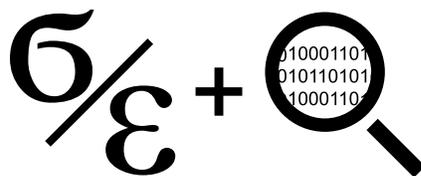


MATERIAL ANALYTICS

Knowledge from Mechanical Behaviour Data

AHMED HOSNY

Submitted to the Graduate School of Design
for the partial fulfillment of the requirements for the degree of
Master in Design Studies in Technology
at
Harvard University



Advisor: Assistant Professor Panagiotis Michalatos
May 2015 – version 2.4

ABSTRACT

This research aims to record, quantify and study material behavior within a data analytics framework. It does so while harnessing the power of the so-called "big data" in physical environments using both analytical and experimental methods. Current structural health monitoring systems are one-off applications that are non-scalable and only act as alert mechanisms. The focus is thus on a material-data interface that is scalable, developed for mass produced objects and is to be utilized during deployment stages. Daily sensor-embedded objects are placed online and hence have their deformation data logged onto the cloud. Data mining is then used to extract useful insights as well as predict future trends. By selectively reducing the degrees of strain measurement within objects and coupling that with machine learning, the research showcases relatively good accuracy in predicting displacements and loads acting on objects. The aggregation of such data over time allows for new design iteration workflows or data-informed design, aids in developing user-object interaction models and allows for a better understanding of challenging load cases that are difficult to model such as fatigue. A foundation for the *Internet of Materials* is thus proposed bridging sensor technologies, machine learning and the Internet of Things.

CONTENTS

1	INTRODUCTION	1
2	THESIS	9
3	INITIAL STUDIES	13
4	DIGITAL WORK FLOW	19
4.1	Sensor Deployment	19
4.1.1	FEA simulation routine	20
4.1.2	Data extraction	21
4.1.3	Identifying optimum sensor location	22
4.2	Physical Training	27
4.2.1	Preparation of physical specimen	27
4.2.2	Material trainer apparatus	28
4.2.3	Computer vision	30
4.2.4	Data exploration	33
4.3	Machine Learning	35
4.4	System Architecture & Data Mining	39
4.4.1	Sensor-embedded objects	39
4.4.2	Cloud prediction & database	40
4.4.3	Realtime visualization	41
4.4.4	Data mining & design recommendations	43
4.4.5	Fatigue prediction	48
5	DISCUSSION	51
5.1	Identifying application scenarios	51
5.2	Data-driven design vs. design intuition	54
A	APPENDIX	57
	BIBLIOGRAPHY	59

LIST OF FIGURES

Figure 1	DIKW pyramid	1
Figure 2	A Pressure and Strain Sensor Fabricated on Soft Artificial Skin by Yong-Lae Park et al.	2
Figure 3	Exaggerated deformation patterns of both LUSAS models. Solid and openwork flyer system with outward (left-pointing) wind load added [22]	3
Figure 4	Amiens cathedral, choir flyers with additional struts [22]	3
Figure 5	Cologne cathedra, choir flyers [22]	4
Figure 6	Various mechanical and optical strain guages [2]	4
Figure 7	Layout of the geometric levelling network at the base of the dome. Green and red lines represent the internal and external networks, respectively. The blue point (3000) is hung on a cable that is connected to the top of the spire. [3]	5
Figure 8	Directionality of reflected light is lower at different densities (top) and media controller (bottom) [24]	6
Figure 9	Relationship between material and data	6
Figure 10	Prediction power and accuracy	7
Figure 11	Web analytics and A/B testing [17] [18]	8
Figure 12	Diagram showing proposed thesis bringing data analytics into physical environments and potential applications.	9
Figure 13	Diagram showing the manifesto for the Internet of Materials.	10
Figure 14	Overall research map. Focus is on the central spine.	12
Figure 15	Comparisons of captured and synthesized deformations for a heterogeneous non-linear pillow (left) and trinocular stereo vision system consisting of three high-resolution cameras (indicated in red) and two to three light sources (indicated in green). The cameras are arranged in a triangular setup, which helps maximize visibility during capture of a contact interaction. The light sources ensure uniform illumination during the acquisition. (right) [5]	13
Figure 16	Web application showing resistance reading of a stretched conductive rubber thread (left) and silicon beam with four embedded sensors along its length (middle and right).	14
Figure 17	Beam section showing arrangement of embedded sensors.	15
Figure 18	Sensor readings at different bending directions.	16
Figure 19	Principal component analysis of bending classes shown in different colors.	17
Figure 20	First iteration of material training apparatus.	17
Figure 21	Sensor readings at different displacement distances recorded from the material trainer.	18
Figure 22	Digital work flow showing different stages.	19

Figure 23	Stress tensors per element per load case.	22
Figure 24	Stress tensors averaged out across load cases.	23
Figure 25	Calculation of eigenvalue/eigenvector pair from the stress tensor. Values shaded in grey are not used in 2.5D problems.	23
Figure 26	(a) through (c) represent three different load cases on a 2.5D beam and the corresponding principal stresses. (d) shows the beam in an undeformed state overlaid with averaged principal stresses across all three load cases.	24
Figure 27	Diagram illustrating principal stresses selection criteria.	25
Figure 28	(a) shows a plot of all averaged eigen1 principal stresses. (b) shows the filtered eigen1 stresses - the maximum 10 values and the minimum 4 values. (c) shows a plot of all averaged eigen2 principal stresses. (d) shows the filtered eigen2 stresses - the maximum 10 values and the minimum 4 values. (e) is an overlay of both eigen1 and eigen2 filtered stresses.	25
Figure 29	Diagram showing the two methods used to evaluate the standard deviation. The first operates on the stress tensor components and the second operates on the eigen values.	26
Figure 30	Diagram showing 10 elements with the highest standard deviations based on a scalar calculated from the stress tensor components	26
Figure 31	Diagram showing 10 principal stresses with the highest standard deviations based on eigen1 (a) and eigen2 (b)	27
Figure 32	3d printed mold (top) and rubber beam specimen with embedded sensors (bottom).	28
Figure 33	Local coordinate system and locations of displacements (top) and rubber beam specimen with surface-mounted sensors (bottom).	28
Figure 34	Second generation material training apparatus.	29
Figure 35	Material training dashboards. Machine control (left) and sensor monitoring (right).	30
Figure 36	Templates detected successfully within images acquired during testing.	31
Figure 37	Star template used (left) and noise generated during template detection within images (right).	31
Figure 38	Computer generated random patterns used for image analysis with subset. [23]	32
Figure 39	Sensor readings at 1 and 2 cm displacements.	33
Figure 40	Noise and prediction accuracy.	34
Figure 41	Pressure sensor readings.	34
Figure 42	Common examples of the three machine learning components [6]	35
Figure 43	Results from a multivariate linear regression model.	36
Figure 44	Results from a multivariate linear regression model using the logarithm of both dependent and independent variables.	37

Figure 45	Results from five models created with each single sensor readings as a single predictor.	38
Figure 46	Example illustrating the regression model accuracy.	38
Figure 47	Predicting displacement field from lookup table	39
Figure 48	Diagram illustrating the system architecture.	40
Figure 49	Real-time visualization of beam deformations on a web browser.	41
Figure 50	Readings from 3 sensors at rest across a period of 7 days.	42
Figure 51	Readings from 5 embedded sensors at rest across a period of 7 days.	43
Figure 52	1500 predicted displacement coordinates (top) and cleaned version eliminating predictions outside beam boundary (bottom).	43
Figure 53	Object in use at red markers and object at rest at blue markers.	44
Figure 54	Lengthening by interaction density.	44
Figure 55	Design for durability (harden by frequency of contact).	45
Figure 56	Design for comfort (soften by amplitude of contact).	46
Figure 57	VonMises stress distribution from single loadcase (top) and average VonMises stress distribution from composite loadcases (bottom).	46
Figure 58	Thickening by Stress.	47
Figure 59	A plot of VonMises stress values across time for a single element.	48
Figure 60	Fatigue prediction parameters of a single element.	49
Figure 61	Fatigue prediction parameters across entire beam (left) and combination of weighted parameters (right).	50
Figure 62	Hypothetical family of sensor-embedded ducks and potential suggested designs.	51
Figure 63	Early prototype of "Quantimetric Self-Sensing" apparatus, 1996 (body sensing apparatus with Digital Eye Glass for realtime display of ECG, EEG, EVG, and other body sensing apparatus output [9])	52
Figure 64	Wired magazine cover, issue 16.07 [12].	54

INTRODUCTION

The tremendous amount of data produced daily has accelerated our understanding of human behavior and trends. It has also motivated queries into means of transforming data into knowledge [Figure 1](#). However, on a data synthesis note, what constitutes data? Live metrics, customer surveys, qualitative or quantitative? Regardless of the format, this research explores data from a design perspective. The UX/UI field, being mainly software driven, has been able to benefit greatly from direct streaming of data to inform design decisions. It is the other non-software based design fields such as architecture and industrial design that are slow, and sometimes resistant, to adopting hard data.

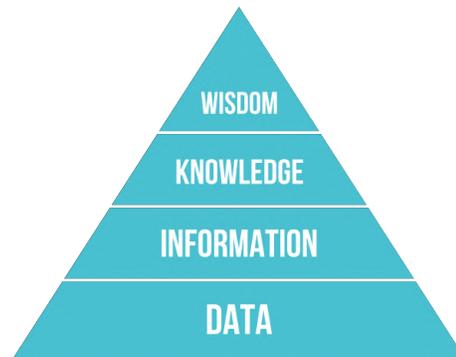


Figure 1: DIKW pyramid

As the title "Material Analytics" suggests, the second half of this discussion will revolve around materials. After all, material matter makes up any non-software based designs. This research examines the numerical analysis performed on materials, but beyond computer processors. While FEA examines materials within virtual environments, embedded sensors can help do just that on a physical level. Therefore the approach of decomposing the material digital domain into finite elements will be explored on physical prototypes. A recent interest in large deformations accompanied the emergence of fields such as soft robotics. This research's material focus is geared towards non-linear systems as new novel applications have emerged lately out of controlled non-linearity such as regulated buckling. Additionally, linear systems have been exhaustively researched. How do we bring physical objects online? The rise of the Internet of Things has been able to give hints in answering this question. There has been a great deal of work on developing sensors that capture motion, light, sound as well as environmental factors. Can this extend to capturing material behavior? The challenge lies in penetrating into a solid, analyzing its behavior and reporting back without compromising the solid's function. Therefore the sensing of material objects can be classified into two categories: one that senses an outer surface and does not go beyond it - an example of that would be 3d-laser scanning that converts surfaces into point clouds. Another type comprises penetrating an object without cutting it open - is what X-ray Computer Tomography scans do. The first cat-

egory only gives us a superficial understanding of material behavior and the second is yet to be made human cell-friendly. The one thing both have in common is the complete dependence on the sensor to perform while the "sensed" objects remain passive. Can we somehow embed sensory elements within the material itself and program these to do 50% if not 100% of sensing?

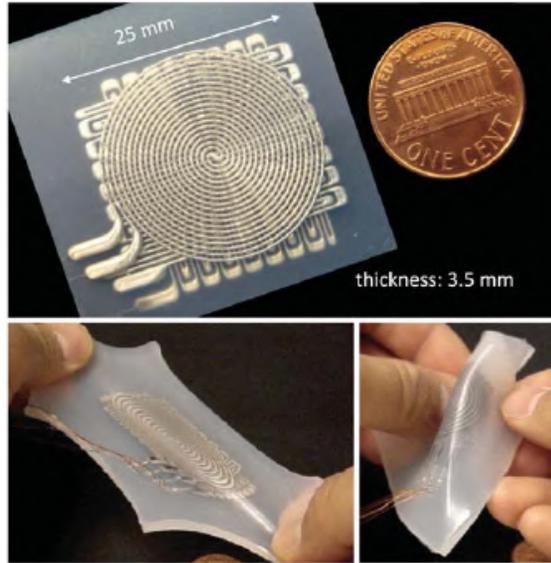


Figure 2: A Pressure and Strain Sensor Fabricated on Soft Artificial Skin by Yong-Lae Park et al.

Additive manufacturing through 3d printing comes to mind in trying to answer this question. As this technology has moved from being solely a rapid prototyping tool to a fully functional manufacturing option, advances in multi-material printing now provide means to print electronic components and sensors into objects within one single build process [Figure 2](#). These electronic components are conductive and through changes in capacitance, metrics such as mechanical strains are quantified. Piezoresistive components that exhibit changes in resistance due to applied mechanical forces is also an active area of research.

In an attempt to define a conceptual scaffold for Material Analytics, a number of related case studies are explored.

