A report on capturing continuous and discrete motion using ultrasonic sensors

Lakshay Chauhan, Krishna Somani

[a] *Indraprastha Institute of Information Technology, Delhi, India*

Abstract

We present an innovative endeavour to enhance the capabilities of ultrasonic sensors, utilizing them to capture and interpret object motions within a spatial domain monitored by an array of these sensors. This report presents a novel tool designed to facilitate imaginative interactions, both discrete and continuous, without necessitating prior technical expertise. Our proposed solution presents a distinctive methodology for motion capture, employing a combination of two ultrasonic sensors, an Arduino which acts as an interface between the sensors and the user, with real-time data processing and visualization facilitated via a connected laptop. We have applied machine learning concepts to refine data captured by sensors. We demonstrate the effectiveness of our solution by creating sample applications, including a calculator, piano and drawing tab.

Keywords: ultrasonic sensor, creativity support tool, motion capture

1. Introduction

In today's rapidly evolving technological landscape, there is a growing demand for innovative tools that can bridge the gap between creative expression and technical expertise. One such challenge lies in empowering individuals to craft interactive experiences with little to no coding knowledge. To address this, we propose developing a novel creativity support tool that harnesses user interactions to seamlessly construct captivating interactive experiences with the help of ultrasonic sensors and an Arduino.

Our approach involves integrating ultrasonic distance sensors as a novel mapping tool and using their ability to capture different (both discrete and continuous) interactions in the physical realm. Our goal is to create user-friendly kits based on existing Creativity Support Tool ideas [1, 2]. Our tool empowers users of all backgrounds to create interactive experiences without complex knowledge of microcontroller mechanics or sensors.

2. Key terminologies

- Sampling Rate: Number of distance readings taken by a sensor in a minute.
- Effectual Angle: Maximum angle offset till which senser is able to detect objects.
- Working Range: Range of values that the sensor can take.

3. Related Work

3.1. Literature Survey

3.1.1. On distance sensors kits

- Accuracy comparision of ultrasonic sensors[1]
- Hoverflow: Exploring Around-Device Interaction with IR Distance Sensors[2]
- Empowering Road Vehicles to Learn Parking Situations Based on Optical Sensor Measurements[3]
- Creation of Interactive Dollhouse with Projection Mapping and Measurement of Distance and Pressure Sensors[4]

3.1.2. On Creativity Support Tools

- Creativity support tools: a tutorial overview[5]
- Designing creativity support tool for failure[6]

3.2. Key Takeaways from Related work

- Ultra-sound sensors have some limitations in terms of the sampling rate (about 20 per second) and range (a few meters).
- The LI-DAR-Lite v3 sensor is an off-the-shelf optical sensor that measures the time-of-flight[1] of an emitted laser signal that is reflected by an object.
- Around-Device Interaction (ADI) is based on 3D input, comprising coarse movement-based hand gestures, as well as static position-based gestures. Basically, using sensors, the interaction space of small mobile devices can be extended beyond the physical boundary of mobile devices to include the full 3D space around them. This is particularly beneficial because no matter how small the device may be, the space beyond the device can be used easily.

[1] Time-of-Flight principle (ToF) is a method for measuring the distance between a sensor and an object, based on the time difference between the emission of a signal and its return to the sensor, after being reflected by an object

- Sampling rate of IR sensors is around 25 Hz.

- The difference values (of distances) can be used instead of the absolute values(of distances) in order to make gesture recognition independent of the distance between the user's hands and the device, i.e. the expression becomes, $D = D_t - D_{t-1}$.

4. Limitations of Existing Solutions

- One key competitor of this idea is Capacitive Sensing[7,8]. Some downsides of using that are:
  - It requires touch sensing, hence can only be done on a surface it is connected with.
  - Less modular, and not scalable to larger spaces.

- Another alternative that can serve as a mapping tool is Makey-Makey [9]. But it also poses some limitations:
  - Limited to small-scale projects.
  - Similar to [7,8], it requires touch sensing, limiting it to the connected surface.

- Hardware like leap motion [10] uses an IR camera to enable hand-based gestural interaction in natural ways, but it has a limited field of view and has a steep learning curve - generally not suitable for end-users. Our project proposes an affordable distance sensor-based solution that beginners can also use.

