optimum distance moved by a pixel between successive frames is measured using the trial-and-error method. To implement the above algorithm, it is necessary to save the coordinates of the markers at the end of the first frame. To do this first, an empty list should be made to save the coordinates of the first frame, and a variable should be defined to count the number of frames. Then using an if condition, the coordinates of the first frame should be copied to the empty list. Then at each other frame, this list can be referred to by calling the list name with the required ID number. Figure 7 shows some sample outputs at each step.

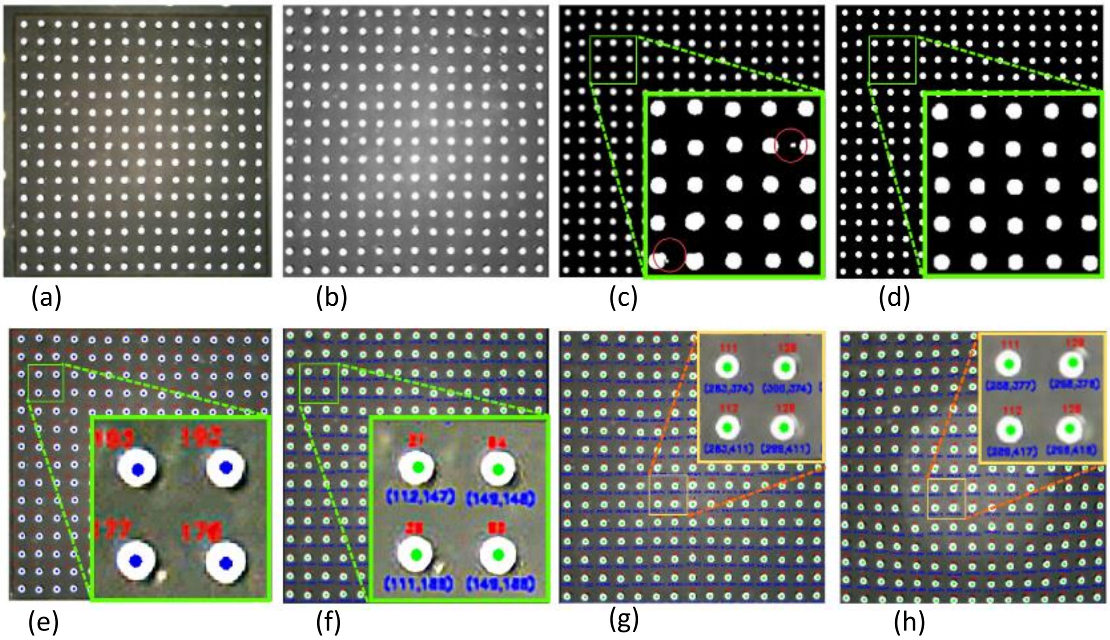


Figure 7: Tracking algorithm implementation; (a)- raw camera image, (b)- gray scale image, (c)- Binary thresholding image, (d)- After morphological operation applied, (e)- Applying ID numbering, (f)- ID sorting applied, (g)- Initial marker IDs, (h)- Marker IDs when deformed.

Depth estimations mainly differ from each other according to the type of vision that has been used; in a stereo vision system, depth can be obtained from the perspective vision. The process of using a stereo camera to determine depth involves using triangulation and stereo-matching techniques. Calibration and rectification are important to ensure that the problem can be modeled on a 2D plane called the epipolar plane (*Depth Estimation: Basics and Intuition | by Daryl Tan | Towards Data Science*, n.d.). Mono camera depth estimation can be performed using two main methods. Estimation of depth using a pre-trained AI model will be an effective method in the open world. This method cannot be used in this scenario due to the poor accuracy of the results. Thus, to measure the depth using a mono camera, a varying physical factor is selected. The intensity of illumination change in the marker points due to deformation can be taken as a physical factor to measure the depth. To accurately measure the depth, it is required to have a constant light source with fixed intensity. This is a challenging process in this prototype design due to the fluctuations in intensity. Consequently, in this case, the surface perimeter is selected as the physical parameter that changes with time.

Depth estimation using object perimeter involves using the dimensions of an object, which is the distance around the object, to estimate the object's depth or distance from the viewer. This method is a highly effective method for estimating the depth using a mono camera. Figure 8 shows the basic principle behind this method. The letter *m* denotes the distance moved by the object, the letter *a* denotes the image height, and the letter *b* denotes the final perimeter of the object.

