

A high resolution grip strength measuring system for rehabilitation of hand conditions

Kevin Smit

*Data-management and biometrics group University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)*

Abstract—Limitation of hand function can have a significant impact on quality of life. The e-Cone is a prototype grip strength measurement system for patient rehabilitation. In a previous research, a 64x64 pressure sensor was developed. It consists of two flexible PCBs with a layer of pressure sensitive Velostat in between. This has been shown to be a possible replacement for the commercial pressure sensors with limited resolution. We have built a new e-Cone prototype with a redesigned sensor. The redesigned sensor has a resolution of 128x64 sensor pixels to increase the sensor area and allow better enclosure of the casing. The readout circuitry has been redesigned to allow for a higher sampling rate. A battery with charging system is included to make the device completely wireless. The user interface has been given a responsive design with increased functionality. The pressure sensor is capable of visualising the grip pattern of a hand. We have explored the possibilities of image processing on both the server and client side. With client-side interpolation, we are able to achieve a frame rate of 15 frames per second at a resolution of 512x256.

Index terms— Velostat, Pressure sensor matrix, Grip force, Analog data acquisition, Image processing

I. INTRODUCTION

Think of how much you rely on a strong grip every day. We use our hands for all kinds of tasks. So hand function is very important to us. Weak grip strength can have a big impact on quality of life [1, 2, 3, 4]. Handgrip strength is a good indicator of future disability problems. As a result of weak grip strength, a task as simple as opening a bottle or pulling a zipper becomes a difficult hurdle. Possible causes of hand function loss include malnutrition, disability, chronic illness, weight loss, stroke, and diabetes [5]. Exercise and therapy can help people improve hand function and grip strength. In most cases, progress can be made. However, the rate at which improvements can be made depends on the cause and the situation. There are many devices that help with exercise. For example, grip strength trainers, finger trainers, and reverse grip trainers. These devices can be very helpful in the rehabilitation process but do not give much feedback to the user.

The e-Cone is a prototype grip strength measurement system for patient rehabilitation. It can help people better understand how they use their hands. As a support for the rehabilitation process. By providing live visual feedback on how they distribute the forces in their hand to grasp an object. Simply hold the e-Cone in your hand like a cup or bottle. It registers how much force you are applying with your hand and fingers. This is then visualised as an image that you can compare to the function of a healthy hand. This way, people can immediately

see any difference and understand how to improve their hand function. A valuable addition to a treatment or rehabilitation process.

In a previous research, a method was presented to replace the need for a commercial pressure sensor by developing a low cost high resolution pressure sensor and acquisition system. It allows us the flexibility to customise the sensor to our specific needs, something that the commercial sensor cannot always do. This thesis aims to continue previous research by improving the high resolution pressure sensor and acquisition system with the goal of integrating it into a next e-Cone prototype. The main research question of this work is: Is the high-resolution pressure sensor suitable for use in the e-Cone? We defined the following sub-questions. (1) What kind of image processing is possible. (2) Can we make the device wireless by integrating a battery system? (3) Determine the suitability of the Raspberry PI Zero W. (4) How does the bending of the sensor affect the usability of the sensor?

The paper is organized as follows. Section II discusses previous work related to this research. Section III gives a brief overview of the goals for this research. Section IV presents the new design of the casing, acquisition system and pressure sensor. Section V describes the verification of the acquisition system using LTSpice. Section VI discusses the software and the studied optimizations. Section VII presents the result of the sensor experiments. Section VIII discusses the results. Possible future work is presented in section IX. Finally, Section X provides a brief summary of the work.

II. BACKGROUND

A. Available training devices

There are a variety of exercise equipment available to help improve hand function. A hand grip trainer to train crushing grip strength. A finger trainer that trains each finger separately and more efficiently with its own resistance. The reverse grip trainer trains the muscles that open your hands to improve the ability of your palms and fingers to reach further. A hand dynamometer is used to measure grip strength. These devices, are just a few examples of training devices. What is generally missing is that none of these devices show you how to distribute the forces in your hand. A hand dynamometer will only show you your grip strength score. Seeing how you distribute the forces in your hand can be significant in understanding how you use your hand and how you could improve that.

B. Original e-Cone

The e-Cone from fig. 1, is a prototype grip strength measurement system for the purpose of patient rehabilitation [6]. It provides a high-resolution, live visualisation of the grip pattern of a hand. The e-Cone consists of a measurement system integrated into the base of a truncated cone, combined with a commercially available sensor developed by Tekscan. The advantage of a cone-shaped carrier is that it can be used for all hand sizes. The Tekscan sensor has a resolution of 44x44 sensor pixels. The acquisition system is built around an FPGA. The sensor readout is converted to an image and sent to a PC via a USB cable for visualisation. A client-side application is required on the client PC. There is no application for smartphones or tablets. On PC, the low-resolution image is interpolated to a high-resolution image that is displayed. The e-Cone is powered by the connected client PC via USB.



Fig. 1: Original e-Cone prototype

C. Sensor principles

The Tekscan sensor, from the original e-Cone, has a resolution of 44x44 sensor pixels. However, the square shaped sensor does not allow us to achieve full coverage of the casing. The long cable and special connector make it difficult to use. The sensor has a high price and complexity, along with low resolution. So it was decided to look for an alternative sensor that would give more flexibility. Other commercial sensors were either too large or too low in resolution. In a previous research, a high resolution pressure sensor was developed. As a low cost alternative for Tekscan's sensor. Papers[7] and [8] describe a method for building a low cost resistive sensor array using a piezoresistive material called Velostat. A row-column architecture is used, where one row and one column can be selected to read a point on the sensor array, see fig. 2. A voltage V_{test} is applied to the selected row and converted to

V_{out} via the selected column and resistor R_k . The unselected rows are set to virtual ground and unselected columns are set open.

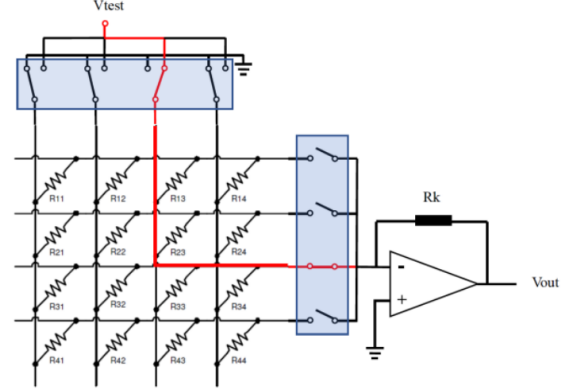


Fig. 2: Principle sensor scanning

The current flowing through the sensor is converted back to a voltage using an op amp based current to voltage converter. The voltage is given by the equation 1. Here R_K is equal to $22K\Omega$ and R_{33} is the resistance of the Velostat.

$$V_{out} = -\frac{R_k}{R_{33}} * V_{test} \quad (1)$$

The resistive sensor array consists of three layers. One layer consisting of vertically arranged electrodes, one layer consisting of horizontally arranged electrodes and in between is a layer of piezoresistive Velostat. The relationship between the variation of the resistance and the applied force over a piezoresistive material is inversely proportional. Thus, the resistance of the material is dependent on the amount of force applied. The sheet of Velostat purchased from Adafruit. It has a surface resistance of 31.000Ω and a volume resistance of $500\Omega \cdot cm$.

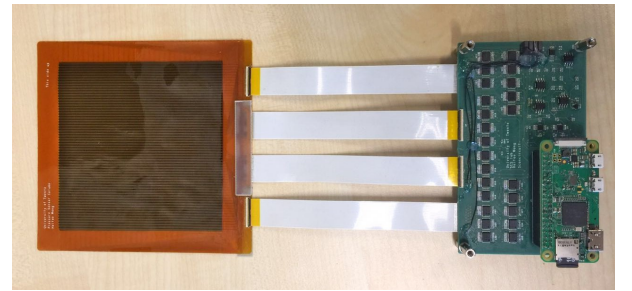


Fig. 3: High resolution sensor setup

D. High resolution pressure sensor

The high-resolution sensor is shown in fig. 3. A Raspberry Pi Zero W is used to read the high-resolution pressure sensor. The sensor has a resolution of 64x64 sensor pixels. This is higher than the 44x44 sensor pixels of the Tekscan sensor. Raspberry Pi's are microprocessors that can run a web server. This allows wireless data transmission to client devices. A user

can interact with the sensor by connecting a smartphone, tablet or PC to the Raspberry Pi Zero W wireless access point. A client device only needs to have an internet browser, making the setup very accessible.