Amiens Cathedral: The intuitive post-completion monitoring of structures is evident in the development of the Gothic flying buttress design. This case study is concerned with the Amiens cathedral, also known as the Cathedral Basilica of Our Lady of Amiens, completed around 1270. It is located 120 km north of Paris. [\[13\]](#) During the construction of the cathedral, an openwork flying buttress scheme was introduced to replace a more conventional solid flyer system. The adaptation of the openwork system introduced several structural problems. It allowed exclusively large deformations of the upper clerestory walls. [Figure 3](#)

The unforgiving behavior of the mullions linking the upper and lower chords of the flyer was also problematic as they tend to pull out of their sockets. But more evidently,

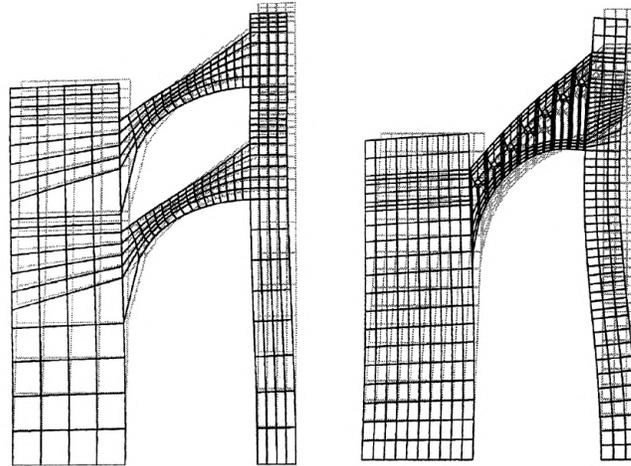


Figure 3: Exaggerated deformation patterns of both LUSAS models. Solid and openwork flyer system with outward (left-pointing) wind load added [22]

the overly high placement of the lower arc in openwork flyers was the major cause of cracks some 200 years later. Around 1500, written sources prove that the flyers were rebuilt. Additional struts [Figure 4](#) were inserted below the rebuilt flyers, demonstrating that the openwork flyers were indeed placed too high. [22]

As a result of these happenings, the form of the openwork flying buttress was abandoned. Amiens was the last major building to employ this specific iteration of flying buttresses.

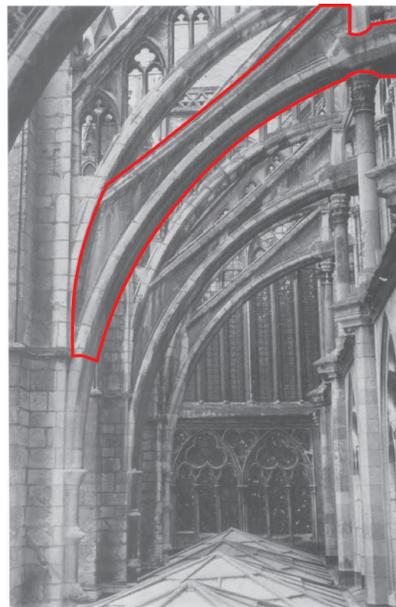


Figure 4: Amiens cathedral, choir flyers with additional struts [22]

Hence, it is evident that the trial and error nature of design iterations in this case (although preformed on the actual finished building), has greatly contributed to a body of design knowledge dedicated to informing and shaping future iterations. Instances

of flying buttresses that followed Amiens had sidestepped the structural problems of the openwork flyer entirely. For instance, the Cologne cathedral employs a lacy rows of openwork rosettes placed on top of the solid flyers. [Figure 5 \[22\]](#)

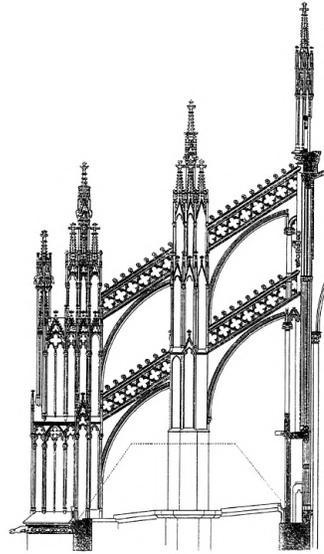


Figure 5: Cologne cathedra, choir flyers [\[22\]](#)

It wasn't until the early 19th century that we developed our understanding of stress-strain relationships. Strain gauges employing mechanical means of magnification appear to have had their origin in England. In 1802, Dr. Thomas Young of London, England proposed the ratio of stress to strain below the proportional limit as a property of the material. Thomas Tredgold, an English experimentalist and a contemporary of Young, attempted to evaluate Young's modulus for several materials using stress and strain data computed from load deflection tests. Certain assumptions on which his computations were made were attacked, and the experimental work, which these attacks caused Tredgold to undertake, led to the beginning of strain gauge development. [\[1\] Figure 6](#)

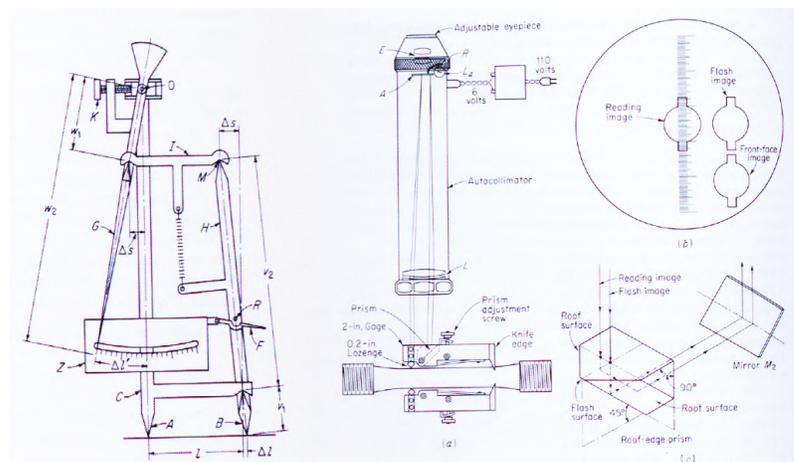


Figure 6: Various mechanical and optical strain gauges [\[2\]](#)

Contemporary technology has accelerated our understanding of such structural problems. We are no longer to wait 200 years for rectifying structures thanks to structural health monitoring.

The restoration work of Duomo di Milano cathedral main spire used such as system [Figure 7](#). A weight of 90 tons due to scaffoldings was applied to the dome and a complex monitoring system was designed in order to measure the deformation of the church and its spire. The system employed optical leveling, robotic total stations, accelerometers, strain gauges, optical plumbs, clinometres, and extensometres. In particular for the analysis of structural elements at higher risk of collapse, fiber optic sensors were selected, based on FBG (Fiber Bragg Grating) technology. [3] The authors also considered changes in thermal conditions and their effect on the structural readings. An alarm system was set to go once vibrations reach a certain threshold prompting the halting of all restoration works.

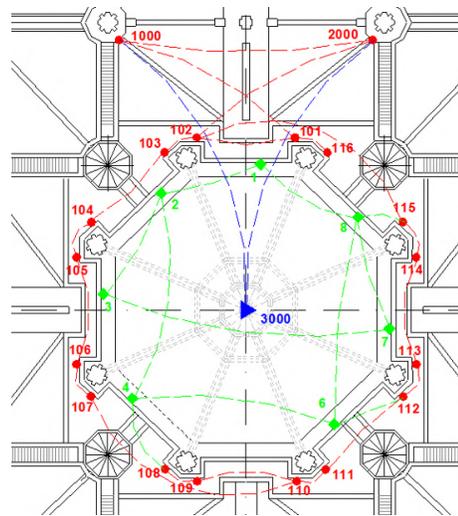


Figure 7: Layout of the geometric levelling network at the base of the dome. Green and red lines represent the internal and external networks, respectively. The blue point (3000) is hung on a cable that is connected to the top of the spire. [3]

Other modes of shape deformation have also proven useful in the area of tangible user interfaces. The FuwaFuwa sensor module [Figure 8](#) is a round, handsize, wireless device for measuring the shape deformations of soft objects such as cushions and plush toys. It can be embedded in typical soft objects in the household without complex installation procedures and without spoiling the softness of the object because it requires no physical connection. Six LEDs in the module emit IR light in six orthogonal directions, and six corresponding photosensors measure the reflected light energy. One can easily convert almost any soft object into a touch-input device that can detect both touch position and surface displacement by embedding multiple FuwaFuwa sensor modules in the object. [24]

From these precedents we are able to map out the relationship between material performance and data [Figure 9](#). In one end of the spectrum, we find direct practical applications such as structural health monitoring. These are monitoring applications that function to provide alerts at times of excess loading eg. too much vibrations on a bridge - stop cars



Figure 8: Directionality of reflected light is lower at different densities (top) and media controller (bottom) [24]

from crossing. They are also one-off meaning that they are design for a specific bridge, building or aircraft. Additionally, they are non-scalable and non transferable across platforms and applications. The other spectrum is research oriented and is very heavy on data. These are on a molecular level during material discovery and development as in the case with material informatics. They could also be on a design level through structural simulation.

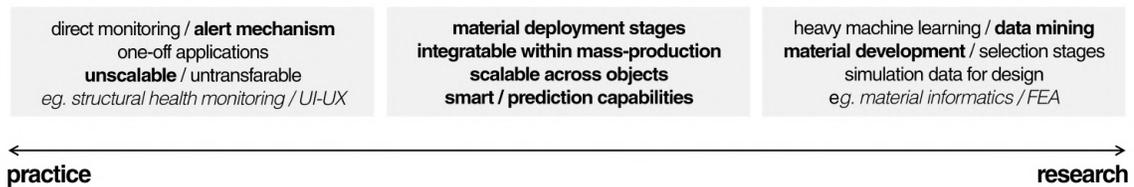


Figure 9: Relationship between material and data

The area of interest lies somewhere in between these two extremes: A domain that is developed for mass-produced objects, scalable across different types of objects and to be utilized while objects are in use. More importantly, the domain should have some sort of prediction power. [Figure 10](#) The importance of such power comes from the need to produce viable solutions. Let's assume that finite element analysis, with its discretization approach, yields a relatively high accuracy in simulating material behavior. If we were to map this dense discretization into degrees of strain measurements in physical objects, this would not be viable and might even impede the functionality of these objects. But if we are able to reduce these degrees of strain measurements to the absolute minimum and couple that with a prediction power, and get good accuracy, then the methodology is valid.

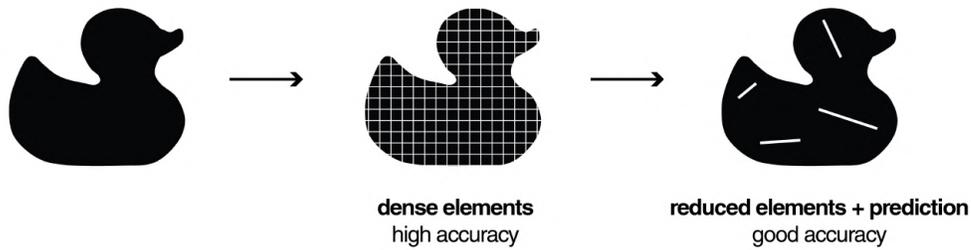


Figure 10: Prediction power and accuracy

Through data aggregation over time as these objects are in use, we would be able to extract mechanical knowledge from the data. These ideas have been motivated by how other software-based fields that have been able to leverage data feedback into the design process. Web design [Figure 11](#) has had several metrics, such as bounce rate and conversion rate, developed specifically for the purpose of quantifying the performance of their web layouts. These metrics are collectively known as KPI's or key performance indicators. One example of these web analytics practices is A/B testing. A/B testing (sometimes called split testing) is comparing two versions of a web page to see which one performs better. You compare two web pages by showing the two variants to similar visitors at the same time. The variant that gives a better conversion rate is deemed to have outperformed the others. [17] These numbers allow designers to move from a "we think" attitude to a "we know" attitude and eventually enables data-backed decisions. The question here that poses itself is then how do we utilize these practices into the physical environments, on an industrial scale to start with?