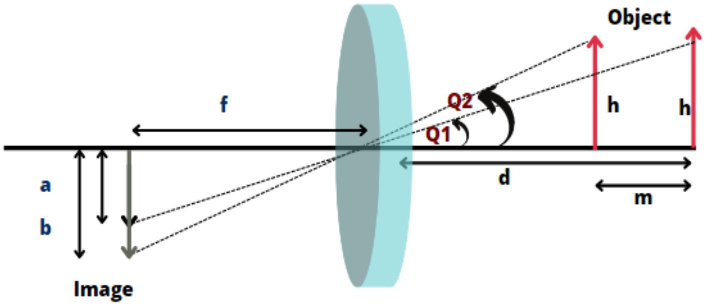


Figure 8: Optical path of image formation on the sensor

Eq (11), shows the modeled relationship between the initial perimeter of the object, the final perimeter of the object, and the distance moved by the object.

$$m = d * (1 - \frac{a}{b}) \tag{11}$$

It is proved that the apparent perimeter of the object increases with the depth and in some scenarios, due to the deformation of the sensor surface. The apparent perimeter of some markers decreases with increasing depth. Thus, this becomes a contradiction to the original concept that was going to be implemented. In order to overcome this issue, instead of taking the perimeter of the markers directly, the perimeter bounded by the corner of four adjacent markers was taken to compare. In this way, the perimeter bounded by the corner markers increases with depth.

.

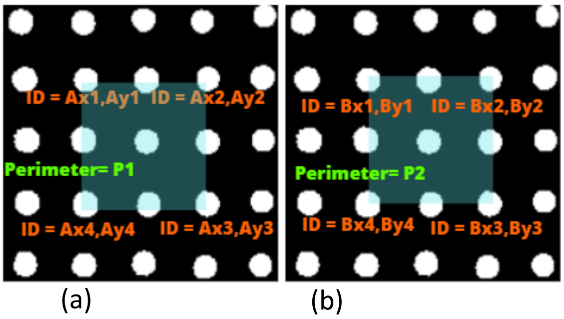


Figure 9: Deformation vs Perimeter; (a)- Perimeter before deformation, (b)- Perimeter after deformation

To carry out this approach, it is necessary to have the initial coordinates of the markers as well as the current coordinates of the markers. When a point is considered, the algorithm will get the ID number of the markers at the four corner coordinates, then it calculates the area of the markers and compares it with the area of the current marker adjacent coordinates. The above algorithm can be mathematically expressed as follows Eq (12) shows the perimeter of the object before the deformation, while Eq (13) shows the perimeter after the deformation.

$$AP\ = (Ay1 - Ay4) + (Ax2 - Ax1) + (Ay2 - Ay1) + (Ax3 - Ax4) \tag{12}$$

$$BP\ = (By1 - By4) + (Bx2 - Bx1) + (By2 - By1) + (Bx3 - Bx4) \tag{13}$$

It can be observed that due to the method that has been implemented to find the depth of the marker points, it is unable to find the *z* coordinates of the marker points at the edge of the image. This can be commonly referred to as erosion of the image, which can be treated using a padding process. In this process, a template that is larger than the frame is merged into the template as shown in Figure 10.

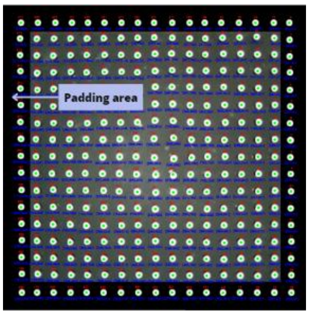


Figure 10: Padded Image

From the above algorithms, it was able to get the 3D coordinates of the marker points. Then these marker points were plotted in a 3D space to visualize the deformation happening on the sensor surface. Inverse distance weighting (IDW) can be used to resolve this matter. To implement the IDW, the first number of iterations and the search radius should be defined. Then an arbitrary coordinate plane should be made according to the number of iteration values. Then using the coordinates in the area defined by the radius, more coordinates are generated using interpolation. Figure 11 given below shows the general interpolation sample data point. Table 2 shows the sample calculation for the IDW method.

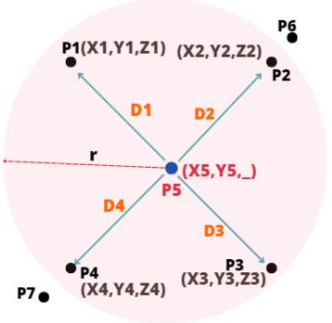


Figure 11: Sample demonstration for point generated from IDW method.