III. REDESIGN E-CONE

In its current form, the high-resolution sensor setup is not yet suitable for use in the e-Cone. To determine the suitability of the high-resolution sensor, a redesign of the e-Cone must be undertaken. At the same time, we want to make the next prototype more accessible and user-friendly. We have created a list of items to be addressed in this research.

1) *Casing*: In the first e-Cone prototype, shown in fig. 1, the electronics are housed in a separate compartment below the cone. The truncated cone shape of the carrier offers little space. This makes it difficult to fit electronics and components inside. Another difficulty is matching the shape of the sensor to that of an unfolded truncated cone. Therefore, the truncated cone shape is changed to a cylinder. For the next prototype, we want a simpler and more compact design without a separate compartment for the electronics.

2) *Pressure sensor*: The sensor from fig. 3 is square shaped and has a resolution of 64x64 sensor pixels. There is a thick border on all four sides of the sensor. Since it is a single layer flexible PCB, all traces must be routed through the bezel. The sensor is connected to the acquisition board via four FFC cables. We want to change the square shape to a rectangle, which is the shape of the unfolded cylinder. So that the sensor covers the casing completely. This will increase the resolution and the number of traces. To get rid of routing traces through the small bezels, we will use a two-layer flexible PCB. We want to reduce the number of FFC cables to simplify the connection and be less fragile.

3) *Acquisition board*: The acquisition board of the high-resolution sensor setup in fig. 3 is too large to fit in the carrier, measuring 120x69 mm. A more compact redesign of the acquisition board is needed to fit into the carrier. Looking at the board fig. 3, you can see that the 24 multiplexers and 4 FFC connectors with 32 contacts each take up a lot of space. Also the cables between the board and the sensor are very delicate. The e-Cone prototype and the high-resolution sensor setup use a wired power connection. Since a wired power supply is not always available or is impractical. We want to make the next e-Cone prototype battery powered. To make the device more user friendly. This includes an integrated charging circuit so that the batteries do not have to be removed from the casing.

4) *Image processing*: With the setup from fig. 3, a frame rate of 15 frames per second with a frame time of 60 ms can be achieved. However, this is only a 64x64 color image. When interpolation is applied, the frame rate reduced to less than 5 frames per second. This is because non-optimized interpolation methods are used. We want to look at the FreeImage and OpenCV libraries. These have more optimized interpolation methods. To explore if the Raspberry Pi Zero W is capable of image processing, what kind and what is the cost in terms of

frame rate and performance. We expect that it will be possible to apply interpolation and maintain the frame rate of 15 frames per second. We will also explore the possibilities of client-side image processing to reduce the frame time.

5) *User interface*: The user interface (UI) from the high resolution sensor setup has limited functionality. The web page consists only of a stream, start and stop button. It does not show which settings are used. There is no possibility to change the settings to adjust the output. We want to redesign the UI so that it gives the user more information about the configuration and control over the e-Cone. We want the UI to be responsive so that it adapts to the screen size of the client device. A responsive design will make the device more user friendly.

IV. DEVELOPMENT OF PRESSURE SENSOR

A. Redesign casing

The new design of the casing can be seen in fig. 4. The main changes from the previous casing in fig. 1 are the shape, which has been changed from a truncated cone to a cylinder, and the removal of the footbox. To make it easier to mount the components inside the casing, it is divided into two half cylinders instead of one full cylinder. The sensor is mounted on the outside of the casing and must be connected to the acquisition board inside the casing. There is a small vertical gap on the side through which the sensor can enter the casing. The gap is in the middle of the casing to allow the sensor to bend gradually so that the pressure sensor is not damaged by sharp bending. The top and bottom covers hold the device together with screws. There is a small recess in the bottom cover to accommodate the wireless receiver. To place the wireless receiver as close as possible to the charging pad. The circumference of the casing is matched to the sensor length to ensure optimal sensor coverage.

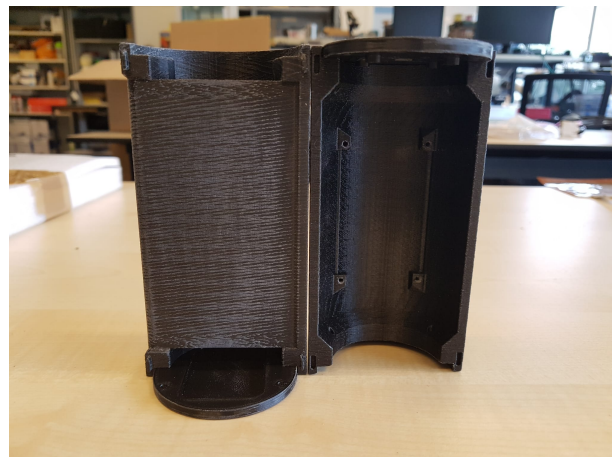


Fig. 4: Redesigned e-Cone casing

B. Pressure sensor

We went with a resolution of 128x64 sensor pixels in order to make the sensor rectangular so that it completely encloses the casing. This gives a total of 8192 sensor pixels. We use

a trace width and spacing of 0.8 mm. This gives the sensor a length of 205 mm, which corresponds to a cylinder diameter of 65.25 mm. A size that should be comfortable to hold in the hand. The sensor layers are flexible 2-layer PCBs with a thickness of 0.13 mm, so they can be bent. The total thickness of the sensor is about 0.36 mm. The bend radius for 2-layer flexible PCBs is 12 times the thickness, which in our case is 1.56 mm. A 2-layer PCB allows us to route all the traces to the rows and columns on the second layer, so we don't need to use the bezels. We remove the vertical bezel to get a full closure around the casing. The horizontal bezels can not be removed. They are needed for gluing, as there can be no glue between the Velostat and the sensor traces.

There is a flap for the components that goes inside the casing so that the components are not exposed to the outside. The component areas are reinforced with a copper pour to increase rigidity and decrease flexibility. The two sensor layers are connected together with a stacking connector. By placing the multiplexers on the sensor, we reduce the number of cables between the sensor and acquisition board from four to one compared to the high-resolution sensor setup. The multiplexers on the column layer are the ADG731, a 32:1 channel analog multiplexer. At the row layer, we use the MAX14661, a 16:2 channel analog multiplexer. With this combination of multiplexers, we only need four on both layers, for a total of eight.

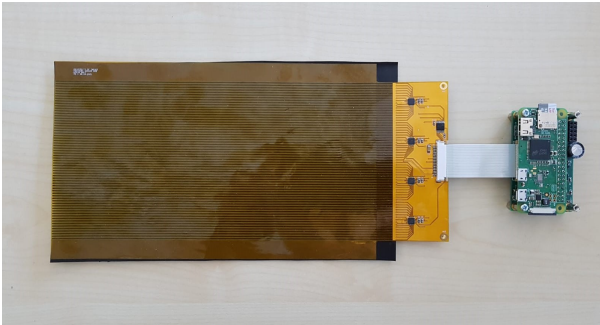


Fig. 5: Pressure sensor and acquisition board

C. Acquisition board

In the redesigned acquisition board, we saved space by moving the multiplexers to the pressure sensor. This allows us to have a larger, less fragile connector on the board. Now the board is not only suitable for one sensor, but can also be used by sensors with a different resolution. This is because only the readout circuitry is on the acquisition board. The electronics to read the sensor are now on the sensor itself. The newly designed acquisition board from fig. 5 is a 4-layer PCB with a ground plane and a power plane. Its dimensions are 64x39 mm. Compared to the previous board with dimensions of 120x69 mm, it is more than three times smaller. It has one FFC connector with 20 contacts. At the same time we have integrated a charging circuit for a Lipo battery. The board is designed to fit a Raspberry Pi Zero W on it. We use a low-profile header to reduce the height of the whole thing so it fits

easier into the casing. To reduce the power consumption, we power the acquisition board with 3.3V instead of 5V. Using the Raspberry Pi Zero W 3.3V pin, which can safely supply up to 500mA of current. The static power consumption reduces with 33% compared to 5V. The GPIO pins on the Raspberry Pi are 3.3V compatible and cannot handle 5V. Reducing the power supply also eliminates the need to level shift the GPIO pins.