5. Setup

5.1. Setup composition

- An Arduino.
- A laptop with Arduino IDE and python installed.
- 2 Proximity sensors, preferred if they are ultrasonic sensors or any other sensor with similar or better sampling rate/working range.
- A Smooth and empty workspace.
- A cylindrical object such as a bold marker to point with, preferred if it is thick but not too thick as it may affect the readings.
- Few jumper wires.

5.2. Setup layout

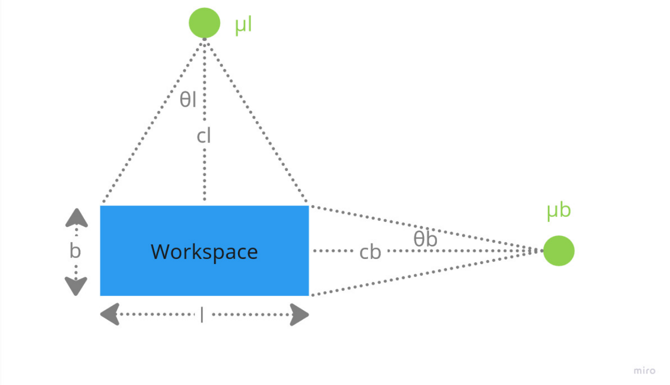


Figure 1: Image depicting the terminologies for distances and measurements

- l: length of workspace (user chosen)
- b: breadth of workspace (user chosen)
- cl: counter to length or perpendicular line from μl to the workspace
- cb: counter to breadth or perpendicular line from μb to the workspace
- μl: Proximity center placed counter to the length
- μb: Proximity center placed counter to the breadth
- θl: angle between cl and a line joining the center of μl to a closest corner (user chosen)
- θb: angle between cb and a line joining the center of μb to a closest corner (user chosen)

5.2.1. Valid range of θl and θb

- $$0 <= \theta l <= effectual\, angle\, of\, \mu l \tag{1}$$
- $$0 <= \theta b <= effectual\, angle\, of\, \mu b \tag{2}$$

5.2.2. Expressions for cl and cb

- $$cl = \frac{l}{2\sin\theta l} \tag{3}$$
- $$cb = \frac{b}{2\sin\theta b} \tag{4}$$

5.3. *Test for setup correctness*

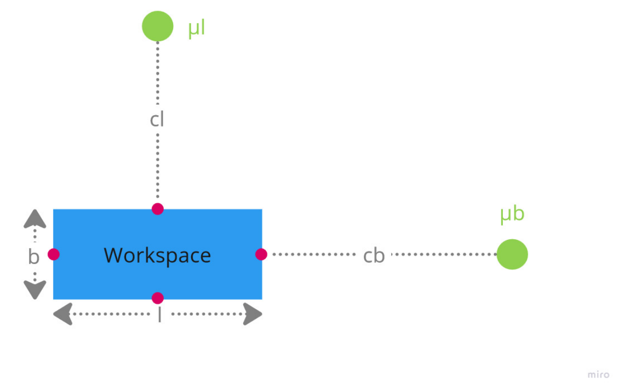


Figure 2: Image depicting four points where we have to test upon

Four point test: To check whether the sensors are positioned and working correctly. Whether the readings coming on above mentioned four points are correct. There are four ranges of readings;

- If the reading is '0'. Then the connections are either wrong or weak.
- If the readings fluctuate or have random noise. Then either the surface is not smooth, or some object is obstructing the field of sensors.
- If the readings are huge and incorrect, then the ultrasonic sensor is broken.
- The reading is correct.