Table 2: Generalized sample calculation

Search radius (r)	Points	Parameter	Distance (d)	1/d	Weight (w)	Value (Z)
r = R	P1	Z1	D1	1/D1	(1/D1) /(1/Dt)	W1*D1
	P2	Z2	D2	1/D2	(1/D4) /(1/Dt)	W1*D2
	P3	Z3	D3	1/D3	(1/D3) /(1/Dt)	W1*D3
	P4	Z4	D4	1/D4	(1/D5) /(1/Dt)	W1*D4
	SUM			∑1/Dt	∑(1/D) /(1/Dt)	∑W*D

4 RESULTS AND DISCUSSION

The proposed contact position algorithm was tested on a Python environment, on a laptop computer with an 8th Gen. Intel® Core™ i7 Processor and 16GB RAM. The camera was able to capture 2560x1440 images with a frame rate of 30 FPs. However, due to the computational lag of the approach, the output of the deformation representation significantly reduced the frame rate to less than one FPS; thus, the 3D mapping algorithm was separately executed at the necessary instance to reduce the overall computational delay. The deformation representation is when a single contact force is applied on the sensor surface; the figures given below show the actual deformation, the camera marker points

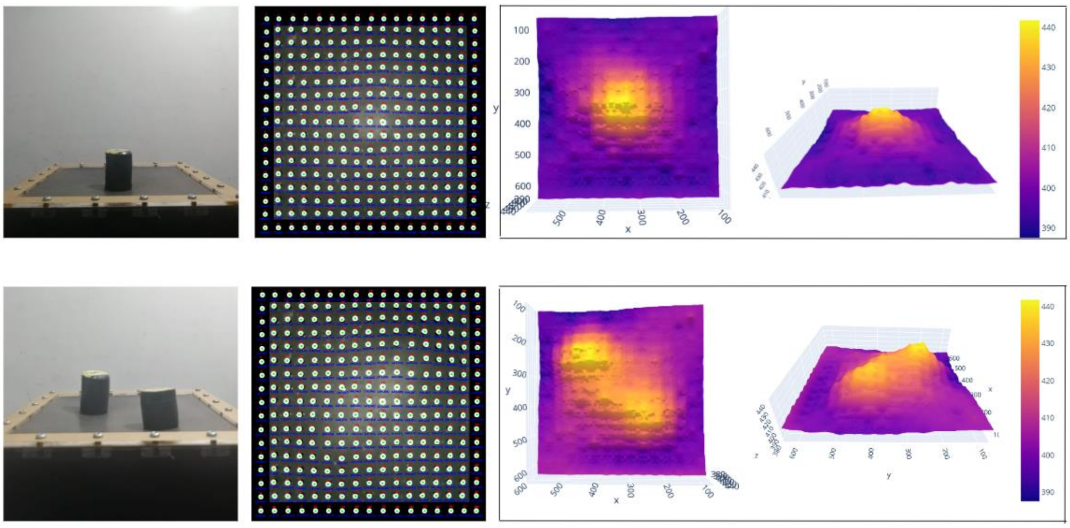
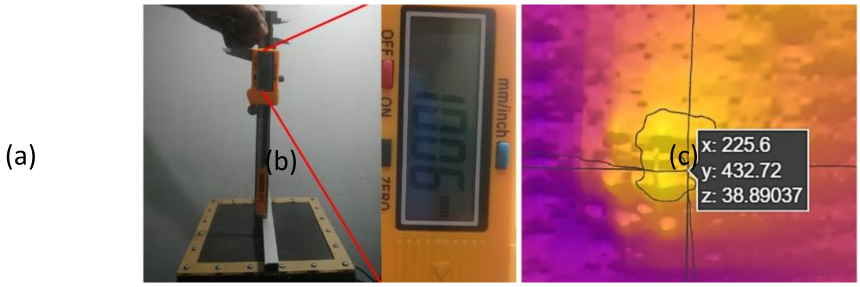


Figure 12: Contact position estimation results; (a)- contact deformation, (b)- Tracking of deformation by algorithm, (c)- graphical representation of the deformation

delocalization, and the virtual graphical representation of the deformation of the sensor surface tracked by the algorithm. When multiple contact forces are applied on the sensor surface, the figures given below (Figure 12) show the actual deformation, the camera marker point delocalization, and the virtual graphical representation deformation of the sensor surface tracked by the algorithm. The deformation given by the algorithm is a 3D representation that can be paned, zoomed and measure deformation at any given point.

Figure 12: Contact position estimation results; (a)- contact deformation, (b)- Tracking of deformation by algorithm, (c)- graphical representation of the deformation

The accuracy of the contact position estimation is measured physically and then compared with the graphical results given by the algorithm. Figure 13 shows the physical approach and the software approach that was used to compare the deformation of the sensor surface. The physical approach shows an effective maximum deformation of the sensor surface as 10.06 mm in the scenario given below, while the simulation deformation shows a 39-pixel value or 10.29mm as the Z coordinate. Thus, this approach was able to yield results with an accuracy of more than 97%. Table 3 shows a comparison and validation of values obtained from the vernier caliper and the developed algorithm.