D. Battery system

The Raspberry Pi Zero W is powered via the 5V pin. When the Raspberry Pi Zero W is powered via the 5V pin, it is not possible to use the Micro-USB PWR port as it is internally connected to the 5V pin. The Raspberry Pi Zero W uses a PAM2306 as buck converter to step down the 5V into 3.3V for the 3.3V pin on the GPIO header and 1.8V core voltage for the BCM2835. According to Raspberry Pi Foundation, all Raspberry Pi models need 5.1V to operate properly. However, the PAM2306, see fig. 6, tells us that it doesn't necessarily need 5.1V, but will work in the 3.5V to 5.5V range. For battery-powered applications, a lower voltage is preferred. Not only because of power consumption, but also because batteries usually not have a nominal voltage of 5V. Otherwise, we need to use a boost converter to step up the battery voltage to 5V. There are situations where 5V is required, such as when using the HDMI, Bluetooth, or Micro USB port. In our application, we will not be using any of these interfaces. This means that the Raspberry Pi Zero W should work fine even if we use a power supply with less than 5.1 V.

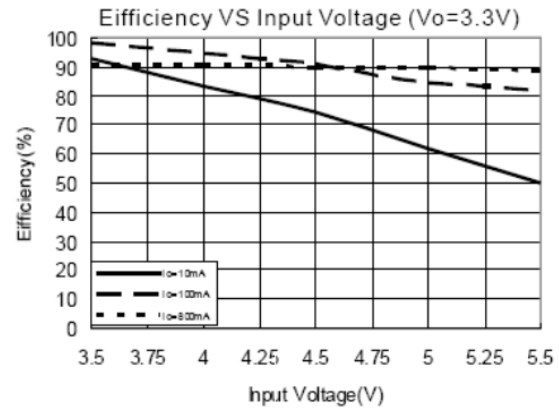


Fig. 6: PAM2306 buck converter

To determine if the Raspberry Pi Zero W still operates correctly at a lower base voltage, we performed measurements. We ran the Raspberry Pi at various voltages from 5V down to 3.4V. We measured power consumption in various states: during boot, idle, Bluetooth disabled, HDMI disabled, ACT Led disabled, and sleep mode. The resulting measurements are shown in table I. You can see that the current draw decreases when HDMI, Bluetooth and ACT Led are disabled. Lowering

the voltage increases the current draw. At a voltage below 3.5 V, the Raspberry Pi starts to have problems. This corresponds to the required range of the PAM2306, which is 5V to 3.5V. If the voltage is below 3.5V, the PAM2306 will not be able to output a stable 3.3V. The Raspberry Pi has a built-in shutdown functionality to save power. However, this mode is not very power efficient as it still draws about 35mA. The sleep mode only puts the core to sleep, the GPIO pins and the rest of the board is still powered.

TABLE I: Power consumption Raspberry Pi Zero W

Supply voltage	Boot (mA)	Idle (mA)	Bluetooth (mA)	HDMI (mA)	ACT led (mA)	Sleep (mA)
5V	240	122	114	97	95	35
4.2V	270	147	138	109	107	34
4V	280	157	142	112	110	34
3.7V	301	169	150	118	116	35
3.5V	312	175	158	125	122	37
3.4V	320	181	161	127	125	37

After determining that running the Raspberry Pi Zero W with a lower supply voltage is possible, we compared different battery types to find the most suitable battery for our application. The batteries compared are commercial batteries that are rechargeable. The batteries are listed in table II. The nominal voltage rating is important because if the voltage is less than 3.5 V, multiple units must be connected in series, which takes up more space in the casing. Higher capacity increases the operating time and battery life. Self-discharge shortens battery life and causes batteries to have less than a full charge when actually in use. Discharge current is the maximum current the battery can safely deliver. The number of cycles indicates how many times a battery can be charged before the capacity drops below 80%. NiCD and to some extent NiMH suffer from the memory effect which causes them to hold less charge. Lipo and Li-ion batteries seem to have the best overall characteristics for our application. It has the highest energy density. The nominal voltage of 3.7 V is higher than the 3.5 V required by the PAM2306.

TABLE II: Battery types

Battery type	Nom. voltage	Capacity (Wh/Kg)	Self discharge (Month)	Discharge current	Cycles (to 80%)
Lipo	3.7V	100-265	5-10%	1C	500-1000
Li-ion	3.7V	100-265	5-10%	1C	500-1000
NiMH	1.2V	60-120	10-20%	0.5C	300-500
NiCD	1.2V	40-60	20%	1C	1500
LiFePo4	3.2V	90-160	3%	1C	2000
Alkaline	1.5V	80	0.3%	0.2C	50

We use a 4400mAh Lipo battery pack from Kiwi Electronics. The battery pack consists of two 18650 cells with built-in protection. It includes under voltage, over voltage and short

circuit protection. This allows the batteries to be used safely. Lipo batteries have a usable voltage range of 4.2V to 3.0V. If the voltage drops below 3.0V, the battery may be permanently damaged. The PAM2306 operates down to 3.5V, which means we cannot use the full capacity of a Lipo battery. A boost converter can step up the battery voltage to 5V so we can use the full capacity. However, if we look at fig. 7, we see that with a discharge current of 0.2C about 94% of the capacity is stored between 4.2V and 3.5V. The battery life gained by using a boost converter is ideally 6%. However, a boost converter is not ideal and will have a power loss. Therefore, adding a boost converter that allows us to use the battery down to a lower voltage does not result in a capacity increase large enough to keep the system running longer.

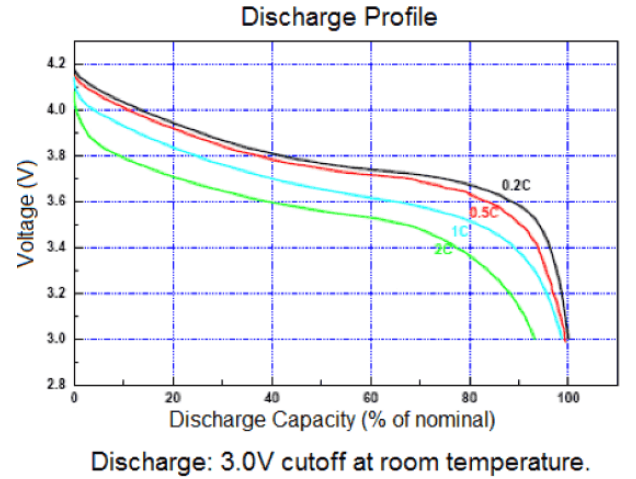


Fig. 7: Lipo discharge profile

To charge the Lipo battery, we added the MCP73833 charge management controller on the acquisition board. An onboard charger eliminates the need for an external charger or removing the battery from the carrier. The MCP73833 requires a 5V input and can charge the battery with a 1 A charge current while providing power to the board. We have provided two charging methods: wired charging and wireless charging. The wireless receiver used is the Universal Qi Wireless Receiver Module from Kiwi Electronics, see fig. 8. The wireless receiver has an output voltage between 4.8-5.2 V and a output current of 500 mA.



Fig. 8: Wireless receiver coil

The Raspberry Pi Zero W is a microprocessor running an operating system. Abruptly cutting the power could crash the operating system and damage the SD card. The state of the operating system need to be managed before cutting the power of the Raspberry Pi Zero W. A low battery voltage could also cause an abrupt lose in power. We use a simple battery monitoring circuit that will signal the Raspberry Pi Zero W when the voltage has dropped below a threshold. We measure two voltage levels, one to indicate low battery voltage and one to initiate a safe shutdown to prevent an sudden loss of power. The low battery voltage indicator is set to 3.69V, which corresponds to a discharge of 80% and the safe shutdown indicator is set to 3.51V at a discharge of 90%.

E. new e-Cone prototype

The newly build e-Cone prototype is shown in fig. 9. The sensor is attached to the outside of the case with double-sided tape. On the top cover you can see a power switch, shutdown button and charging LEDs. Using a shutdown button the Raspberry Pi Zero W will go into sleep mode so that the power can be disconnected safely.

TABLE III: Prototype power consumption

Supply voltage	Idle (mA)	Active (mA)	Sleep (mA)
4.2V	120	200	43
4.0V	125	211	46
3.8V	130	234	47
3.75V	133	239	48

We measured the power consumption of the prototype, as shown in table III. Three states are measured: Idle, Active and Sleep. Idle is when no client is connected. Active is when a client is connected and the sensor is operating. The Raspberry Pi Zero W alone can run at 3.5V with WiFi still operation. When the acquisition board and sensor are connected the WiFi stops working when the battery voltage gets below 3.75V. Due to the additional power consumption. Therefore, we did not measure the power consumption below 3.75V when we lose WiFi connection. The effect on the operating time is limited, as 80% of the Lipo discharge capacity is stored above 3.7V.