6. Mathematics behind the working

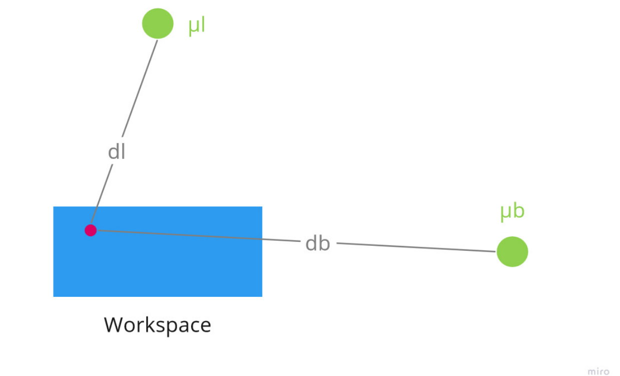


Figure 3: Image depicting the terminology for distance between point and Ultrasonic sensor

- dl: distance between μl and the point
- db: distance between μb and the point

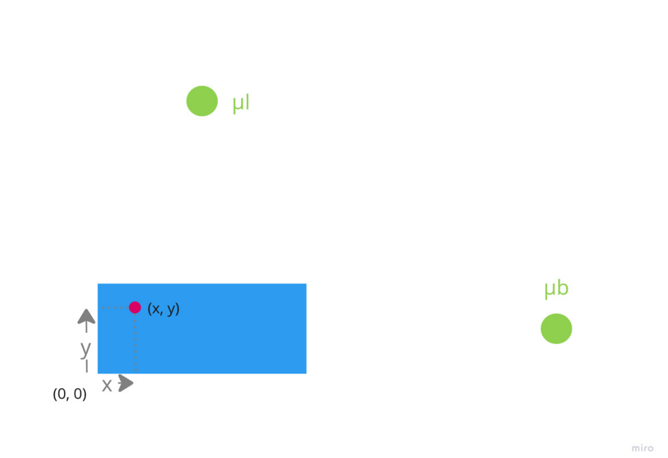


Figure 4: Image depicting mapping of distances to coordinates

- x: coordinate of point projected on the horizontal axis
- y: coordinate of point projected on the vertical axis
- Bottom left corner of the workspace is the origin i.e. (0, 0). And every other point would be relative to origin.

First we take the distances, dl & db from proximity sensors and map it to (x, y) coordinates which we can use to plot on the drawing board.

6.1. *Expressions for dl & db*

- $$dl = \sqrt{\left(\frac{l}{2} - x\right)^2 + (b + cl - y)^2} \quad (5)$$
- $$db = \sqrt{\left(\frac{b}{2} - y\right)^2 + (l + cb - x)^2} \quad (6)$$

All the entities except x & y are known. So, by solving the two equations we can find x & y which are the coordinates of a point being mapped.

- dl & db: given by the Ultrasonic sensors
- l & b: constants (user chosen)
- cl & cb: calculated earlier

6.2. *Valid range of dl & db*

- $$cl <= dl <= \sqrt{(cl + b)^2 + \left(\frac{l}{2}\right)^2} \quad (7)$$
- $$cb <= db <= \sqrt{(cb + l)^2 + \left(\frac{b}{2}\right)^2} \quad (8)$$

As we only want to capture points on workspace. So we will ignore all the distances measurements outside the workspace.

6.3. Expressions for dl & db can be represented as equations of intersecting circles [11]

- $$p^2 = (x-q)^2 + (y-r)^2 \tag{9}$$
- $$u^2 = (x-v)^2 + (y-w)^2 \tag{10}$$
- Relation with the original expression
- $$p = dl \tag{11}$$
- $$q = \frac{l}{2} \tag{12}$$
- $$r = b + cl \tag{13}$$
- $$u = db \tag{14}$$
- $$v = l + cb \tag{15}$$
- $$w = \frac{b}{2} \tag{16}$$
- $$e = \sqrt{(q-v)^2 + (r-w)^2} \tag{17}$$
- $$f = \frac{p^2 - u^2 + e^2}{2e} \tag{18}$$
- $$g = \sqrt{p^2 - f^2} \tag{19}$$

6.4. Expressions for x & y

- $$x = (\frac{f}{e}) * (v-q) \pm (\frac{g}{e}) * (w-r) + q \tag{20}$$
- $$y = (\frac{f}{e}) * (w-r) \mp (\frac{g}{e}) * (v-q) + r \tag{21}$$

"Notice that there are two pairs of valid (x, y). That is because two different circles can intersect at max at different two points. But we want one to one mapping. Thus, we will only take the first pair and ignore the second. It will not affect the final result as we are taking the first pair for all the points that are being mapped"