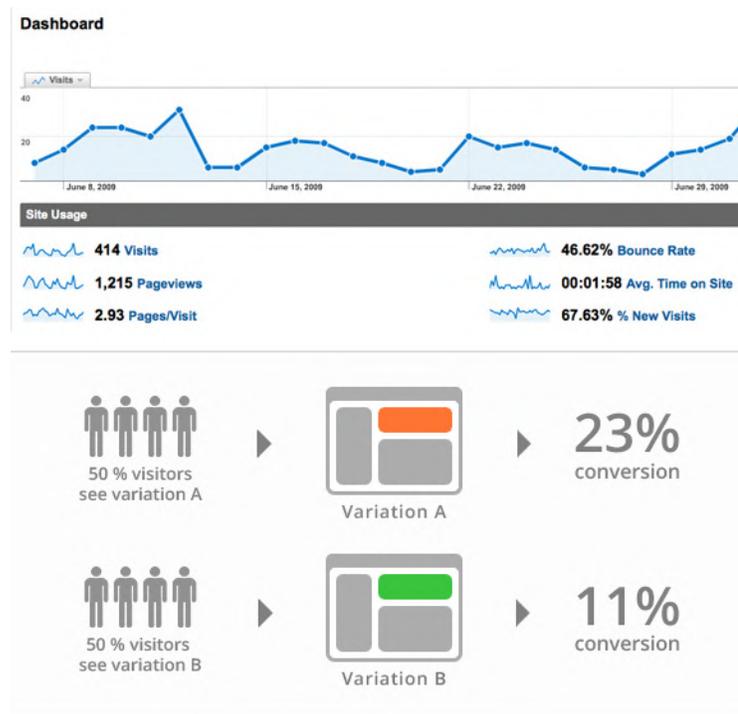


Figure 11: Web analytics and A/B testing [17] [18]

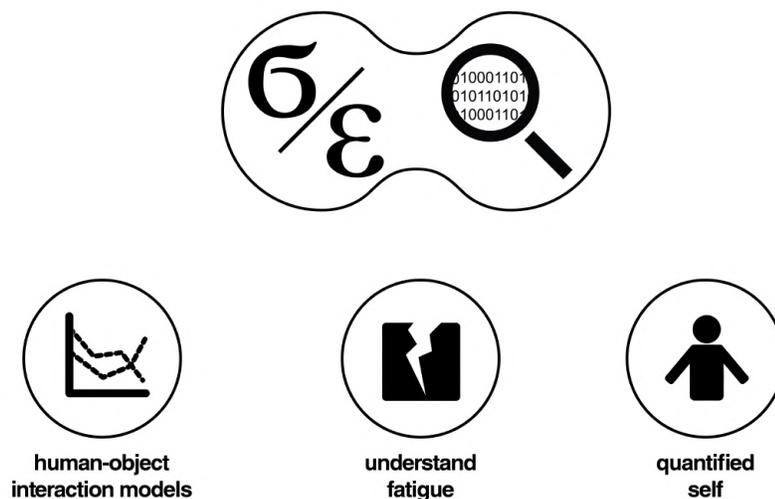


Figure 12: Diagram showing proposed thesis bringing data analytics into physical environments and potential applications.

The thesis for this research can hence be outlined as :

"An opportunity driven research that aims to drive a data analytics framework into physical environments using both numerical and experimental means." [Figure 12](#)

It is through the aggregation of mechanical behaviour data over a period of time that one can start exploring:

- **Human-Object Interaction Models:** These models could be used to validate or simply attempt to gauge the performance of new designs. They could predict human interaction patterns and tendencies with such designs. A human-object interaction can be seen as an extension of human-factors and ergonomics (HF & E) as well as human-computer interaction (HCI). It differs from both these concepts that it is object-focused rather than being human-focused. This includes using human input as variables for design output.
- **Understanding Fatigue :** Fatigue is the weakening of a material caused by repeatedly applied loads. It is the progressive and localized structural damage that occurs when a material is subjected to cyclic loading. [16] It remains one of many load cases that are extremely challenging to model and predict as they are heavily reliant on the time factor. Additionally, it is nearly impossible to detect any progressive changes in material behavior during the fatigue process, so failures often occur without warning.[15] Such aggregated data will go beyond simply providing alerts

to potential failures. It will accelerate our understanding of fatigue and perhaps help us build more accurate predictive models.

- **Quantified-Self:** With the emergence of wearables, smart watches and other devices that track our daily activities, notions of self tracking have quickly gained wide interest especially with the tremendous amount of data that has been generated as a result. An indirect method of tracking ourselves could then become through tracking objects that we deal with. Hence, we are able to understand ourselves better through how we interact with objects around us.

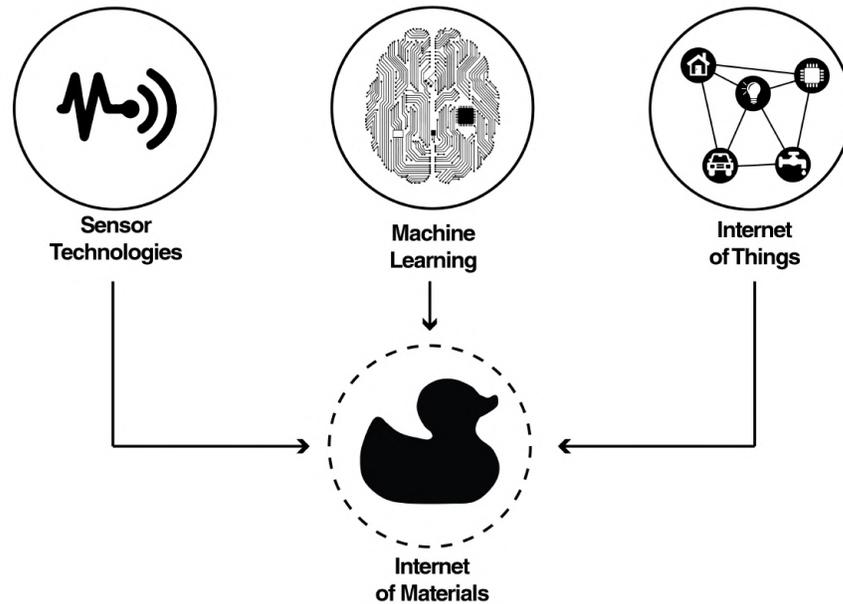


Figure 13: Diagram showing the manifesto for the Internet of Materials.

The research also aims to build up a framework outlining the **Internet of Materials** or **IoM**. Three actively researched areas could be explored simultaneously and collectively bundled to build this framework: [Figure 13](#)

- **Sensor Technologies:** From electrical strain gauges to fiber optics sensors, sensor technologies have been widely developed to cater for a variety of needs. Costs have dropped significantly to an extent that DIY hackers can purchase relatively sophisticated sensors at reasonable prices. However, these sensors still remain exclusive to certain applications and objects. The IoM ensures that sensor adaptation expands to reach a broader range of everyday objects.
- **Machine Learning:** Whether it be supervised or non-supervised learning, from support vector machines to neural networks, machine learning is simply based on computers' ability to learn from data and make predictions. With the emergence of data scientists as a hit profession in the past five years, machine learning has become at the heart of almost every new digital application. The IoM will utilize machine learning to reduce measured quantities while maintaining good prediction scores.

- **Internet of Things:** This relatively new term implies connecting object to the internet through embedded electronics. Now, a sensor embedded into your milk carton will detect that you are running out of milk and will place an order on your behalf on amazon.com. It is expected that by 2020, there will be 26 million devices connected to the Internet (Gartner, Inc). The IoM is to be seen as a natural, more specific extension of the internet of things. However, IoM is more concerned with analog vs digital signals. Additionally, it is focused on quantifying how objects feel rather than solely monitoring or controlling them.

As applications built on top of the IoM framework are yet to exist, there are numerous unknowns and areas to be explored. [Figure 14](#) shows a map of these potential routes, each representing an entire research project. One can explore the 3d printing of embedded sensors within objects in single builds. Another path would explore UI/UX strategies in defining how users interact with this new type of mechanical deformation data. One could also start exploring creating different API's (application programming interfaces) for retrieving and mining data pertaining to specific objects. The research path chosen to be pursued here is the absolute minimum prerequisite for all other potential paths mentioned above. This path involves three main steps: **A.**embedding sensors in objects in the appropriate locations, **B.**transmitting and storing such data and **C.**extracting useful insights from it.

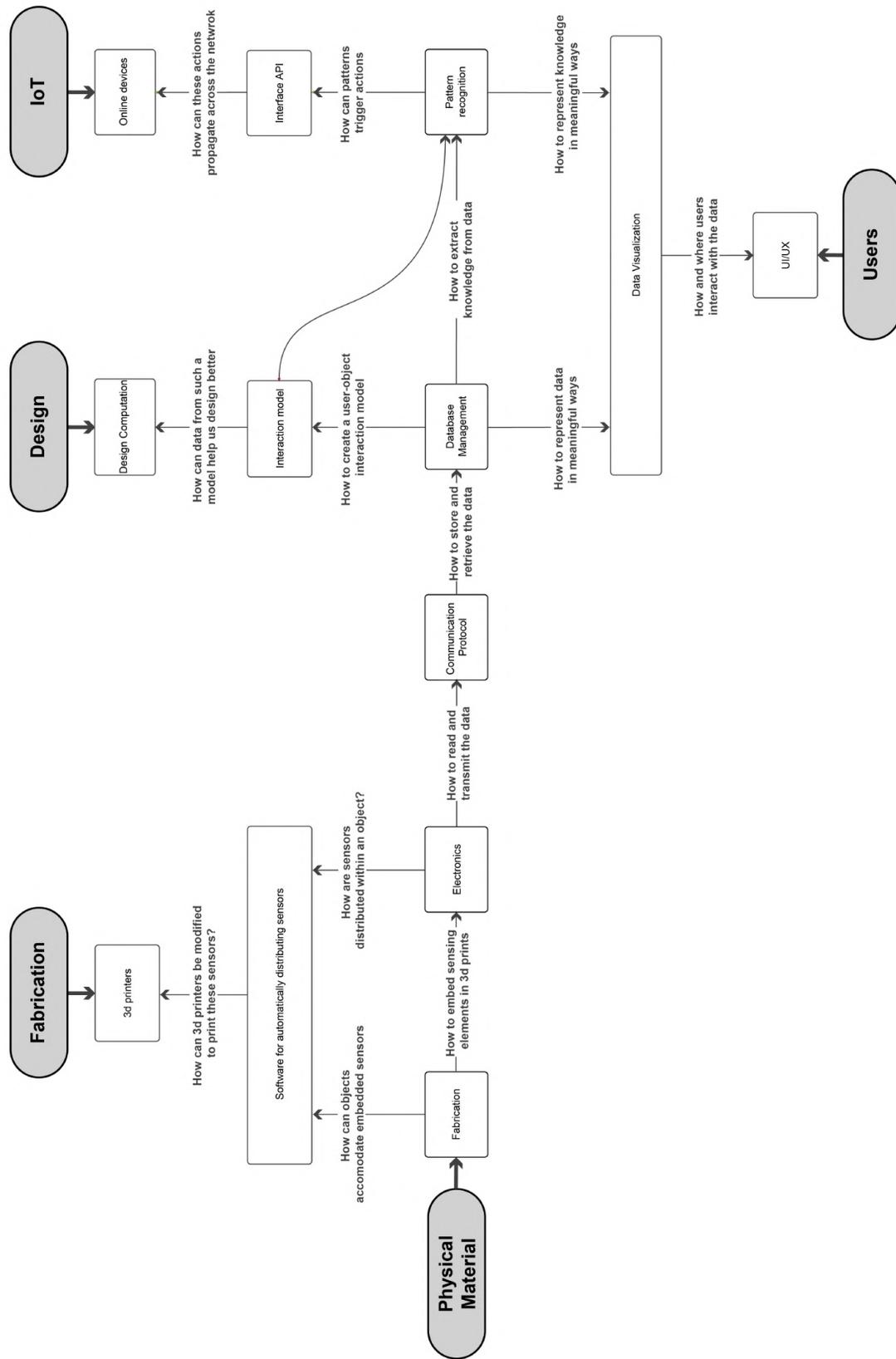


Figure 14: Overall research map. Focus is on the central spine.

INITIAL STUDIES

With an interest in large deformations, initial studies start to research possible means of measuring and tracking deformation. Existing tracking methods are vision-based, relying mainly on either depth or color analysis or both. Through image data, combining depth sensors with color video cameras allows for capturing 3d motion - similar to Kinect's underlying principal. Digital Image Correlation (DIC) is also used where changes in images are tracked and measured.

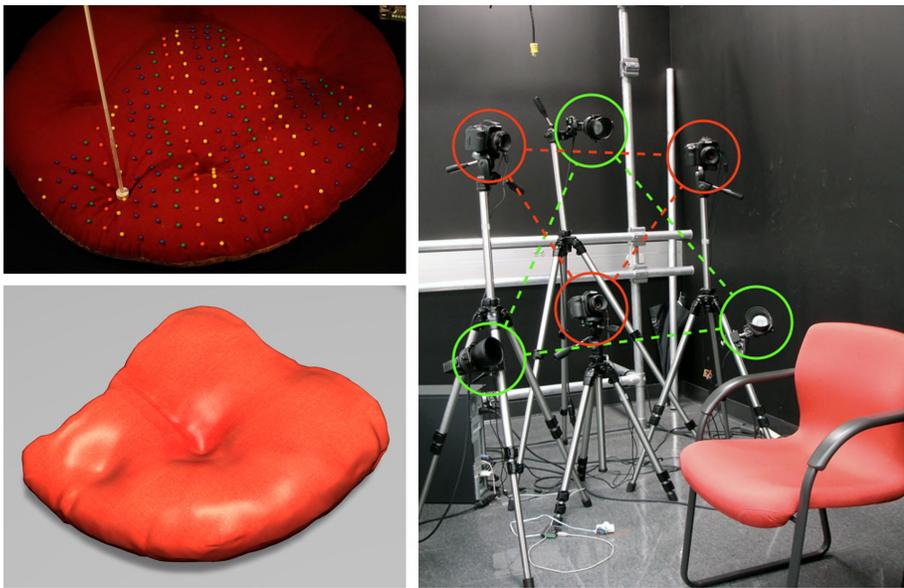


Figure 15: Comparisons of captured and synthesized deformations for a heterogeneous non-linear pillow (left) and trinocular stereo vision system consisting of three high-resolution cameras (indicated in red) and two to three light sources (indicated in green). The cameras are arranged in a triangular setup, which helps maximize visibility during capture of a contact interaction. The light sources ensure uniform illumination during the acquisition. (right) [5]

An example of computer vision tracking is that developed by Bickel et al. [Figure 15](#) for the capture and modeling of non-Linear heterogeneous soft tissue. The method captures deformation examples by combining a marker-based trinocular stereo-vision acquisition system and force sensors. Deformations are induced by physical interaction with the object through a probe. These probes have arbitrary shapes and circular disks of different diameters. A marker-based system is used to due to its simplicity, robustness, and independence of the object's surface properties. [5] While giving very accurate deformation mapping results, these stereo vision approaches are external to the sensed objects. They are susceptible to light and shadow, require the addition of markers to surfaces, and are more appropriate to one-time rather than continuous deformation measurement. They are also confined to tracking surface behavior rather than an overall registration of

sub-surface deformations. This global simplification is beneficial in computer animation applications where the deformation richness and complexity is retained with a relatively low computational cost. Hence, video games, virtual surgery and films amongst other applications rely on such methods. Most importantly, they have limitation when scaling into 3d and would require additional tracking cameras to capture as many view angles as possible. There are also other sensor selection criteria, namely Signal I/O. Sensors need have a suitable interface rate, i.e. allow for measuring quantities at small enough intervals. They should also allow for one-way wireless communication.