Fig. 9: e-Cone Prototype with high resolution sensor

V. SIMULATION ACQUISITION CIRCUIT

The sensor has 8192 sensor pixels that we need to sample. We want a frame rate of 15 fps. That means we need to sample 122880 data points per second. We need to ensure that the signal settles in time to be read by ADC. If the signal does not settle in time, we will get settling errors and our measurements will not be accurate. To verify the behaviour of our circuit, we performed simulations using LTSpice. The LTSpice component library does not include a model of the ADS8862. We created a SAR ADC model based on the input sampling stage equivalent circuit from the ADS8862 datasheet. We use the minimum acquisition time t_{acq} , the conversion time t_{conv} , the input capacitance C_{sh} and the series resistance R_{sh} . The value of R_{sh} can be derived using equation 2. An overview of the circuit using the SAR ADC model is shown in fig. 10.

$$R_{sh} = \frac{t_{acq}}{100 * C_{sh}} \quad (2)$$

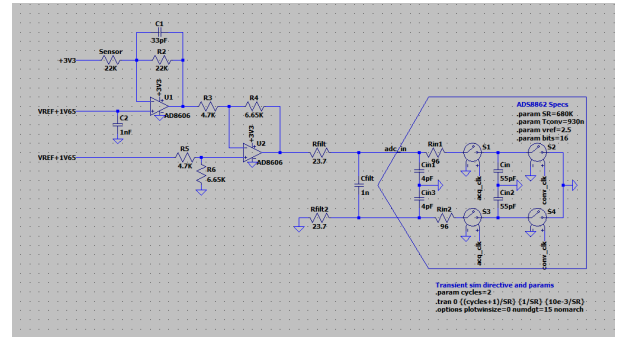


Fig. 10: Acquisition circuit in LTSpice

The ADS8862 has a throughput rate of 680KHz, an acquisition time of $t_{acq}=540\text{ns}$, and a conversion time $t_{con}=930\text{ns}$. In practise, it is sufficient for the internal sample-and-hold capacitor of the ADC to settle to less than $\frac{1}{2}\text{LSB}$ within the acquisition phase. Since settling to a higher accuracy does not improve ADC accuracy due to quantization noise. The reference voltage is set to 2.5V. Using the equation 3, we can calculate the maximum settling error that must not be exceed as $19.07\mu\text{V}$.

$$V_{Settle} = \frac{1}{2}\text{LSB} = \frac{V_{Ref}}{2^{(bits+1)}} \quad (3)$$

At the end of the conversion phase, the internal sample and hold capacitors are discharged to ground. This causes a voltage drop at the input at the beginning of the acquisition phase. The circuit driving the input must bring the voltage back up before the end of the acquisition phase. The t_{acq} should be greater than the time to recover from the voltage kickback.

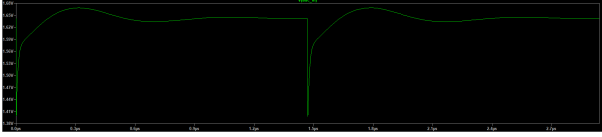


Fig. 11: Settling behaviour without buffer

In fig. 11 you can see what happens when the system is unable to bring the voltage back up after discharge. An unwanted oscillation, also called ringing, occurs on the input signal of the ADC with a large settling error as a result. When the t_{acq} is over, we still have an error of about 3.9326mV , which is well above the required V_{Settle} of $19.07\mu\text{V}$.

$$GBW \geq 4 * \left(\frac{1}{2\pi(R_{Filt} + R_{Filt}) * C_{Filt}} \right) \quad (4)$$

An ADC driver amplifier helps bring the voltage back up after discharging. Using the equation 4 we find that a GBW of at least 13.35MHz is required. This is larger than the 10MHz of the AD8606, which explains the ringing on the input signal. Therefore, we added the AD8031 as an input driver to achieve the desired accuracy. The AD8031 has a bandwidth of 80MHz and is well suited for high speed, A/D buffers and battery-operated systems. The improved circuit is shown in fig. 12.

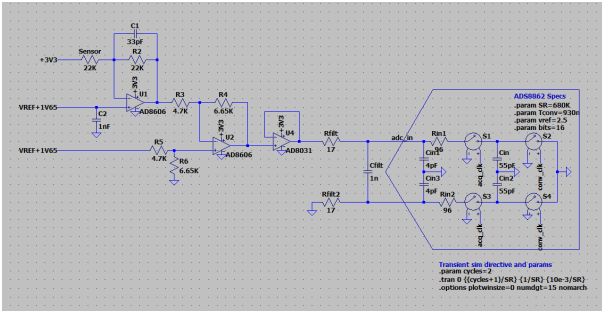


Fig. 12: Acquisition circuit with driver amplifier

According to fig. 13, the input signal of the ADC contains no ringing. The settling error reduces to 170.41nV , well below the required $\frac{1}{2}\text{LSB}$ of $19.07\mu\text{V}$.

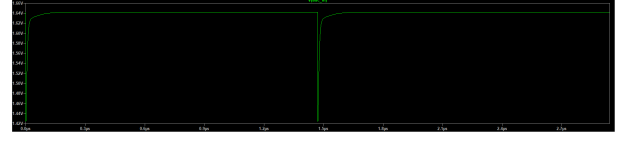


Fig. 13: Settling behaviour with buffer

By optimizing RC charge bucket filter, we can further reduce the settling error. We use Analog Engineers Calculator, to find the value range for the resistors. In LTSpice, we perform a step sweep based on the values given by Analog Engineers Calculator. All resistor values are standard 1% values. The filter capacitor C_{Filt} must be at least 10 times larger than the sample and hold capacitor. We chose a C_{Filt} of 1.1nF . With this simulation, we found a suitable value for R_{Filt} as 17Ω with this value, the settling error is less than 1nV and well below V_{Settle} before the end of t_{acq} of the ADS8862.

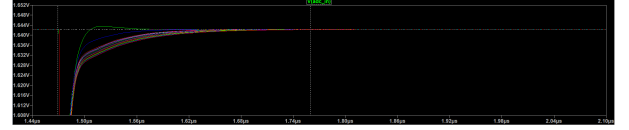


Fig. 14: Step sweep

VI. SOFTWARE

The Raspberry Pi Zero W is set up as a wireless access point that one client can connect to at a time. We chose NGINX as the web server because it is lightweight and powerful. The overall software can be viewed at fig. 15. Once a client connects to the e-Cone via WiFi and starts the stream, the configuration parameters are sent as JSON object to the Raspberry Pi Zero W via an XMLHttpRequest. On the server, a PHP script writes the received parameters to a text file and start and stop the C++ program via executable shell script. The shell script terminates an already running e-Cone process and restarts it. Before each sensor scan, the C++ program reads the parameters from the text file to check for changes. The resulting image is written to a memory buffer and sent to the connected client over a TCP connection using port 8080. The C++ program will scan the sensor until the stream is stopped by the user. We use the Serial Peripheral Interface (SPI) to read the ADC, set the DAC channel voltage and control the multiplexers. For this we have chosen a C library for Raspberry Pi called BCM2835. It provides access to GPIO and other IO functions on the Broadcom BCM 2835 chip of the Raspberry Pi Zero W.

The ADG731 and MAX14661 do not support daisy chaining. The SPI clock frequency is set to 10MHz . We need to write one configuration byte to the ADG731 and 8 configuration bytes to the MAX14661. This makes the configuration of the MAX14661 slower than that of the ADG731. Therefore,

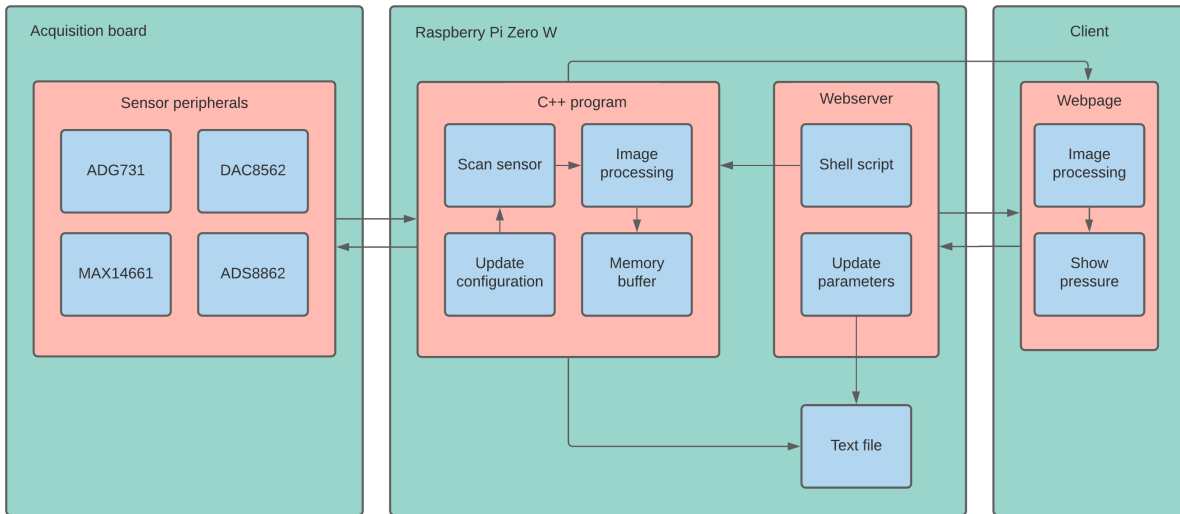


Fig. 15: Overall software

we try to minimize the number of changes in row configuration for optimal throughput. We read the sensor one row at a time. That is, we select a row by configuring the MAX14661 and read all columns iteratively. In this way, it takes 36 ms to scan the sensor.