6.5. Accuracy test for coordinate mapping

dl & db	Actual coordinates	Mapped coordinates
$dl = cl + b$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(50, 0)	(48, 0)
$dl = cl + \frac{b}{2}$ $db = cb + \frac{l}{2}$	(50, 50)	(48, 49)
$dl = cl$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(50, 100)	(48, 99)
$dl = \sqrt{(\frac{l}{2})^2 + (b + cl)^2}$ $db = \sqrt{(cb + l)^2 + (\frac{b}{2})^2}$	(0, 0)	(0, 0)
$dl = \sqrt{(\frac{l}{2})^2 + (\frac{b}{2} + cl)^2}$ $db = cb$	(100, 50)	(98, 49)

Table1: Test with $l = 100$ & $b = 100$

dl & db	Actual coordinates	Mapped coordinates
$dl = cl + b$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(25, 0)	(25, 0)
$dl = cl + \frac{b}{2}$ $db = cb + \frac{l}{2}$	(25, 25)	(24, 24)
$dl = cl$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(25, 50)	(25, 49)
$dl = \sqrt{(\frac{l}{2})^2 + (b + cl)^2}$ $db = \sqrt{(cb + l)^2 + (\frac{b}{2})^2}$	(0, 0)	(0, 0)
$dl = \sqrt{(\frac{l}{2})^2 + (\frac{b}{2} + cl)^2}$ $db = cb$	(50, 25)	(49, 24)

Table2: Test with $l = 50$ & $b = 50$

dl & db	Actual coordinates	Mapped coordinates
$dl = cl + b$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(5, 0)	(3, 0)
$dl = cl + \frac{b}{2}$ $db = cb + \frac{l}{2}$	(5, 5)	(3, 5)
$dl = cl$ $db = \sqrt{(cb + \frac{l}{2})^2 + (\frac{b}{2})^2)}$	(5, 10)	(3, 9)
$dl = \sqrt{(\frac{l}{2})^2 + (b + cl)^2}$ $db = \sqrt{(cb + l)^2 + (\frac{b}{2})^2}$	(0, 0)	(0, 0)
$dl = \sqrt{(\frac{l}{2})^2 + (\frac{b}{2} + cl)^2}$ $db = cb$	(10, 5)	(8, 5)

Table3: Test with $l = 10$ & $b = 10$

6.5.1. Findings

$$accuarcy \propto length * breadth \tag{22}$$

7. Circuit diagram

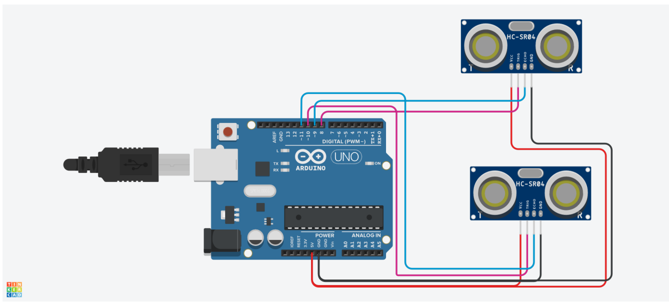


Figure 5: Circuit diagram

8. Limitations

- To correctly capture the distance it requires a thick item to point with that can reflect back rays the most (in case of ultrasonic sensor).

- The pointer object should be perpendicular to the surface while pointing (should not lean towards any direction) otherwise it may reflect back the rays early or late (depending on the leaning direction) and give the wrong distance reading.

- The mathematics behind mapping is accurate but in real practice such precision with capturing distances from sensors is highly unlikely. Thus, to maximize correctness we have to give up on speed. And between any two subsequent readings there are three sets of delays;
  - As ultrasonic sensors have a sampling rate of 20 i.e. we need to wait for atleast 50 ms between subsequent readings. [Required and cannot be removed].
  - Delay between taking readings of both sensors so that they do not interfere with each other's readings. [Required and cannot be removed].
  - Taking an average of a couple of readings to improve accuracy. [Not required but just there to improve the accuracy].