(a) (b)

Figure 13: Accuracy calculation: (a)-Physical depth estimation, (b)-depth estimation using the graphical representation.

Table 3: Results justification and comparison

	X	Y	Z
Measured value (mm)	60	115	10.06
Scale	1 :1		
Software approach (pix)	225.6	432.72	38.89
Software approach (mm)	59.69	114.5	10.289
Accuracy	99.48%	99.56%	97.06%

5 CONCLUSION

Vision-based tactile sensors are a novel approach to artificial tactile sensation. Many approaches have been taken by researchers over the past several years to present new ideas to overcome some challenges with vision-based tactile sensors. This study was also conducted to address some of the existing challenges. The main aim of this study was to implement and design a vision-based tactile sensor that can localize the contact position of the force. The novelty of this study is to design a method to estimate the contact position and the magnitude of multiple simultaneous contacts.

The contact position estimation was carried out by tracking the deformation of the sensor material when a force is applied to it. To track the sensor material deformation, a set of markers was set underside of the sensor surface. Then using a preprocessing image algorithm, these markers were detected. Then a tracking algorithm is used to track the movement of the markers within consecutive frames. Once tracking was implemented, it gave x and y the coordinates of the marker points. To obtain the depth coordinates or the z coordinate, a separate algorithm was set to estimate the depth using the area of the marker points. Then these data points were plotted in a 3D space. To increase the density of the points, a linear interpolation algorithm was used. Finally, the 3D plot of deformation is obtained.

The tracking algorithm consists of several image preprocessing steps carried out for many calculations per frame. Thus, the frame rate of the video has been reduced by a considerable amount. The 3D plotting algorithm also requires considerable computational power. Therefore, the plotting process has a substantial amount of delay. The plotting algorithm is not embedded in the main code because it slows down the entire code and lowers the frame rate dramatically. When the contact motion on the surface is high due to computational lag and noise in the image, the program may stop and give errors. These issues can be overcome by increasing the raw computational power or optimizing the code to use parrel computation. In conclusion, it can be stated that, this approach was effective.

REFERENCES

Depth Estimation: Basics and Intuition | by Daryl Tan | Towards Data Science. (n.d.). Retrieved December 17, 2022, from https://towardsdatascience.com/depth-estimation-1-basics-and-intuition-86f2c9538cd1

Donlon, E., Dong, S., Liu, M., Li, J., Adelson, E., & Rodriguez, A. (2018). GelSlim: A High-Resolution, Compact, Robust, and Calibrated Tactile-sensing Finger. *IEEE International Conference on Intelligent Robots and Systems*, 1927–1934. https://doi.org/10.48550/arxiv.1803.00628

Downloads | Support | SOLIDWORKS. (n.d.). Retrieved March 17, 2023, from https://www.solidworks.com/sw/support/downloads.htm

Gomes, D. F., Lin, Z., & Luo, S. (2020). Blocks World of Touch: Exploiting the Advantages of All-Around Finger Sensing in Robot Grasping. *Frontiers in Robotics and AI*, *7*, 127. https://doi.org/10.3389/FROBT.2020.541661/BIBTEX

Kakani, V., Cui, X., Ma, M., & Kim, H. (2021). Vision-Based Tactile Sensor Mechanism for the Estimation of Contact Position and Force Distribution Using Deep Learning. *Sensors 2021, Vol. 21, Page 1920*, *21*(5), 1920. https://doi.org/10.3390/S21051920

Liu, Y., Bao, R., Tao, J., Li, J., Dong, M., & Pan, C. (2020). Recent progress in tactile sensors and their applications in intelligent systems. *Science Bulletin*, *65*(1), 70–88. https://doi.org/10.1016/J.SCIB.2019.10.021

Van Duong, L., & Ho, V. A. (2021a). Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. *IEEE Transactions on Robotics*, *37*(2), 390–403. https://doi.org/10.1109/TRO.2020.3031251

Van Duong, L., & Ho, V. A. (2021b). Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. *IEEE Transactions on Robotics*, *37*(2), 390–403. https://doi.org/10.1109/TRO.2020.3031251

Wan, Y., Wang, Y., & Guo, C. F. (2017). Recent progresses on flexible tactile sensors. *Materials Today Physics*, *1*, 61–73. https://doi.org/10.1016/J.MTPHYS.2017.06.002

Zhu, X., Jain, S., Tomizuka, M., & Van Baar, J. (2022). Learning to Synthesize Volumetric Meshes from Vision-based Tactile Imprints. *Proceedings - IEEE International Conference on Robotics and Automation*, 4833–4839. https://doi.org/10.48550/arxiv.2203.15155