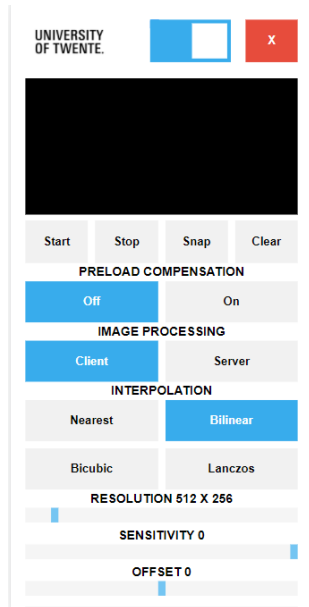


Fig. 16: e-Cone user interface on a Pixel 2

The UI allows a client to interact with the e-Cone. The UI is written in HTML5, CSS3 and Javascript. To make the UI suitable for different devices, we use a responsive design that adjusts depending on the screen width. Any device with a web browser and WiFi is able to connect. The UI is presented in fig. 16. At the top is a navigation bar with a logo, a view toggle to display an additional snapshot view and a shutdown

button that puts the Raspberry Pi Zero W into sleep mode. Then we have the stream and snapshot view. Settings that can be configured are: Preload, Image processing side, Resolution, Interpolation Method, Sensitivity and Offset. Finally, there is a reset button to reset all configurable settings to default.

A. Interpolation methods

The sensor has a high resolution of 128x64, but for an image this resolution is rather low. To increase the resolution of the images, we use interpolation. The previous interpolation methods were too slow. We explore if there are faster methods that can be used without sacrificing too much frame rate. The interpolation libraries compared are a Student, FreeImage 3.17 and OpenCV 4.5.0. We compared the following interpolation methods: Nearest Neighbor, Bilinear, Bicubic and Lanczos. In table IV, the result of interpolation methods between different libraries is shown. The input image has a resolution of 128x64 and is interpolated to an image with a resolution of 512x256.

TABLE IV: Comparison interpolation methods

Library	Nearest Neighbour (ms)	Bilinear (ms)	Bicubic (ms)	Lanczos (ms)
Student	31	68	702	-
FreeImage	20	23	34	47
OpenCV	11	8	13	22

From the results of table IV, we see that OpenCV has the most optimized interpolation methods. Examples of interpolation using the OpenCV methods are shown in fig. 18. To make the differences between the methods more obvious, the images are colorized after the interpolation has been applied. The example image shown in fig. 17 is used as input. The bicubic interpolation seems to give the most satisfactory results. The image is smoother than the bilinear method and has fewer artifacts than the Lanczos method.



Fig. 17: Sensor readout

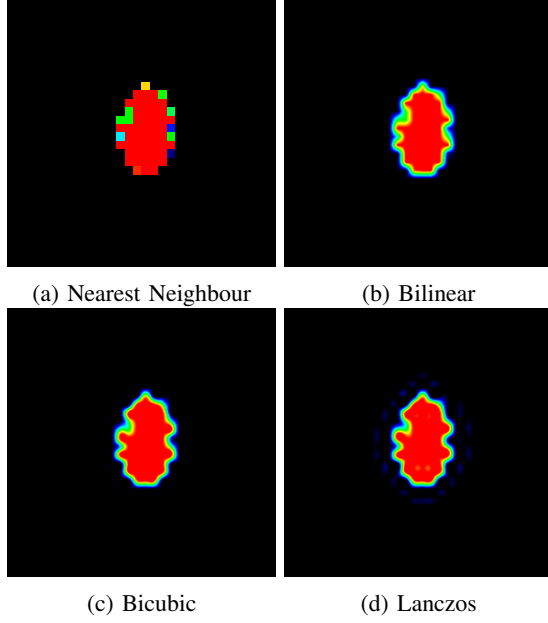


Fig. 18: Comparison of interpolation methods

B. Colorization methods

To make the pressure images more understandable to humans, we apply colorization. Each pixel has a grayscale intensity value that is mapped to a corresponding RGB value. We use a 16-bit ADC, which gives us values ranging from 0 to 65535. The color scheme used is presented below in table V. It consists of 6 parts with a gradual color change.

TABLE V: Color scheme

Grayscale	RGB	Color
0 ~ 12000	0.0.0 ~ 0.0.255	Black ~ Blue
12000 ~ 24000	0.0.255 ~ 0.255.255	Blue ~ Cyan
24000 ~ 36000	0.255.255 ~ 0.255.0	Cyan ~ Green
36000 ~ 48000	0.255.0 ~ 255.255.0	Green ~ Yellow
48000 ~ 60000	255.255.0 ~ 255.0.0	Yellow ~ Red
60000 ~ 65535	255.0.0	Red

The simplest way to color an grayscale image is to use an iterative method. This iterates through all the pixels in the image. For each pixel, the RGB values are calculated based on the grayscale value of that pixel. This method requires that a calculation be performed for each pixel. A more efficient method of coloring a grayscale image is to use a lookup table (LUT). Using a LUT, the RGB values are calculated once for each possible grayscale value and stored in an array. When we

iterate through the image, we don't have to do any calculations, we just look up the RGB values in the array.

For the selected libraries, we compared the complete image processing step consisting of interpolation, colorization, and storing the image. The input is an 128x64 image and the output is a colorized 512x256 PNG image. In table VI, it can be seen that the OpenCV methods are significantly faster than the FreeImage or Student methods. Using a LUT and a memory buffer further reduces processing time.

TABLE VI: Image processing time

Library	Bilinear Interpolation	Colorize (ms)	Write image (ms)	Total (ms)
Student	68	93	93	254
FreeImage	23	35	70	128
OpenCV	8	22	32	62
OpenCV + LUT	8	10	32	50
OpenCV + LUT + Memory buffer	8	10	29	47

After we create a 16-bit image, it needs to be stored somewhere. We can write the image to the SD card of the Raspberry Pi. However, excessive writing to a SD card will eventually lead to failure. To avoid damaging the SD card, images can be encoded in memory. Using a memory buffer is a little faster than writing an image to the SD card. The encoding time of an image depends on both the resolution and the type (RGB or grayscale). If we look at table VII, we can see that encoding the original 128x64 image in a memory buffer takes only 1.3 ms. If we apply bilinear interpolation to increase the resolution to 512x256, we see that both the encoding time and the image size increase. If we then apply colorization, it takes even longer, but the image size seems to decrease a bit. When we increase the resolution, both the encoding time and the image size increase even more.

TABLE VII: Encoding image in memory buffer

Image	Encoding (ms)	Buffer (bytes)
Sensor readout	1.3	301
Interpolated	19.6	2923
Interpolated + Coloring	29.0	2831

C. Client side image processing

To improve the frame rate and reduce the load on the Raspberry Pi Zero W, we use client-side image processing. We assume that the client device is a smartphone, tablet or laptop with more processing power than a Raspberry Pi Zero W. Therefore, it should be able to perform image processing faster than a Raspberry Pi Zero W.

The Canvas API provides a means to draw graphics on the screen using HTML5 and JavaScript. It allows pixel manipulation of 2D images. To access the pixel data, the image

must be loaded into a canvas element. We use two canvas elements. An offscreen canvas to load the received sensor image and perform pixel manipulation. The processed image is then loaded into an onscreen canvas. The use of an offscreen canvas prevents intermediate results from being drawn on the screen. The first step is to colorize a grayscale image on the client-side. As on the server, we do this with a lookup table (LUT) in JavaScript. A HTML5 canvas only supports 8-bit images, so all images and therefore all pixel manipulations are done with 8-bit values. We use the same color scheme as in table V, only with values ranging from 0 to 255. To perform the interpolation, we use the OpenCV 3.4.0 JavaScript library.