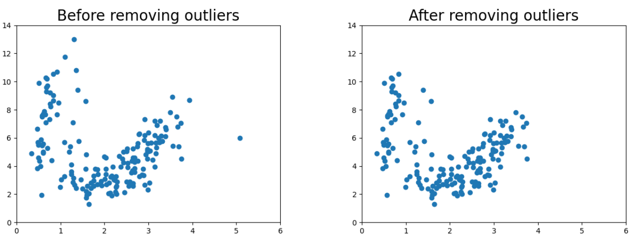
- Drawing a line may result in dotted line because of the above delays. So, to draw you have to move the pointer slowly.

- As the workspace size increases, thus space required between the sensor and the workspace increases almost twice as fast. If the workspace size is 100mm x 100mm, and the output image is 1 to 1 scaled, it would be tiny on the computer screen. So we need to scale it up, but it will make the image uncompressed, resulting in poor image quality. Moreover, we cannot subdivide 100mm into smaller distance units because of the precision limit of the ultrasonic sensor, i.e. 1 mm.

9. Accuracy improvement

- Further tuning of each kind of delay may result in faster and overall better performance.

- Replacing ultrasonic sensor with different proximity sensor which has higher sampling rate (less delays), and higher effectual angle (so the sensor and the workspace are not far apart).



(a) label 1 (b) label 2

Figure 6: Removing outliers using DBSCAN algorithm

- Using ML algorithms like DBSCAN, Local Outlier Factor(LOF in short), Mahalanobis distance, etc to detect outliers and remove them.
  - Removing outliers simultaneously with fetching input and drawing is not possible due to delay in computing and it requires prior data to remove outliers from.
  - So, first fetching all the readings and then by pressing a certain key on the keyboard we can start the algorithm to remove outliers.

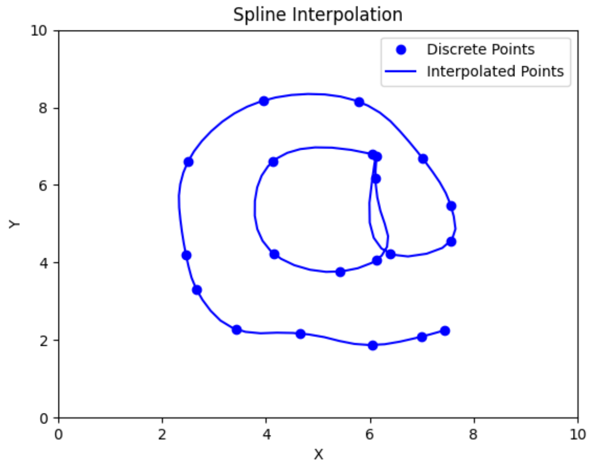


Figure 7: Estimating points using spline interpolation

- Using mathematical interpolation to estimate values between given data points. This is particularly useful in case of continuous motion/gestures. If we receive discrete points from users that need to be in a continuous line of motion we can use interpolation there. Interpolation can be of multiple types: linear interpolation, Spline interpolation, polynomial interpolation etc. For our work of mapping continuous interpolation we can use Spline interpolation. Spline interpolation is more helpful since it works by connecting the data points with a curve that is flexible and smooth. This curve passes through or comes close to the given data points, allowing us to estimate values at locations that were not originally part of the data.

"Interpolation is good, and all theoretically, but in practicality, users might accidentally have their finger obstruct the sensor's field of view unintentionally, and even one wrong reading can change the whole structure, resulting in a bad user experience. Moreover, occasionally, due to accuracy limits sometimes, you get wrong readings which can also cause the issue. Thus, require careful handling and a whole lot of hard thresholds."

10. Our take on this approach

- GitHub repository: https://github.com/Hyouteki/Surf-project

- Blog: https://hyouteki.github.io/me/blogs/capturing-motion-using-ultrasonic-sensors/

11. Making it Customizable

Making our tool user-friendly and customizable was the priority from the first day.

- In our take of this approach; we added a 'parameters.json' file in which user can change the dimensions of the workspace.

- User can choose which ever proximity sensor they like and only have specify that in the json file.
  We prefer sensor to be ultrasonic because of its overall performance.

- User can also change the thetas i.e. θl & θb under a range.

All in all we emphasize customizability and user experience through out the making of this project.

12. Sample Applications

- Sample Applications are implemented to explain the usage of our tool, test the limits of the tool and guide users about how they can utilize the tool well.