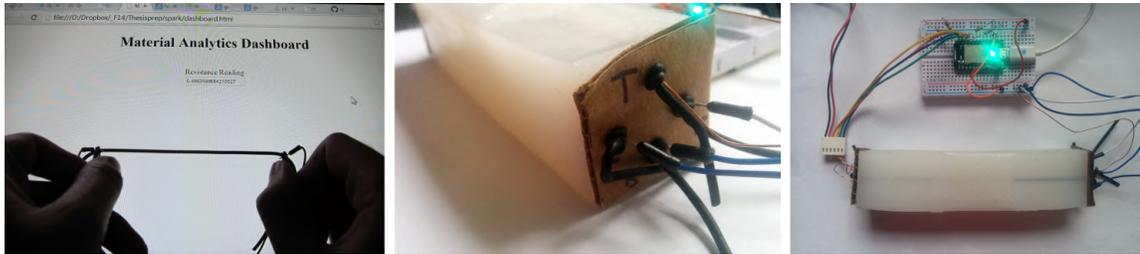


Figure 16: Web application showing resistance reading of a stretched conductive rubber thread (left) and silicon beam with four embedded sensors along its length (middle and right).

The most important criterion perhaps is the ability to embed the sensors within objects so that measurements are from within. Electrical strain gauges were explored. These gauges are usually made out of metal. They exhibit a change in electrical resistance when being stretched. This resistance change could then be mapped to a strain value. However, these metal gauges are not flexible enough to measure large deformations. Extensions beyond an unforgiving threshold will cause the metal sensor to fail. Hence, conductive rubber threads were explored. Stretching these carbon-impregnated rubber threads will move carbon compounds away and increase resistance. These threads are coupled with a microprocessor, a "Spark Core" that is designed as a microprocessor for the cloud. The quarter coin-sized electronic component sends a voltage through the thread and measures the outgoing voltage hence measuring the resistance. These values are sent to the cloud and could be easily retrieved through an HTTP request. This falls within the REST API protocol. A prototype web application was developed to visualize resistance readings [Figure 16](#). Additionally, four threads were embedded into a 3d silicon beam. [Figure 17](#) shows the configuration of said threads within the section of the beam. The threads are pre-stressed enabling them to measure both tension and compression; An increase in resistance translates to tension and a decrease translates to compression. They are arranged in a diamond shape allowing for measuring bending in all four directions. For instance, an upward bending would cause the blue sensor to loose tension while the orange sensor would undergo larger tension. The red and green sensors might not exhibit much of a change then.

The aim of this exercise is to prove that these bending classes would produce unique sensor reading patterns. [Figure 18](#) illustrates that the readings are infact unique. Given a new set of readings, one could easily predict the direction of bending without visually monitoring the beam specimen. At this level of simplicity, the prediction can even be done without sophisticated machine learning algorithms.

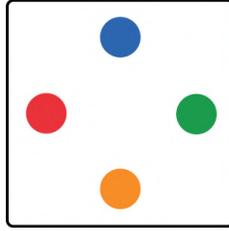


Figure 17: Beam section showing arrangement of embedded sensors.

15 readings were taken at each bending direction i.e. while the beam is under static loading. These 15 readings differ even though the beam was static. This is due to noise and imperfections as a function of the sensors used. Principal component analysis or PCA was used to further analyze the data. PCA helps reduce a complex data set to a lower dimension. This helps in revealing the sometimes hidden structures that underlie the data. [Figure 19](#) shows a plot of the first vs the second principal components of the data. Each bending class is represented in a different color. Here we see that, even with noise artifacts, the classes are very well clustered i.e. the colors are nicely packed into discrete regions. We notice that noise is highly reduced within the unloaded bending class (in red). This is demonstrated by the 15 red data points falling almost on top of one other. The leftwards bending class (in purple) shows the most deviation and scattering. If we were to introduce a new data point and by using a simple machine learning algorithm, such as K-nearest neighbours or KNN, we would easily be able to identify the class that this new point belongs to.

However, knowledge of the direction of bending is not enough. Another important metric to track is the displacement distance. That is, how much the specimen is bending rather than in which direction it is bending. This led to the development of a first iteration of a material training apparatus [Figure 20](#). This apparatus is not to be seen as a material testing one, it is a non-destructive training tool used to generate sensor training data for subsequent machine learning. It features a lead screw that positions a probe at exact z-axis dimensions simulating specific displacements on the specimen. The probe is thus positioned at some location, readings from all four sensors are noted down then the probe moves to the next training displacement value. Currently, this lead screw is unable to move along the x-axis. Further iterations of the apparatus should allow that.

[Figure 21](#) shows a plot of the acquired sensor readings at 2.0 mm intervals from 0.0 mm (unloaded) to 26.0 mm (fully loaded). We notice that readings from sensors L and R remain almost flat throughout the experiment as expected. This implies that they will have little significance when it comes to predicting the displacement value. T and B, on the other hand, show relatively good variation. As long as one single value shows variation, we can predict the displacement given the sensor readings. A linear regression model would allow interpolating between these values and hence predicting displacements that were not explicitly part of the training dataset.

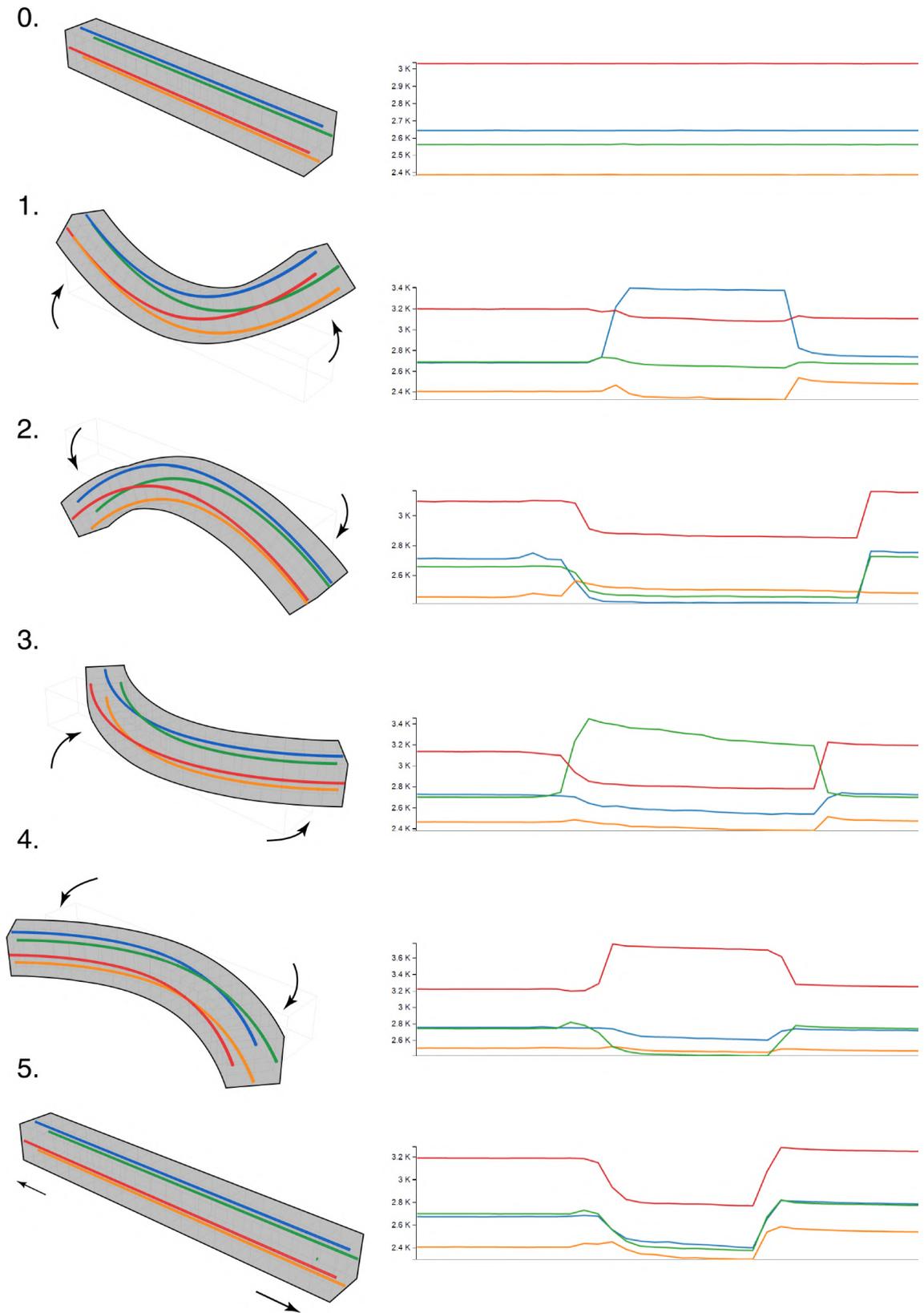


Figure 18: Sensor readings at different bending directions.

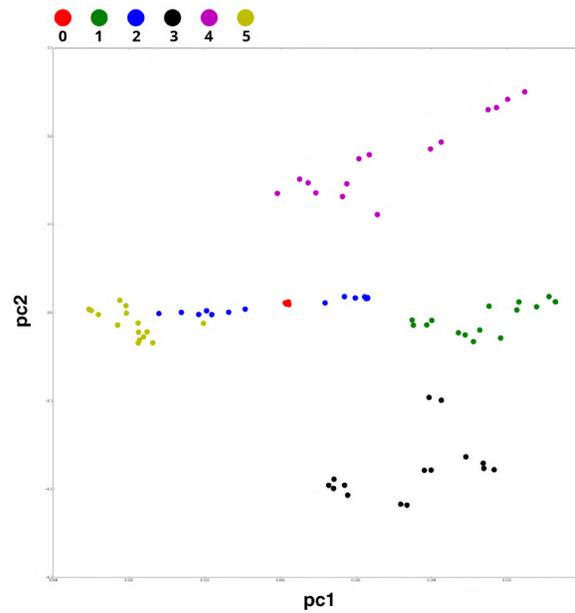


Figure 19: Principal component analysis of bending classes shown in different colors.

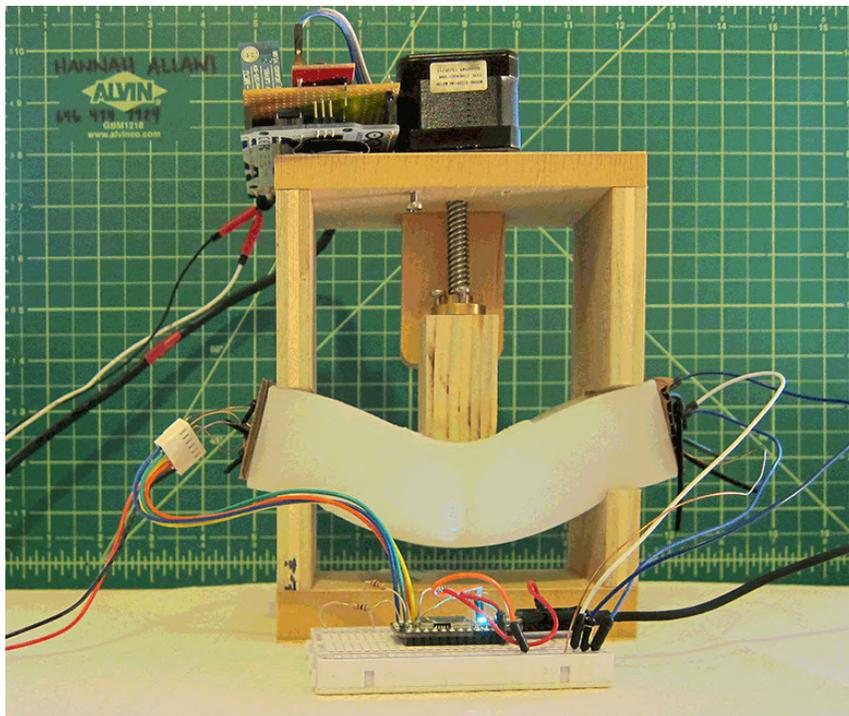


Figure 20: First iteration of material training apparatus.