TABLE VIII: Image processing on Dell XPS 15

Resolution	Colorize (ms)	Bilinear (ms)	Bicubic (ms)	Lanczos (ms)
256x128	0.4	8.8	10.1	12.1
384x192	0.7	9.4	12.7	15.2
512x256	1.5	11.2	15.3	19.1
640x320	2.2	13.1	16.9	22.6
768x384	3.1	14.2	17.8	29.5
896x448	3.5	15.7	20.6	37.1
1024x512	5.5	17.4	24.5	45.2

The measurements shown in table VIII were performed on a Dell XPS 15 9570 with 8th Gen i7 and 16GB RAM. The input is a 128x64 grayscale image. The colorization is faster than that on the Raspberry Pi Zero W. However, the interpolation is slower than that on the Raspberry Pi Zero W. The results are not the same on every device. It can be faster or slower depending on the processing power of the client device.

D. Preload compensation

A side effect of bending the sensor around the casing is that it causes an initial bending pressure on the sensor. This reduces the available pressure range and introduces unwanted noise in our images. We do not want preload values to be visible in the images since it is not a user pressure. Therefore, we apply preload compensation to reduce the visibility in images. We do this on the Raspberry Pi Zero W, by scanning the sensor without pressure, taking this matrix and subtracting it from each scan forward on. It is important that no pressure is applied during the preload scan as this will affect the image. It costs the Raspberry Pi Zero W 0.3 ms to perform the matrix subtraction. This has little impact on our frame time and frame rate. In flat position, the sensor has average values between 209 and 213. The entire sensor has homogeneous values. After preload subtraction, the sensor pixel values are reduced to the range of 0 to 3. In table IX, the difference that preload compensation makes is shown. On the left, you can see a portion of a sensor scan taken without pressure applied to the sensor. On the right, you see the same sensor pixels after subtracting the preload values. With flat orientation, the average preload value of a sensor pixel is less than 5% of the maximum value.

TABLE IX: Flat sensor preload

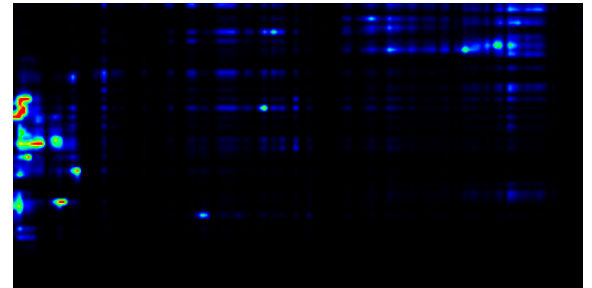
212	213	317	344	212	212	0	0	0	0	0	0
212	212	212	212	211	211	0	0	0	0	0	0
213	213	212	212	212	214	0	1	0	0	0	0
212	212	213	212	212	212	0	1	2	0	0	0
213	212	213	212	213	216	1	0	0	0	0	1
212	214	209	211	209	212	0	1	0	0	0	1

Preload on the curved sensor is more severe than on the flat sensor. However, it is not just the result of bending pressure. There are air bubbles between the sensor and the material due to difficulties in attaching the sensor to the casing. Also the end of the sensor is fixed with adhesive tape that results in additional pressure. If we look at the preload caused by the bending, the values only increase by about 13% compared to the flat sensor. The average preload value of a sensor pixel caused by bending is still less than 5% of the maximum value.

TABLE X: Bended sensor preload

253	253	253	253	253	254	0	0	0	0	0	0
254	253	254	254	253	254	0	0	1	0	0	0
254	254	254	253	254	254	0	1	0	1	1	0
254	253	254	252	253	254	0	0	1	0	0	0
253	252	253	254	254	253	2	0	0	0	0	0
254	253	254	253	254	253	0	1	0	0	0	0

An example of preload on a bended sensor is shown in fig. 19. Above the initial preload on the bended sensor is shown. On the left you can see where the tape is placed as there is significant preload. Also the right top corner is where a air bubble is. Below the result after removing the preload is shown. Most of this pressure can be removed with preload compensation.



(a) Initial preload



(b) After preload compensation

Fig. 19: Preload on the bended sensor

VII. EXPERIMENTS

A. Sensor characteristics

To determine the relationship between the pressure applied and the reading of the sensor, we conducted an experiment. We applied forces to the sensor with copper weights of 78 grams each. We measured the average pressure over the area to which the force was applied. The area to which the weights are applied is 6.25cm^2 . This corresponds to about 256 sensor pixels. The experiment is repeated at different locations on the sensor to check for differences in the sensor. Here, the pressure P is defined as the force F applied to an area A that is perpendicular to the force. A given force can have a different effect depending on the area over which the force is applied. The pressure is measured in pascal where $1\text{PA} = 1\text{N}/\text{m}^2$.

$$P = \frac{F}{A} \quad (5)$$

In this experiment, R_K is fixed at $22\text{K}\Omega$ and V_{test} is varied. The resistance of R_k determines the sensitivity of the sensor. A higher value for R_K gives the sensor a higher sensitivity. The test setup can be viewed in fig. 20.

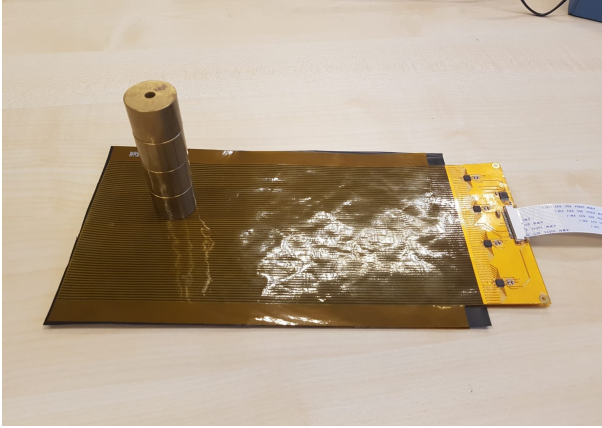


Fig. 20: Test setup with copper weights

The output voltage V_{out} of the sensor ranges from 0V to 1.65V. The reference voltage of the ADC is set to 2.5V, giving a measurable range of 0 to 2.5V. We do not use the full range of the ADC. Therefore, we add a small gain to V_{out} . This is done in fig. 12, by setting R_3 and R_5 to 4.7K and R_4 and R_6 to 6.65K. This gives us a voltage range of 0V to 2.33V, which corresponds to a ADC value of 0 to 61199. This gives us a little margin to tune the signal with an offset without clipping.

$$V_{opamp} = \frac{R_3}{R_4} * (V_2 - V_1) \quad (6)$$

The measured points are plotted in fig. 21, 22 and 23. It can be observed that in all measurements, the higher V_{test} leads to a larger pressure and thus to a higher sensitivity. Differences between the points are visible. In all three measurements, it is seen that the average pressure flattens out as more weights are applied.

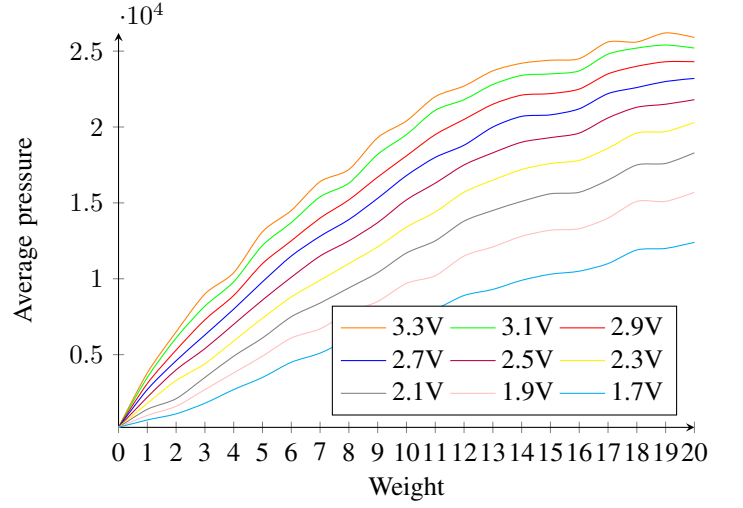


Fig. 21: Sensor characteristics at point 1 (44,29)