  – Calculator (For Discrete Motion)

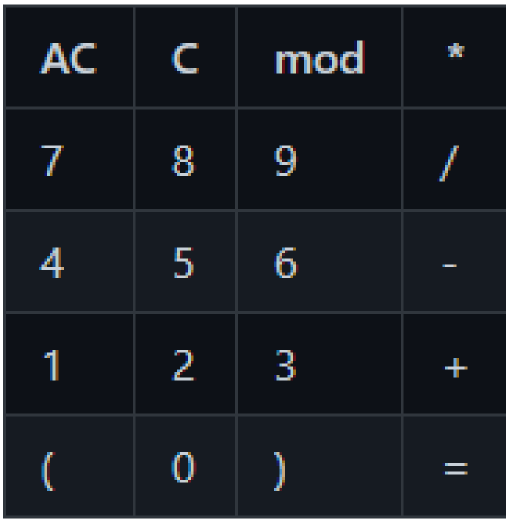


Figure 8: Layout for calculator application

click this for more information on this application

  – Piano (For Discrete Motion)

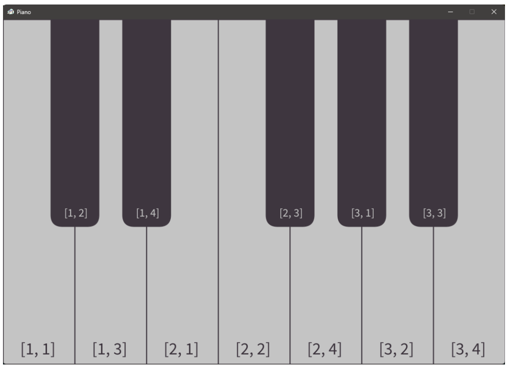


Figure 9: Layout for piano application

click this for more information on this application

  – Drawing Tab (For Continuous Motion)

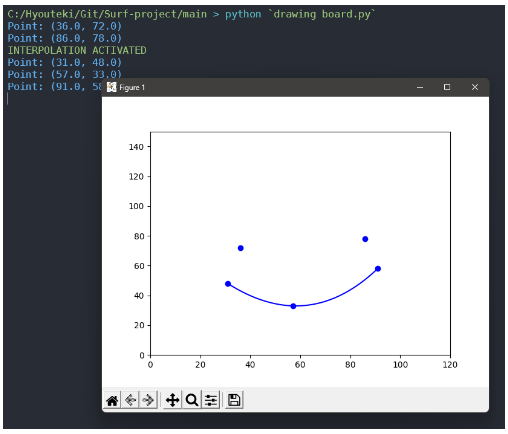


Figure 10: Our attempt to draw a smiley face using this approach

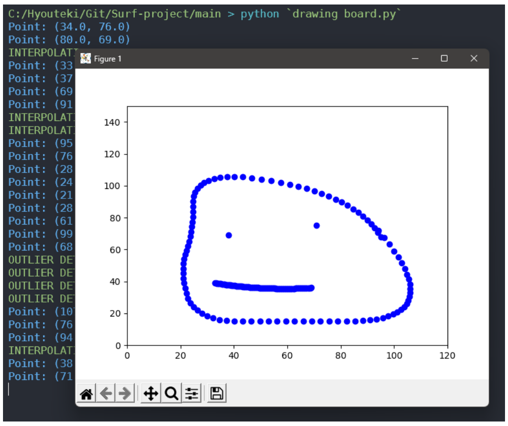


Figure 11: Our attempt to draw a Blob using this approach

click this for more information on this application

13. What we achieved and Room for growth

13.1. Our Achievements

- Our extensive research found no such article or paper exploring the concept of mapping interactions without touching a conductive surface. Thus, we conclude that this concept is a novelty.

- The kit requires no surface. Thus, we can use this kit anywhere: on the bed, on the table, etc. It can also be placed vertically on a wall as a whiteboard.
- Used capabilities of ultrasonic sensors to their fullest to map discrete and continuous motion coordinates.
- Devised an approach that does the mentioned above.
- Applied a combination of different algorithms to make continuous motion smoother and more accurate.
- Discrete motion output is working well with impressive accuracy.