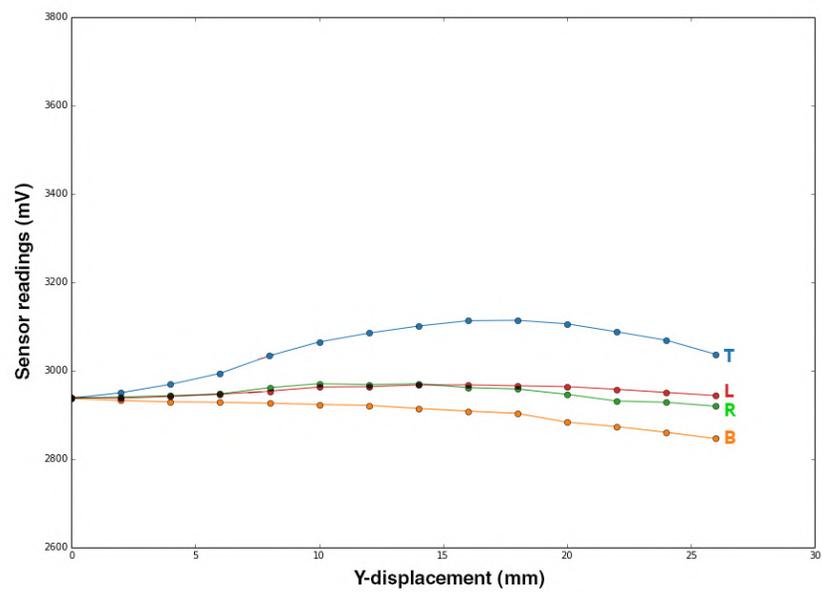


Figure 21: Sensor readings at different displacement distances recorded from the material trainer.

DIGITAL WORK FLOW

This chapter outlines both numerical and experimental work flows [Figure 22](#) developed simultaneously to implement and train sensor-embedded objects. This starts from load case definition within digital simulation environments to identify most optimum sensor placement locations. A physical experiment is then set up to generate sensor training data. This is followed by a machine learning exercise that uses training data to build a prediction model. Finally, the system architecture is explained illustrating how data is transmitted, saved and eventually mined to extract useful insights. A 2.5D rubber beam with side supports will be used to showcase the work flow.

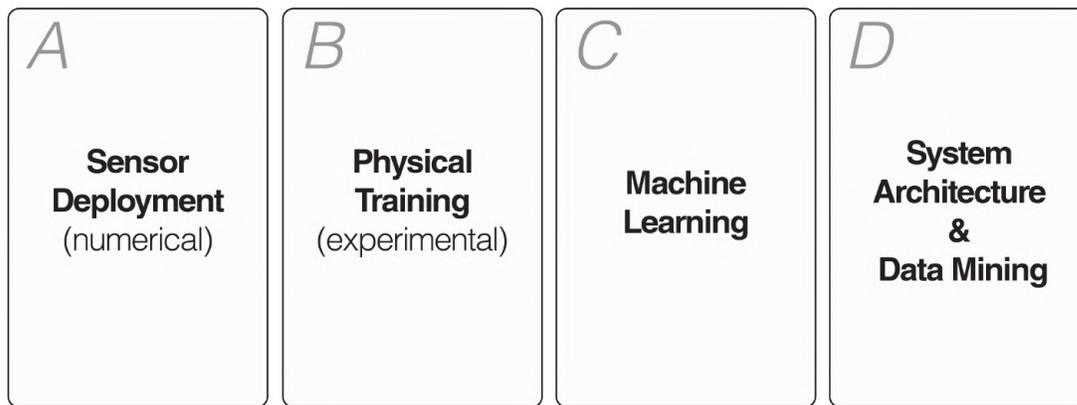


Figure 22: Digital work flow showing different stages.

4.1 SENSOR DEPLOYMENT

The work flow starts with specifying load cases the specimen is expected to be subjected to. These are typically theoretical, based on the designer's intuition or simply based on the relative position of the specimen within the larger assembly.

This section is thus concerned with identifying specific locations within objects that would be most adequate for sensor placement. These are locations that would exhibit **a.** relatively high stresses and **b.** relatively large variation in stress values across all proposed load cases. Other considerations might deal with the practicality of placing sensors in locations that would impede the functionality of the object. The later will not be addressed in this exemplary work flow. Future applications could accommodate such considerations and allow culling out regions that are not good candidates for sensor placement due to clashes with functional portions of the object.

Within the structural health monitoring field, economic factors often dictate the number of sensors used. Generally, the more sensors are placed, the more information could

be obtained. The number of sensors is however strictly constrained by cost [8]. It therefore became crucial to answer the following question: What would be the minimum number of sensors able to give the best description of forces/displacements acting on the object?

4.1.1 FEA simulation routine

Abaqus, a software suite for finite element analysis capable of solving highly non-linear systems is utilized here due to the hyper-elastic nature of the rubber specimen.

solid modeling The routine starts by the identification of two solid models, the first being of the specimen itself and a second representing a body, hereafter referred to as a pin, exerting force onto the specimen. These models are imported into Abaqus i.e. they are modeled externally (in meters, .igs format). They will be meshed within Abaqus and therefore mesh sensitivity issues will be addressed thereafter.

materials A considerable amount of literature has been published on the modeling of rubber materials. The choice of the model depends to the application, corresponding variables and the availability of data to determine material parameters.[4] The "Yeoh" strain energy model has been chosen to describe the hyper-elastic properties of rubber compounds as it is applicable to a much wider range of deformation. It is able to predict the stress-strain behavior in different deformation modes from data gained from one simple deformation mode (eg. uniaxial extension) [7]

The specimen material is thus defined as an isotropic hyper-elastic material based on the "Yeoh" model. While this currently acts as a theoretical model, future physical testing could help verify if this specific hyper-elastic model is the most suitable for modeling the specimen's material. The material properties are based on the following parameters:

$$E = 1.0e + 06 \text{ N/m}^2 \quad \nu = 0.499 \quad \rho = 1150 \text{ kg/m}^3 \quad (1)$$

The model coefficients C_{10} and D_1 are calculated from the initial shear and bulk modulus as follows:

$$\mu = 2 * C_{10} \quad \kappa = \frac{2}{D_1} \quad (2)$$

where

$$\mu = \frac{E}{2 * (1.0 + \nu)} \quad \kappa = \frac{E}{3.0 * (1.0 - 2.0 * \nu)} \quad (3)$$

The pin material as defined as an isotropic elastic material with the following parameters:

$$E = 2.0e + 07 \text{ N/m}^2 \quad \nu = 0.3 \quad (4)$$

sections A solid homogeneous section is applied to both the specimen and pin models without plane stress or plane strain considerations.

meshing and sets Both the specimen and pin are meshed. The specimen mesh is to match the superimposed mesh on future physical tests. Exported nodal displacements

from this simulation can thus be compatible for comparison with experimental counterparts. Specimen sets are defined as follows: all nodes, all elements and support faces. Pin sets are defined as follows: all nodes and all faces.

assembly and interactions Parts are merged and positioned appropriately in an assembly. Coinciding faces on both the specimen and pin are defined as surfaces. These surfaces are then used to model a surface-to-surface standard contact interaction model. The pin surface is assigned as the master surface while the specimen surface is assigned as a slave. The interaction property is defined with a "Normal" hard-contact behavior and a penalty "Tangential" behavior with a 0.2 friction coefficient. Surface smoothing is enabled.

constrains and motion Within this specific problem, the pin travels along the -z direction and exerts force on the specimen. A reference point is identified and tied to the pin as a rigid body constraint. Any translations or displacements that are applied to the reference point will be mirrored to the pin. This will be helpful in quickly setting up multiple load cases where the location of the pin varies.

analysis settings Non-linear analysis is set up within a static-general step right after the initial step. The step parameters are defined as follows: Maximum number of increments: 10000, Increment size: initial: 0.001, min: 1E-006 and max: 1.

boundary conditions Three main boundary conditions are applied:

- **Supports:** Within the initial step, the support regions on the specimen are encastred along all possible six degrees of freedom.
- **Displacement:** Within the static step, a displacement is imposed on the reference point causing the pin to travel and deform the specimen. It was decided to apply displacement distances rather than loads. This allows the routine to be duplicated in the experimental setting as accurately as possible; An exact force value will be difficult to replicate in the proposed physical setting. A pressure sensor could be used to record the force exerted on the specimen at each load case.
- **D0F along the Y:** As this problem is formulated as 2.5D for simplicity, all nodes within the entire model are given a displacement of 0 along the y-axis i.e. no out of plane movements are allowed.

4.1.2 Data extraction

Three main data types are extracted from each load case. These are:

- **Stress Tensor components:** $S_{11}, S_{22}, S_{33}, S_{12}, S_{13}$ and S_{23} components are extracted. These will be used to construct the stress tensor.
- **Integration point coordinates:** These X,Y,Z coordinates correspond to where the stress tensor has been calculated. Within this scope and for simplicity, a single integration point is used per element (at the center of the element).
- **Node coordinates:** These are X,Y,Z coordinates of the nodes at the last step of the non-linear analysis routine. These could potentially be used to compare results

taken from the physical experiment and hence gauge the accuracy of the numerical analysis.

4.1.3 Identifying optimum sensor location

By understanding the forces acting on the specimen under these multiple load cases, an educated guess can be made on where sensors could be deployed effectively. Locations of interest are those that exhibit large concentrations of stress i.e. where the specimen experiences large amounts of strains (where stresses are directly proportional to strains). Another criterion to identify these locations are those where magnitudes of specific stress tensors within the specimen experience the widest range of variation across all load cases i.e. they have a large standard deviation. Large variation in values allows load cases to be easily identifiable and hence will result in relatively accurate machine learning predictions.

Extracted data across all load cases is then processed simultaneously. A data frame is constructed, per load case, housing the stress tensors for all elements [Figure 23](#).

element	tensor
1	$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$
2	$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix}$

Figure 23: Stress tensors per element per load case.

Element-specific tensors are then averaged out across all load cases. This is done by adding the tensors and dividing them by the number of load cases [Figure 24](#).

In order to extract the principal stress magnitudes and directions (maximum and minimum), the eigenvalues and eigenvectors of the stress tensor need to be calculated respectively. The eigenvector x of a matrix A is represented by:

$$A * x = \lambda * x$$

where λ is the eigenvalue paired to the eigenvector x . (5)

Eigenvectors can't be added and therefore the averaging process was performed on the tensors themselves as oppose to the eigenvalue/vector pair. Three value/vector pairs were calculated. Since this exemplary test is formulated as a 2.5D problem, out-of-plane forces were constrained when setting the boundary conditions. Hence the out-of-plane

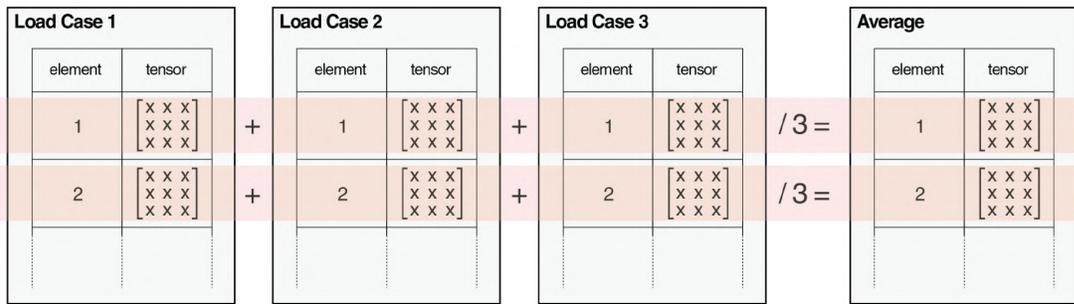


Figure 24: Stress tensors averaged out across load cases.

values of the first and second eigenvectors as well as the entire third eigenvalue/vector pair are not used [Figure 25](#).

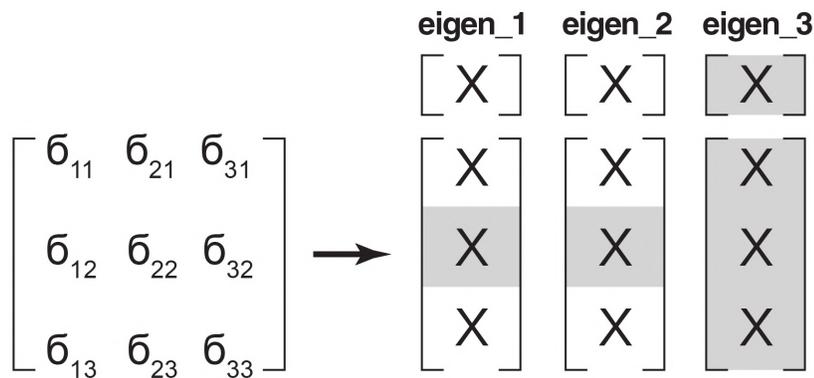


Figure 25: Calculation of eigenvalue/eigenvector pair from the stress tensor. Values shaded in grey are not used in 2.5D problems.