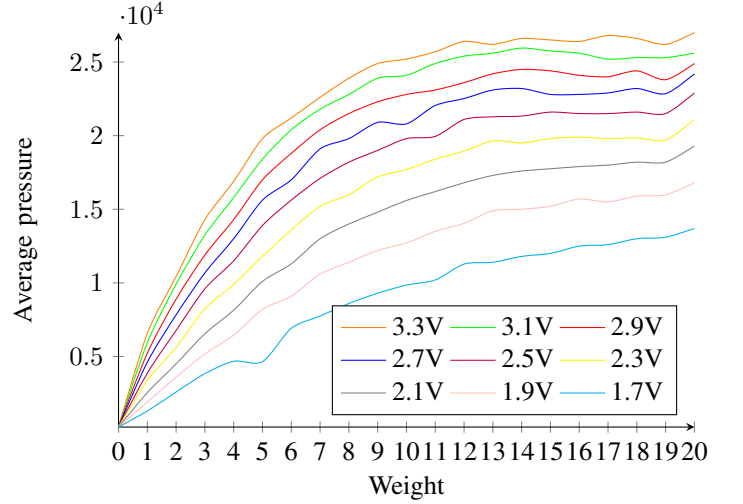


Fig. 22: Sensor characteristics at point 2 (14,74)

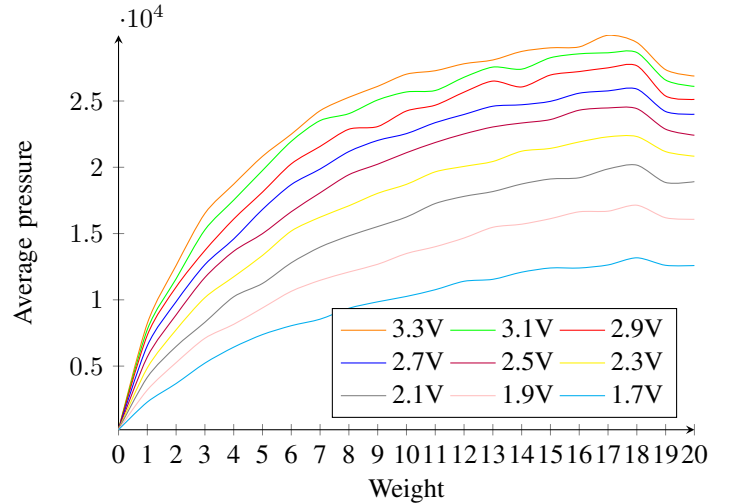


Fig. 23: Sensor characteristics at point 3 (27,98)

B. Hysteresis

There is a difference in the output value between series of increasing measurements and series of decreasing measurements, this is called hysteresis. Hysteresis is caused by the natural reluctance of a material to return to its original state after a load is added and removed. We define hysteresis as the maximum difference for a single load.

$$Hysteresis = \frac{Y_d - Y_i}{Y_{max} - Y_{min}} * 100\% \quad (7)$$

Here Y_d and Y_i are the pressure values of the decreasing and increasing series, and Y_{min} and Y_{max} are the minimum and maximum pressure values. The hysteresis error is measured at several points, as shown in table XI.

TABLE XI: Hysteresis errors

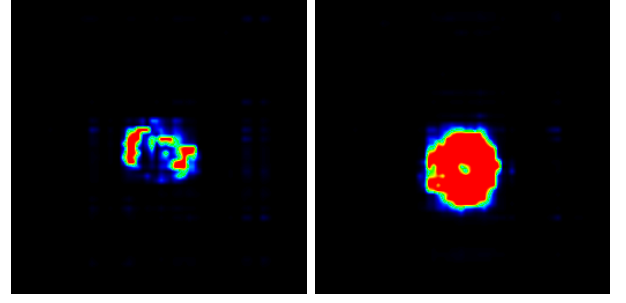
Weight	1	5	10	15
Point 1 (8,6)	3.59%	25.67%	13.25%	5.41%
Point 2 (40,40)	4.88%	20.39%	8.96%	4.83%
Point 3 (99,24)	12.08%	11.88%	8.94%	0.25%
Point 4 (90,36)	4.68%	12.50%	6.34%	1.17%

We use copper weights to increase and decrease the load on the sensor. With a maximum of 20 weights. The voltage V_{test} is set to 3.3V. We measure the average pressure applied to an area of 16x16 sensor pixels. In all measurement points, the decreasing series has higher values than the increasing series. After applying a high load to the sensor for some time, the Velostat seems to remember this pressure for a while after the load is removed.

C. Weight distribution

In conducting the sensor characteristic experiment, we found that the sensor has difficulty detecting pressure of hard objects. The sensor works by measuring pressure at a sensor pixel, a point where row and column intersect. Pressure that is not applied at a sensor pixel is not measured. For a hard object, most of the pressure is applied to a small portion of the object's surface. Therefore, the pressure is not properly distributed over the object's surface. Since most of the pressure is applied to a few sensor pixels, the values are often limited to the maximum value even though the actual value is higher. This makes it difficult to determine how closely the measured value of the pressure sensor comes to the actual pressure. When we place the copper weights used in the experiment, see fig. 20, on the sensor, we expect to see a round shape with a hole in the middle. An example is illustrated in fig. 24. On the left side, we have 7 weights totaling 546 grams placed on the sensor. On the right side, we use a single weight of 78 grams with a foam layer between the weight and the sensor. Notice the difference that the left image with seven weights barely gives a result, while the right image with a single weight gives a good result. Increasing the weight does not result in a better distribution of pressure. The problem of weight distribution only occurs with

hard objects. A finger or hand is mostly made of soft tissue, which already distributes the pressure by itself.

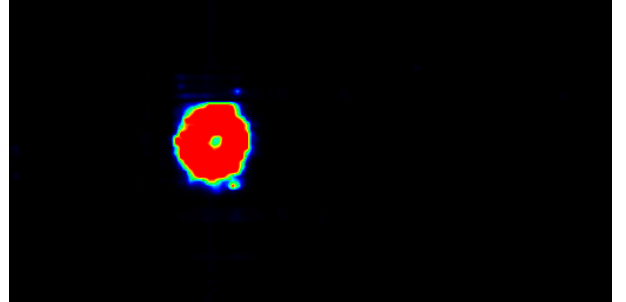


(a) Poor pressure distribution (b) Good pressure distribution

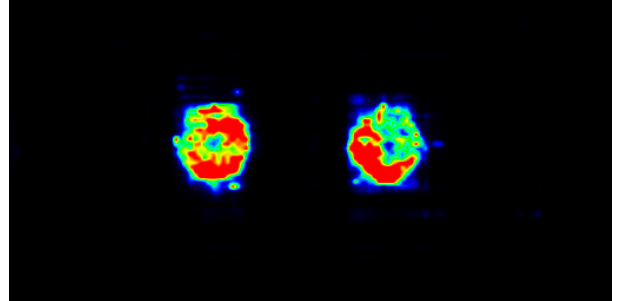
Fig. 24: Comparison of interpolation methods

D. Cross-talk

In this prototype, we use a sensor with a 128x64 resolution. Large sensors are more susceptible to crosstalk. Crosstalk is a phenomenon where the measurement of one sensor pixel is affected by other sensor pixels on the sensor. In our sensor, crosstalk is also a problem. It occurs in the row-column architecture due to the formation of parasitic parallel paths. An example of crosstalk in our sensor is shown in fig. 25a and fig. 25b.



(a) A single object on the sensor



(b) Two objects on the sensor

Fig. 25: Effect of crosstalk on the sensor

First, a single copper weight is placed on the sensor as a reference. The copper weight is placed on a foam layer and a constant pressure is applied. When a second copper weight is placed horizontally next to it, it can be seen that the weights affect each other. Without a drop in pressure, the reading of

the first weight decreases. If we calculate the average pressure of the single weight, we get 40048. If we calculate the average pressure after placing the second weight, we see that the pressure has dropped to 20973. This is a pressure drop of 48%. The average pressure is measured over a square area of 24x24 sensor pixels. The crosstalk is most noticeable when the pressures are applied in horizontal direction. When the pressures are applied vertically, the crosstalk is not as severe because the input rows are different.

E. Handgrip visualisation

The purpose of the e-Cone is to visualise the distribution of the forces of a hand to grasp an object. In fig. 26, a left hand is shown pressed onto the pressure sensor in a flat orientation. You can see the contours of the hand, the palm and the fingertips. However the fingers are not detected by the sensor.

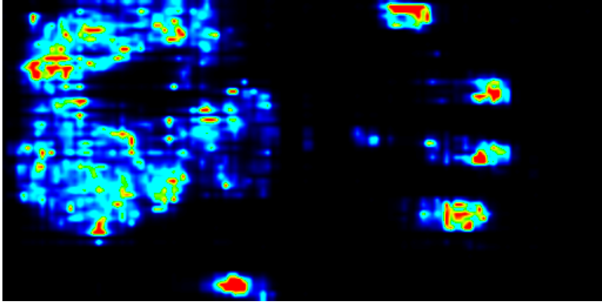


Fig. 26: A hand pressed on a flat sensor

In fig. 27 a right hand is shown grasping the prototype from fig. 9. On the left side you can see four fingertips and on the right side you can see the palm of the hand. Again the fingers are not detected by the sensor.