13.2. Room For Growth

- Although we have tried multiple ways and algorithms to improve the accuracy of the result from continuous motion, the result can be improved even further and made more accurate. So, we still can look into other algorithms and devise other ways to improve continuous mapping output.
- Spline interpolation and outlier detection give good results individually but are not very compatible with each other. Spline interpolation requires some spread-out points, but outlier detection will remove those spread-out points as those are outliers. For this, we have taken some creative liberty and switched outlier detection when interpolation is used. So, we can find a better alternative for this.

14. Future Work

- A sturdy and extendable contraption can be made to keep the ultrasonic sensors and Arduino in place.
- It can be extended to a 3-dimensional plane that can track the movement of an object in space. An application of this may include hand movement in virtual reality games or gestures.
- It can be equipped with more advanced machine learning to make continuous motion even more accurate.
- It can also be equipped with different kinds of sensors.

Acknowledgements

Special thanks to Dr. Anmol Srivastava and Dr. Sujay Deb for their mentor-ship. We are really grateful for their guidance throughout. We will love to work under them in future too.

Appendix A. Why use ultrasonic sensor for this?

Comparing different kinds of proximity sensors

- Time of Flight Sensor[VL53L0X]: Smaller working range than Ultrasonic sensor and very expensive
- Ultrasonic Sensor[HCSR04]: Decent working range and very cost-effective compared to other sensors
- Infrared Sensor[LM393]: Extremely small working range; so, not suitable for this task
- TF Mini Li-DAR: Great working range and sampling rate; but very expensive

"Ultrasonic sensor is a good enough combination of decent working range, decent sampling rate and cost-effectiveness. Therefore it is most suitable for this task"

References

[1] Saputra, Adi & Hindarto, Djarot & Haryono, Haryono. (2023). Supervised Learning from Data Mining on Process Data Loggers on Micro-Controllers. Sinkron. 8. 157-165. 10.33395/sinkron.v8i1.11942.

[2] Vijay Kumar and Kolin Paul. 2023. Device Fingerprinting for Cyber-Physical Systems: A Survey. ACM Comput. Surv. 55, 14s, Article 302 (December 2023), 41 pages. https://doi.org/10.1145/3584944

[3] Markus Hiesmair and Karin Anna Hummel. 2017. Empowering road vehicles to learn parking situations based on optical sensor measurements. In Proceedings of the Seventh International Conference on the Internet of Things (IoT '17). Association for Computing Machinery, New York, NY, USA, Article 39, 1–2. https://doi.org/10.1145/3131542.3140277

[4] Makoto Jimbu, Minori Yoshida, Hiro Bizen, and Yasuo Kawai. 2020. Creation of Interactive Dollhouse with Projection Mapping and Measurement of Distance and Pressure Sensors. In SIGGRAPH Asia 2020 Posters (SA '20). Association for Computing Machinery, New York, NY, USA, Article 10, 1–2. https://doi.org/10.1145/3415264.3425461

[5] Ben Shneiderman. 2002. Creativity support tools: a tutorial overview. In Proceedings of the 4th conference on Creativity & cognition (C&C '02). Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/581710.581711

[6] Joy Kim, Avi Bagla, and Michael S. Bernstein. 2015. Designing Creativity Support Tools for Failure. In Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition (C&C '15). Association for Computing Machinery, New York, NY, USA, 157–160. https://doi.org/10.1145/2757226.2764542

[7] HCI Group, Saarland University, "Multi-Touch Kit: Building your own multi-touch device". https://hci.cs.uni-saarland.de/projects/multi-touch-kit/

[8] Bare Conductive. Touch Board. $https://www.bareconductive.com/collections/touch-board$

[9] Makey Makey, "Makey Makey - An Invention Kit for Everyone". https://makeymakey.com/

[10] Gordon Wetzstein and Robert Konrad, "Hand movement and gesture recognition using Leap Motion Controller," Stanford University, EE267 Course Report. $https://stanford.edu/class/ee267/Spring2016/report_lin.pdf$

[11] Joe Elder (https://math.stackexchange.com/users/52597/joe-elder), How can I find the points at which two circles intersect?, URL (version: 2012-12-11): https://math. stackexchange.com/q/256100