The following step involves the visualization of the averaged eigenvalue/vector pairs. The coordinates of the integration points extracted from the FEA routine are used as origin points per element. Principal stresses are drawn along the eigenvector directions with a magnitude proportional to the corresponding eigenvalues. [Figure 26](#) shows the averaged principal stresses extracted from three load cases. As expected, large stress concentrations exist around supports at either side. Additionally the top and bottom surfaces of the 2.5D beam are moderately stressed.

The filtering stage is the terminal and most critical stage in identifying the optimal sensor placement locations within the specimen. Two filtering techniques were explored. Principal stresses selected by both techniques were identified as most optimum [Figure 27](#). These are:

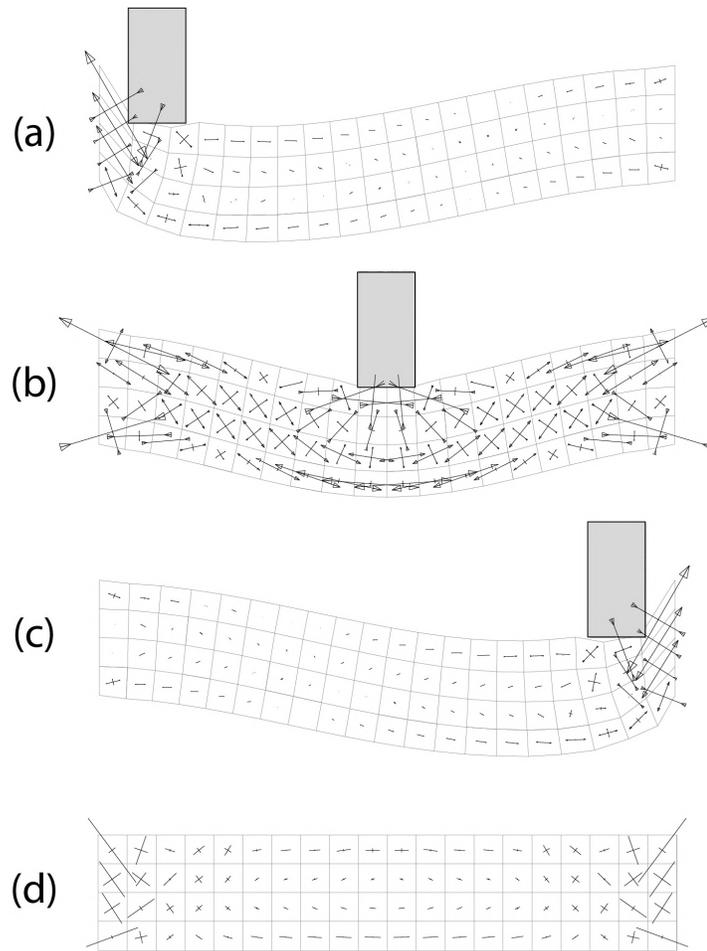


Figure 26: (a) through (c) represent three different load cases on a 2.5D beam and the corresponding principal stresses. (d) shows the beam in an undeformed state overlaid with averaged principal stresses across all three load cases.

- A. Filtering by averaged magnitude (eigenvalue): A threshold is defined whereby the maximum and minimum principal stress eigenvalues are extracted. Maximum positive eigenvalues correspond to maximum stretching within the specimen while maximum negative (or otherwise minimum values) correspond to areas of maximum compression. The conductive rubber sensors used to measure strain are more suitable to detecting tension than they are to compression. The ability to measure compression require pre-stressing the rubber and pre-calibrating the readings as per the undeformed state. Hence, a larger threshold is set for maximum positive values (tension) than that set for minimum negative values (compression). Since the specimen is symmetrical, even numbers are chosen for both thresholds. This process is repeated twice for both sets of eigenvalue/vector pair. [Figure 28](#) shows the averaged principal stress vectors together with the filtered maximum and minimum eigenvalues.

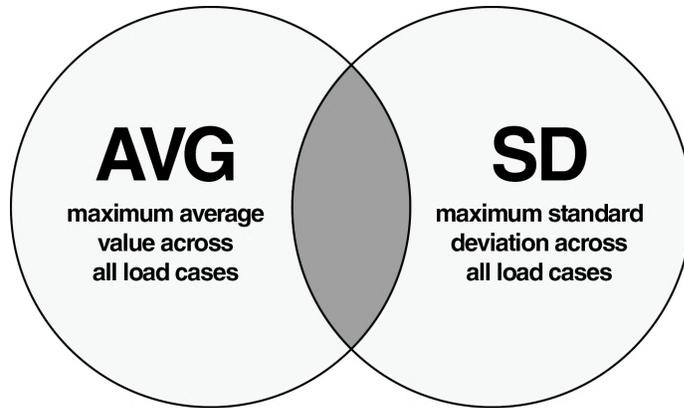


Figure 27: Diagram illustrating principal stresses selection criteria.

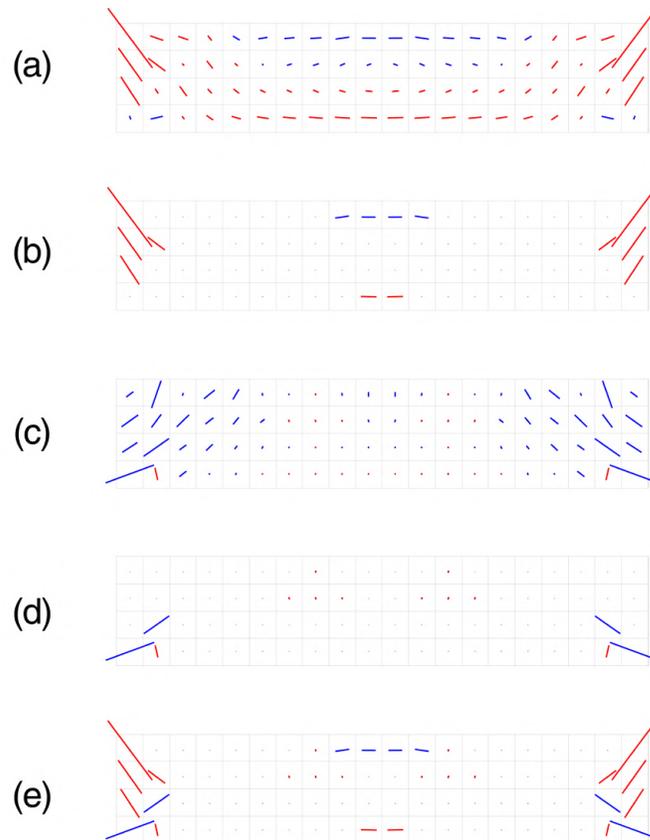


Figure 28: (a) shows a plot of all averaged eigen1 principal stresses. (b) shows the filtered eigen1 stresses - the maximum 10 values and the minimum 4 values. (c) shows a plot of all averaged eigen2 principal stresses. (d) shows the filtered eigen2 stresses - the maximum 10 values and the minimum 4 values. (e) is an overlay of both eigen1 and eigen2 filtered stresses.

- B. Filtering by standard deviation: The second filter is responsible for comparing the eigenvalues corresponding to each element across all load cases. Two methods were

explored [Figure 29](#). The first operates on the stress tensor components and the second operates on the eigenvalues themselves. The former ensures that inaccuracies due to the inevitable switching of eigenvalue/vector pairs are eliminated i.e. the first eigenvalue in element 1 / case 1 might end up being the second eigenvalue in element 1 / case 2 and so on. The first method is non-eigen specific and returns the standard deviation of the sum of in-plane stress components S_{11} , S_{21} , S_{13} and S_{23} across all load cases. Therefore the result can be interpreted element-wise rather than vector-wise. i.e. elements that have principal stresses displayed in [Figure 30](#) are those with highest standard deviation averaged in all directions. Thus their geometry undergoes the most deformation across all load cases. [Figure 31](#) shows the result from the second method. Notice the accumulation of principal stresses towards the supports as expected.

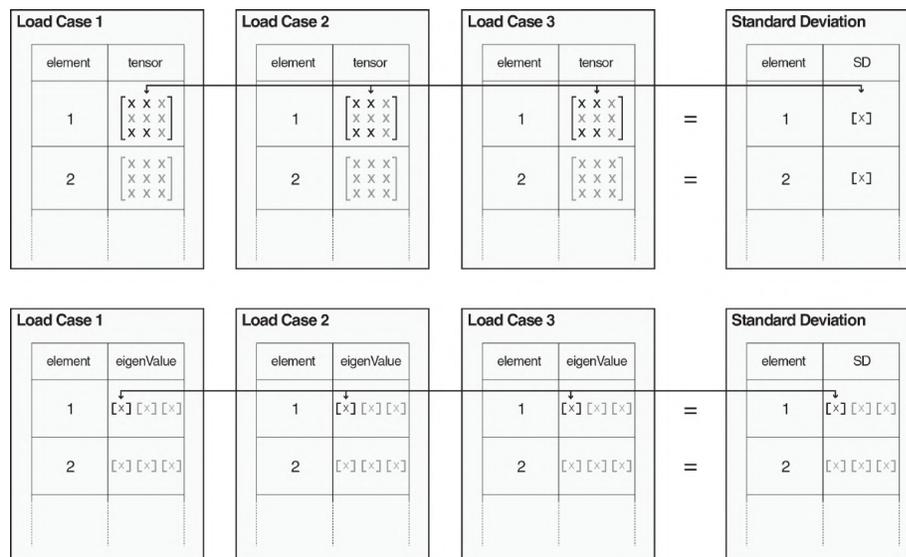


Figure 29: Diagram showing the two methods used to evaluate the standard deviation. The first operates on the stress tensor components and the second operates on the eigen values.

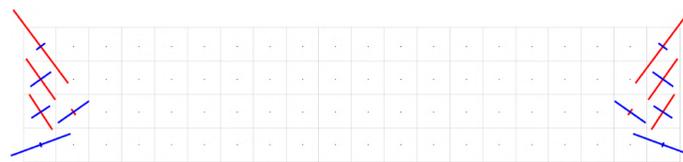


Figure 30: Diagram showing 10 elements with the highest standard deviations based on a scalar calculated from the stress tensor components

Ideally, some sort of clustering method would be employed to group the principal stresses and hence identify X number of "ideal" locations for sensor placement. This calculation would then take into consideration both highest averages and highest standard deviations. Additionally, areas could be identified as no-go-zones where sensors would impede the functionality of the object. Sensor type and size could also be set as a parameter in deciding on most appropriate locations.



Figure 31: Diagram showing 10 principal stresses with the highest standard deviations based on eigen1 (a) and eigen2 (b)

4.2 PHYSICAL TRAINING

The second stage of the digital workflow outlines the experimental portion of the research. This section is concerned with generating sensor data that will be used to train a machine learning algorithm. The numerical methods carried out in the previous section are thus replicated in an experimental setting, but with the addition of the sensor readings variable.

4.2.1 Preparation of physical specimen

The sensor deployment exercise has given a good indication of the most appropriate areas for sensor deployment. These were concentrated around the supports as well as in the mid-lower and mid-upper sections of the beam. Now, a physical prototype of the beam can be built incorporating these embedded sensors. An ABS plastic mold has been printed for casting the rubber beam [Figure 32](#). The mold introduces positives as cavities for embedding sensors into the beam. Metal clips are used to constrain the sensors at their ends. Additional positives are then introduced into the mold to allow installing the clips. Initial tests showed that the clips successfully constrained the sensors and prevented creep. However, the bulk of removed material around the sensors (spaces used to install the clips) has disengaged the sensor for the larger material bulk of the beam. That is, displacements applied to the beam had almost no effect on the overall length of the sensors.

In lieu of embedding the sensors within the beam, they were instead applied on the surface [Figure 33](#). Metal wire cross through the beam from one side to another and is used to constrain the sensors. The wires are then attached to an electronic breadboard with a mounted "Spark Core" microprocessor. While this approach has been successful i.e. sensor lengths corresponded to the beam deformation, the approach is not scalable into 3d cases. Active research is undergoing in strategically embedding sensors in everyday objects while **A.** maintaining the functionality of said objects and **B.** ensuring reliable and accurate deformation measurement. These methods employ high precision tooling and processes to ensure these conditions are met. The aim of this surface-mounted sensor prototype is to demonstrate that syncing sensor and body is viable. Future research in



Figure 32: 3d printed mold (top) and rubber beam specimen with embedded sensors (bottom).

appropriate fabrication techniques would allow embedding the sensors and capturing body rather than surface deformations.