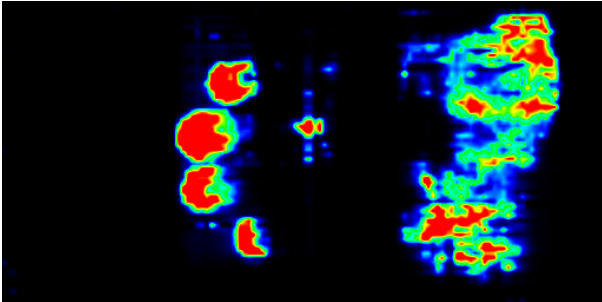


Fig. 27: A hand pressed on a curved sensor

VIII. DISCUSSION

A previous research showed a low-cost high-resolution pressure sensor using the piezoresistive material Velostat as a possible replacement for commercial sensors. In this work, we built a next e-Cone prototype with a 128x64 pressure sensor. About 81% of the e-Cone's curved surface is covered by the sensor. Using client-side image processing, we are able to

provide visual feedback at a resolution of 512x256 and a frame rate of 15 fps. A significant improvement over previous work.

The pressure sensor is capable of visualising the force distribution of a hand. However, we find that crosstalk has a significant impact on our results. A hand is held in the horizontal direction, which is also where the crosstalk is strongest. Not all of the pressure applied in the horizontal direction is measured. In the measurements, we have seen that crosstalk causes a 48% decrease in the pressure reading. When a hand is held vertically, more of the hand is detected because the crosstalk is not as strong in that direction.

There are also concerns about the feasibility of measuring the force acting on the sensor. In addition to crosstalk, the sensor is also affected by weight distribution and hysteresis. Poor weight distribution will result in a force being sensed by fewer sensor pixels. After a force is applied to the sensor, the sensor remembers that pressure to some degree for some time. We have found hysteresis errors as high as 25.67%. The combination of these causes makes it difficult to quantify a force applied to the sensor.

Another difficulty is attaching the sensor to the curved surface of the casing. To minimize bending pressure, the sensor is attached to the casing layer by layer, rather than the entire sensor at once. The sensor must be perfectly aligned. Once glued together, it is almost impossible to separate the layers again without damaging the sensor. Also, there must be no air bubbles between the layers. Air bubbles between the layers create visible noise in the images. To protect the sensor gap and the outer end of the sensor, we cover it with adhesive tape. This also results in a preload on the sensor.

To improve weight distribution, we experimented with multiple layers of Velostat and a foam layer on top or underneath the sensor. Multiple separate layers of Velostat did not work well. A foam layer under the sensor seems to improve pressure distribution somewhat, but the sensor is deformed and potentially damaged. A foam layer on top seems to distribute pressure much better. Using foam also increases the pressure a bit beyond the surface of the object. However, measurements are still difficult with the black ESD foam we use. Since the sensor is not calibrated appropriately for the foam.

Furthermore, we explored the possibility of placing the e-Cone within a local WiFi network. This allows all devices on the network to connect to the e-Cone and maintain an internet connection. So you do not have to change WiFi networks. On the local WiFi network we used, the e-Cone was unusable. There was a severe delay in streaming, which had a negative impact on performance. Therefore, we decided to use the e-Cone as a standalone access point.

We also observed interference between the HDMI port, Bluetooth, and the sensor. When the HDMI port and Bluetooth were enabled, the images had visible noise. This can easily be solved by disabling the HDMI port and Bluetooth in the software, but should be mentioned.

IX. FUTURE WORK

Future research should look at methods to reduce crosstalk and improve the weight distribution of the sensor. This would improve both the quality of the handgrip images and the ability to measure force with the sensor. We are only using one type of piezoresistive Velostat. It would also be interesting to study the effect of different piezoresistive materials. Caplinq, the manufacturer of Velostat, develops a wide range of conductive piezoresistive materials with different thicknesses and properties. A next research could explore different piezoresistive materials.

X. CONCLUSION

The goal of this work is to determine if the high-resolution pressure sensor can replace a commercial sensor for the purpose of e-Cone. We defined the main research question as follows: Is the high-resolution pressure sensor suitable for use in the e-Cone? To answer this research question, we have defined three sub-questions, which we will answer first. In section VI, we discussed about the image processing capabilities. Bicubic interpolation seems to give the best results. On its own, the Raspberry Pi Zero W is not capable of performing all the image processing required. With a combination of client-side and server-side image processing, it is possible to achieve the desired frame rate of 15 fps at a resolution of 512x256. If you use only client-side image processing, you can achieve even higher resolution images at a frame rate of 15 fps. This answers the first sub-question, what kind of image processing is possible.

The Raspberry Pi Zero W has relatively high power consumption for a battery-powered application, see table I. The power consumption of the prototype is shown in table III. Of the batteries compared in table II, we chose the Lipo battery. With a 4400 mAh LiPo battery, it is possible to achieve a full working day on one battery charge. With the integrated wireless charger, the e-Cone works completely wireless. This answers the second sub-question of making the device wireless by integrating a battery system.

We have built a working e-Cone prototype using a Raspberry Pi Zero W. However, the Raspberry Pi Zero W is pushed to its limits. The Raspberry Pi Zero W does not have enough computing power to perform image processing on its own. A client-side device is needed to do some of the image processing. In addition, the Raspberry Pi Zero W is the limiting factor when scanning the sensor. Therefore, there are concerns about how future-proof the Raspberry Pi Zero W actually is. However, we can conclude that a Raspberry Pi Zero W is suitable for the purpose of the e-Cone. This answers the third sub-question about the suitability of the Raspberry Pi Zero W.

With the sensor in flat orientation, we have an initial preload of 5%. Bending the sensor around the curved casing increases the preload a little. However, the increase caused by the

bending is not significant. The average pressure is still less than 5% of the maximum value, as can be seen in table X. By using preload compensation, we can remove much of the pressure from the results, as seen in fig. 19. So bending the sensor does not have much effect on the measurements. This answers our last sub-question about the effect of bending the sensor.

At last we can answer the main research question. Is the high resolution pressure sensor suitable for use in the e-Cone? The sensor can be used to visualise a handgrip pattern. However, the pressure images are not as complete as we had hoped. We find it difficult to obtain images of a complete hand where the palm, fingers and fingertips are visible at the same time. Measurements have shown that crosstalk causes a 48% drop in readings. The sensor is not yet suitable for force measurement. Further improvements are necessary to improve the usability of the sensor.

REFERENCES

- [1] Texas instruments. Hand function, participation and quality of life (qol) among people with hand conditions. *SDAA011A*, Aug. 2020.
- [2] Wiesław Zarzycki Zenon Mariak Maria Górka Elzbieta Krajewska-Kulak Jolanta Lewko, Jan Kochanowicz. Poor hand function in diabetics: Its causes and effects on the quality of life. *Saudi Medical Journal*, Apr. 2012.
- [3] Andrew J. Butler Steven L. Wolf Jay L. Alberts Nancy G. Kutner, Rebecca Zhang. Quality-of-life change associated with robotic-assisted therapy to improve hand motor function in patients with subacute stroke: A randomized clinical trial. *Physical Therapy*, Feb. 2010.
- [4] Lars B. Dahlin Ingrid Reitan and Hans-Eric Rosberg. Patient-reported quality of life and hand disability in elderly patients after a traumatic hand injury – a retrospective study. *American Journal of Occupational Therapy*, Aug. 2020.
- [5] R. Porzel M. Bonfert and R. Malaka. Get a grip! handgrip strength as a health screening tool. *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages pp. 604–612, 2014.
- [6] A. Hoeksma L. Roorda P. Kroon L. Spreeuwers, A. Aarden and D. Spreeuwers. Ontwikkeling van een exercise cone met druksensoren voor revalidatie en behoud van werkvermogen van reumapatiënten. verslag van de haalbaarheidsstudie. 2008.
- [7] P. Wang L. Wang, T. Ding. Thin flexible pressure sensor array based on carbon black/silicone rubber nanocomposite. *IEEE Sensors Journal*, Vol. 9(No. 9):1130–1145, 2009.
- [8] A.W. Setiawan et al S.S. Suprpto. Low-cost pressure sensor matrix using velostat. *5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, pages pp. 137–140, 6-7 Nov. 2017.