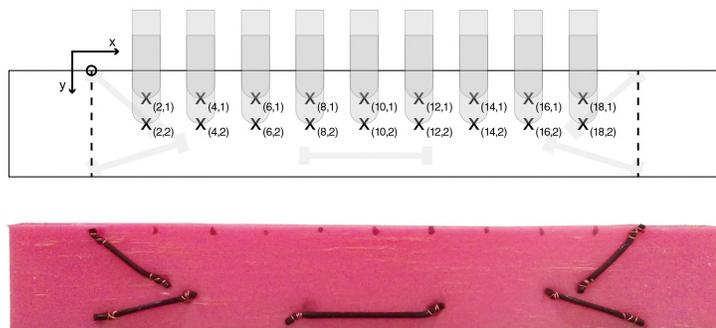


Figure 33: Local coordinate system and locations of displacements (top) and rubber beam specimen with surface-mounted sensors (bottom).

A beam-specific local coordinate system was also defined [Figure 33](#) with the top left corner of the beam as origin. This system is based on the loadcases previously defined in the numerical FEA portion of the research. Mechanical displacements by the training apparatus described hereafter will utilize said coordinate system.

4.2.2 *Material trainer apparatus*

The second generation of the material training apparatus [Figure 34](#) is built upon the first iteration described in [Chapter 3](#). Two stepper motors ensure accurate positioning of both X and Y- axes. An Arduino microprocessor controls the number of coil turns within each motor and subsequently the location of the pin (in blue). Stepper motors hold their position as long as they are wired making them ideal for such application.

A chassis made of acrylic sheets and threaded rods ensure the setup is stable during training. Adjustable supports allow for training beams of different lengths.

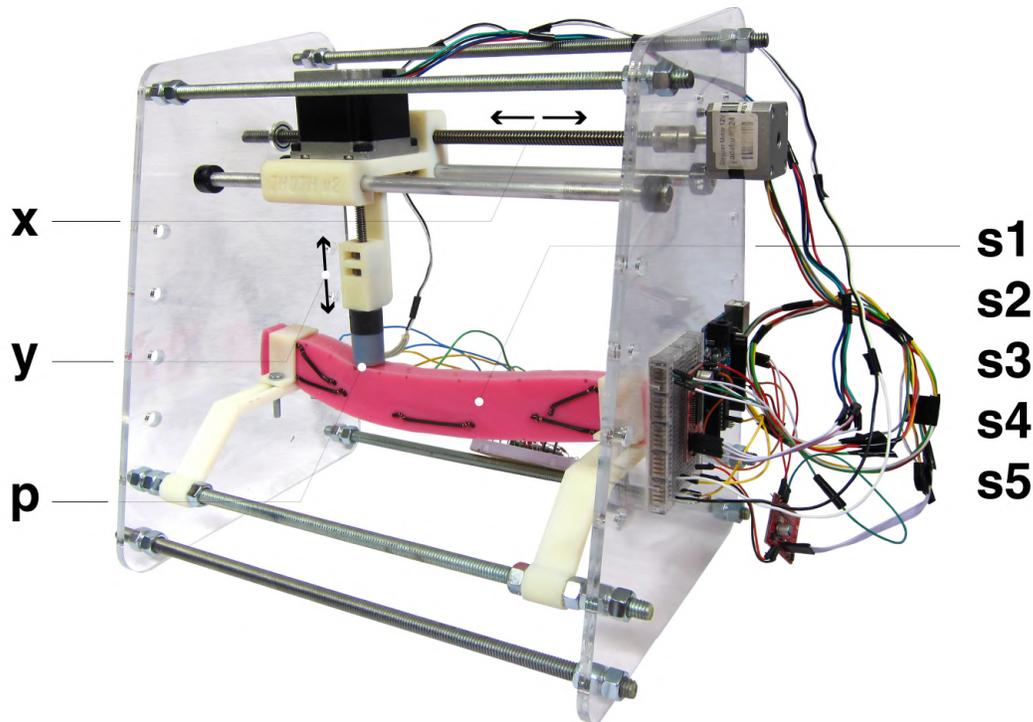


Figure 34: Second generation material training apparatus.

Improvements in this apparatus include:

- **Adjustable X-axis:** Displacements can now be applied at different locations across the length of the beam in addition to varying the Y-axis height.
- **Pressure sensors:** Since displacements are applied, a pressure sensor is used to quantify loads. This sensor is essentially a force sensitive resistor with its resistance varying with the amount of pressure applied to the sensing area.
- **Control/monitoring dashboards:** The apparatus is coupled with two dashboards [Figure 35](#). The first is a machine control dashboard connected to the apparatus-mounted Arduino. It is based on Guino, a dashboard system developed for Arduino. Here, desired X and Y locations of the pin are set. The pressure value P is then recorded. The second is a sensor monitoring dashboard connected to the beam-mounted Spark Core. It is a web based dashboard that utilizes client-side JavaScript to access sensor readings as well as the enoch Javascript library to visualize real-time graphs. Here, all 5 sensor readings are monitored. They are then written onto a comma-separated file (CSV) for storage and further processing.

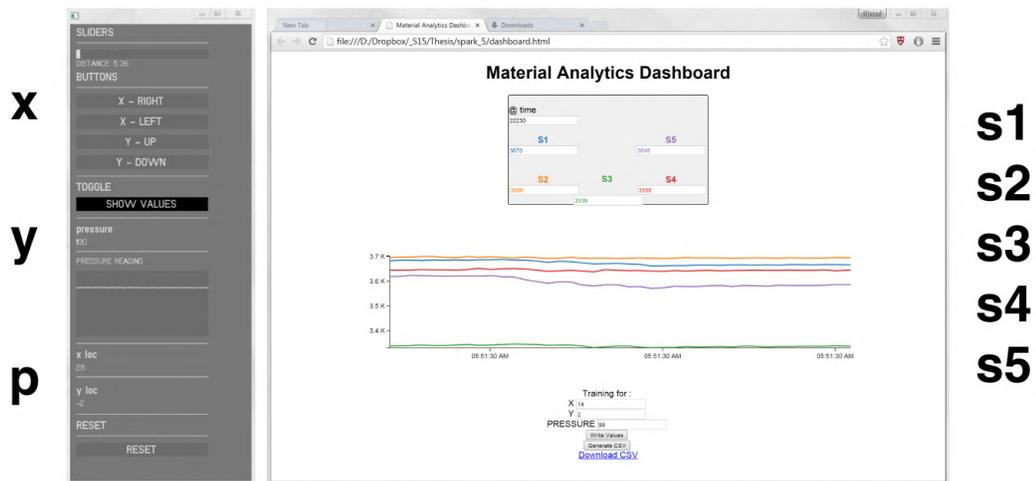


Figure 35: Material training dashboards. Machine control (left) and sensor monitoring (right).

4.2.3 Computer vision

While specific loading conditions are being applied and sensor readings taken, it is crucial to record the corresponding displacement fields as these will act as training data in the machine learning framework. Two methods have been explored:

- **Template Matching:**

The python OpenCV library was utilized to find a specified template within an image of the specimen, whether loaded or unloaded. There are two primary inputs to the algorithm **A**. the source image in which to expect to find a template and **B**. the template itself. The template image is compared against the source by sliding it one pixel at a time, left to right and up to down. At each location, a metric is calculated relative to the strength of the match. Results are stored in a 2d matrix. The maximum values in this matrix thus represent high correlation and the template coordinates in the source image [14].

The template here is seen as a nodal marker - placed at nodes where displacement measurement is desired. A star shaped geometry is used and passed on to the algorithm as the template. The same geometry is then superimposed at six location on the specimen. While the template matching algorithm is able to successfully identify the feature locations within images acquired during testing [Figure 36](#) , there are several drawbacks to this method:

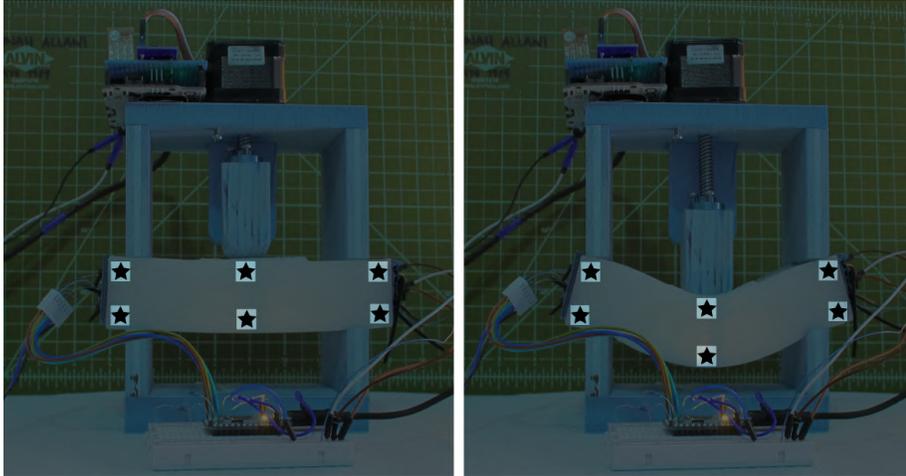


Figure 36: Templates detected successfully within images aquired during testing.

* The template size in pixels must match the feature size within the image. Additionally, if the feature has been rotated, it will most likely go undetected. This is likely to happen due to the large deformation nature of the specimen. The method is therefore unable to detect the feature if it has been scaled, rotated, skewed or been subjected to non-affine transformation.

* The feature count on the specimen represents a mesh where displacements are to be measured. By increasing the mesh resolution, more features will need to be appended to the specimen and eventually detected. This adds complexity and is computationally expensive.



Figure 37: Star template used (left) and noise generated during template detection within images (right).

* The algorithm is more well-suited where a single occurrence of the template is to be detected such as finding Waldo in a "Where's Waldo" puzzle. For finding multiple occurrences, one needs to set a threshold and search the result matrix space for these values. Additionally, multiple peaks are detected per occurrence. [Figure 36](#) extraction routine used the max 80 values in the result matrix to detect 6

occurrences. Noise is attributed to this discrepancy [Figure 37](#). Even with clustering the top matches into 6 coordinate pairs, it is difficult to ensure these will be in the same order when the algorithm is run on another image. Hence, it will be not be feasible to compare unloaded and loaded images.

- Digital Image Correlation:

Digital Image Correlation (DIC) is a computer based image analysis technique utilizing random patterns. Due to recent advances in computers, digital cameras and image processing software, DIC has been extensively used in the measurement of mechanical displacements and deformations.[23] Patterns used in DIC have been widely researched as well. Stoilov et al. have developed image generation algorithms to improve the accuracy of DIC. As is the case with template tracking, two major drawbacks araised when exploring the utilization of DIC:

* DIC is more suited for 2d problems where patterns can be expressed on flat surfaces. With light and shadow artifacts affecting accuracy, DIC is not scalable to 3d problems.

* The accuracy of DIC is relative to the number of frames taken during the deformation i.e. the number of images from when the load is first applied to full application. More frames enhance the correlation and allows the DIC algorithm to better track the random patterns across the frames. This then requires more data acquisition efforts during the experiment. Additionally, with the sensors mounted on the surface of the beam [Figure 33](#), printing speckle patterns on the surface is not viable.

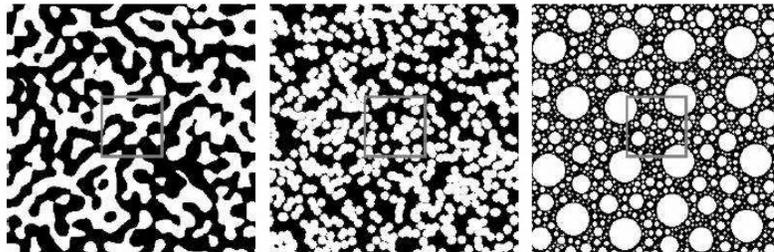


Figure 38: Computer generated random patterns used for image analysis with subset. [23]

It was hence decided not to utilize these computer vision-based techniques. A more robust method would be to generate a displacement lookup table from the previously generated finite element analysis. Such table would include the displacement field per load case. The table is explained in detail in [Section 4.4](#). With the increase in 3d printing resolution, a potential avenue of research would be to print objects with embedded patterns in one build - in lieu of simple surface application. The distribution of acquisition cameras to track pattern embedded objects is an interesting alternate approach to utilizing strain sensors.

4.2.4 Data exploration

Prior to using the experimental training data to generate the prediction model, exploratory data analysis is performed revealing some useful insights. The first of these exercises involves plotting all 5 sensor values across all trained load cases. Each polygon in Figure 39 illustrates a single load case. There are two main observations here. In both 1 and 2 cm deformations, the graph is not symmetrical around the "s3" dotted line. It is expected that the sensor readings be more or less symmetrical as the sensors themselves are. As the sensors have been cut and installed manually, it happened that "s5" is longer than "s1" causing its readings to be generally lower. The second observation deals with the fact that the sensor readings for most loadcases are generally lower than those for the unloaded model (shown in red). This implies that all sensors have undergone tension causing their resistance to increase and the voltage readings to drop. From the numerical analysis model, the sensor expected behavior is as follows: "s1", "s3" and "s5" are in tension while "s2" and "s4" are in compression. The data hence shows us that "s2" and "s4" have undergone tension rather than compression. This is perhaps due to their orientation being relatively horizontal or their length being relatively long.

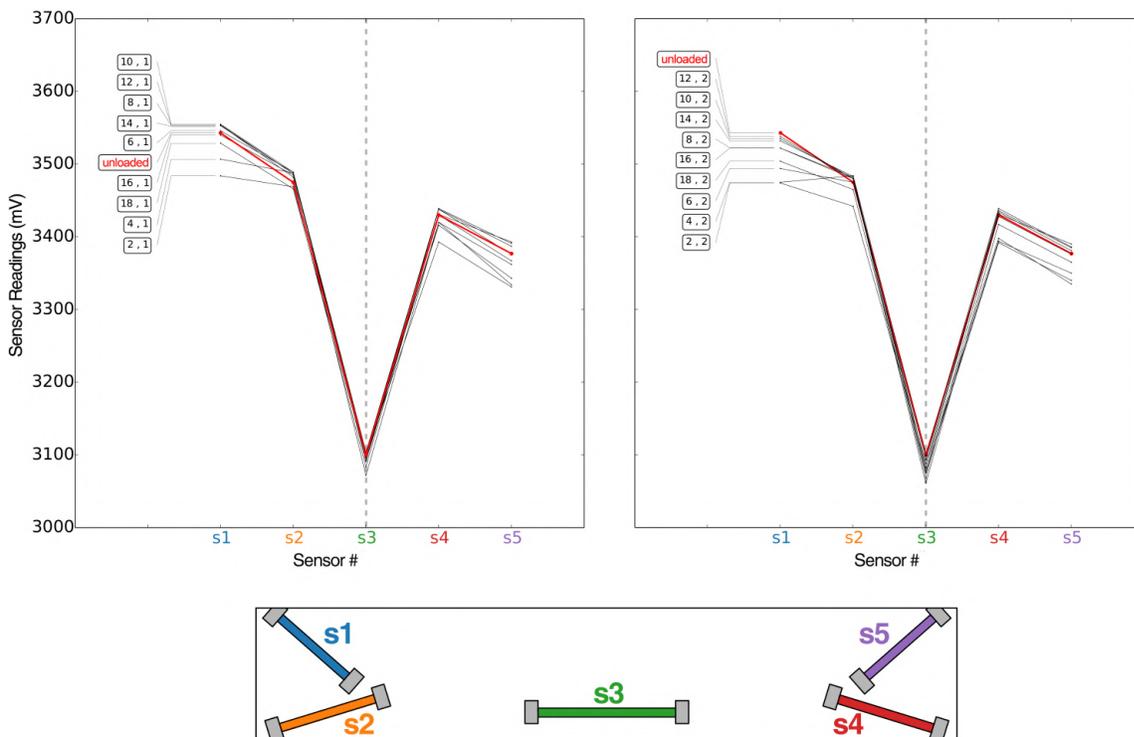


Figure 39: Sensor readings at 1 and 2 cm displacements.

The second exercise explores noise levels in the data and how that would effect the accuracy of prediction models built using this dataset. Two loadcase pairs are isolated and compared Figure 40. The red halo around the first load case represents the noise threshold of that particular loadcase. If all 5 readings of another loadcase (in blue) fall within the noise threshold of the other, the model accuracy is weak as the two loadcases can't be differentiated. If, on the other hand, at least one of the 5 readings falls outside

that threshold level, a good prediction accuracy is expected. The more readings that fall beyond the threshold, the greater the accuracy. Hence, the signal/noise ratio will highly affect the accuracy. This can be addressed by using sensors with a low signal/noise ratio. Additionally, multiple sensors placement configurations could be explored and tested as a means to maximize the differences between readings for individual loadcases.

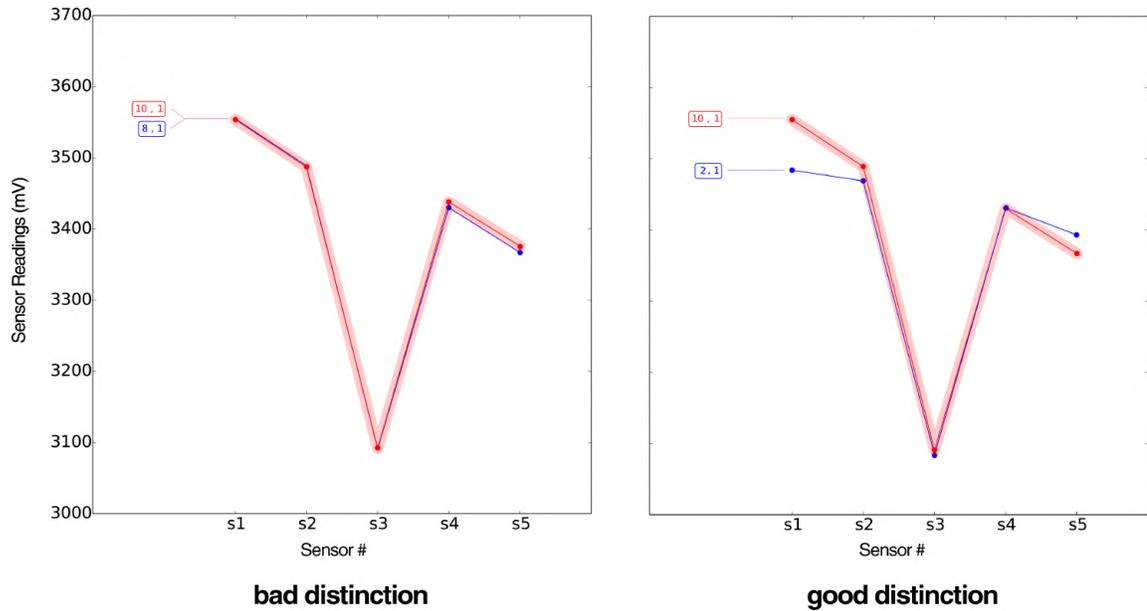


Figure 40: Noise and prediction accuracy.

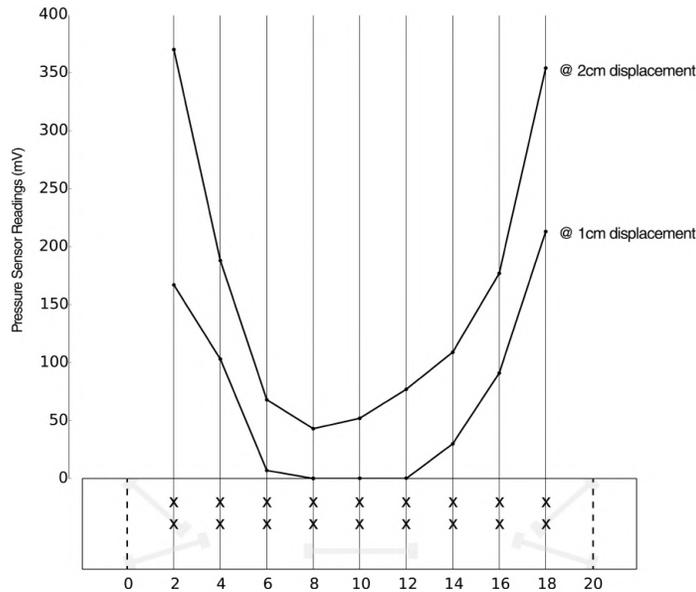


Figure 41: Pressure sensor readings.

The final exercise plots the measured pressure along the entire length of the beam. As expected, it is harder to move material towards the supports and thus pressure readings

are generally higher in these areas. Even when applying a displacement of 1cm, the mid-span region of the beam experiences no pressure. This is due to the fact that the unloaded beam had a preexisting sag. Thus the displacement of the pin did not cause any corresponding displacements in the beam.

4.3 MACHINE LEARNING

As scientific fields are best described by the questions they attempt to answer, the field of Machine Learning seeks to answer : 'How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?' [21] A survey of machine learning algorithms was conducted to identify the most appropriate for the purpose of this research. There are generally two types of machine learning algorithms: Supervised algorithms where input data with know results is used for training and unsupervised learning where the training data has no known results i.e. it is not labeled. The experimental data generated from the previous exercise is labeled and hence a supervised learning algorithm is to be explored. In selecting the appropriate algorithm, the supervised space is broken down into three components: **Representation:** A classifier must be represented in some formal language that the computer can handle. **Evaluation:** An evaluation function (also called objective function or scoring function) is needed to distinguish good classifiers from bad ones. **Optimization:** A method to search among the classifiers for the highest-scoring one. [6]. The key is to select the appropriate combination of methods from Figure 42.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
<i>K</i> -nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Figure 42: Common examples of the three machine learning components [6]

The scope of this section will focus on the representational component. Linear regression was selected as the most widely used of statistical techniques and one of the most traditional modes of machine learning. It is also fairly simple to interpret as oppose to other 'black box' algorithms such as neural networks. Additionally, it is used to predict continuous variables as oppose to classes as is the case with *K*-nearest neighbor algorithms. Linear regression models simply describe the relationship between a dependent variable (*Y*) and independent variable or variables (*X*) which are also know as the predictors. In our case, the dependent variables (*Y*) are the load case parameters: namely the *x* and *y* coordinates of applied displacement as well as the applied pressure *p*. The independent variables (*X*) are the five sensor readings i.e. predictors. Since we are dealing

with multiple dependent and independent variables, the process is identified as multivariate linear regression.

The dataset is randomly split into two halves, one for training and the other for testing. The model is fit with the training half and is supplied with the independent variables (predictors) from the testing half. A score is then given to the model on how close the predicted values are from the actual ones. Figure 43 shows the actual values on the horizontal axis and the predicted values on the vertical for each of the three predicted quantities. Here we notice that the model has a high accuracy in predicting x and p . This is illustrated by the high correlation values between actual and predicted values. The model has performed poorly in predicting y . This is mainly due to the small data sampling as y readings only consist of three integers 0, 1, and 2. Overall, the model has a good prediction score of 0.79.

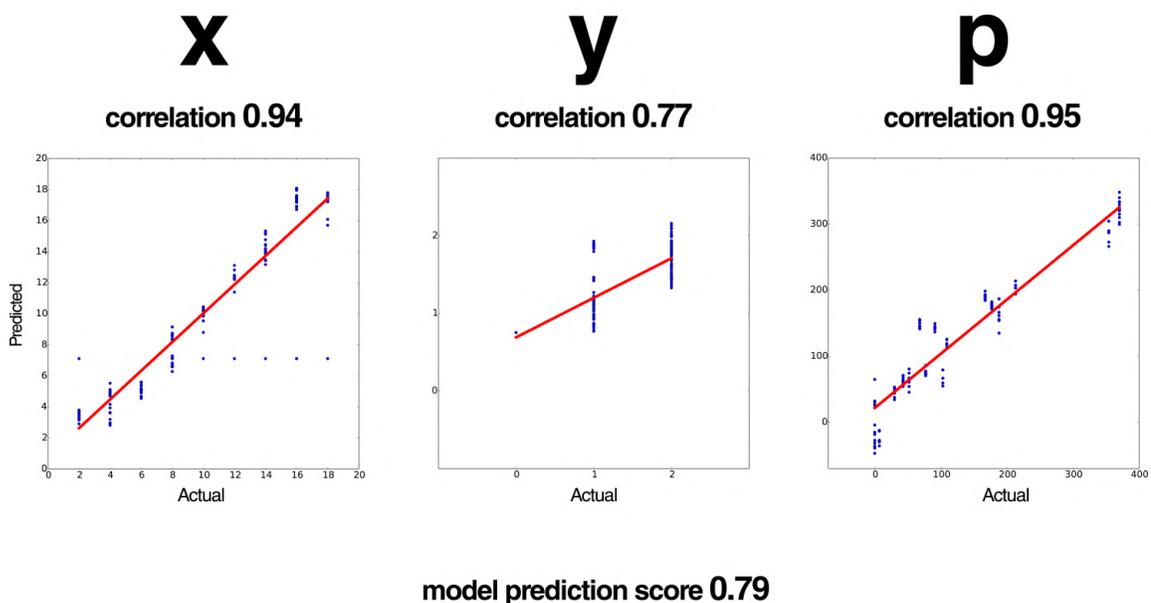


Figure 43: Results from a multivariate linear regression model.

The second attempt at creating a regression model involved calculating the natural logarithm of both the dependent and independent variables and using these to fit the model. It is generally accepted to fit regression models with the log value as oppose to actual values. This reduces the effects of outliers, helps linearize the model and eventually should produce a better fit and a well behaved model. It does so by altering the scale and making a skewed variable more normally distributed. Figure 44 shows results from the log-based model. The logarithm approach appears to have a negative effect on the accuracy and prediction power. This is, again, due to the interger variables comprising the value of Y . The log value of 0 is undefined and the log of 1 is zero. The completely skews the data rather than linearize it. By association, since multivariate linear regression takes into account correlation between the multiple dependent variables, the prediction accuracy of both x and p has also been affected negatively. It was thus decided to use the

actual values rather than their natural logarithm as this approach is not valid with this particular dataset.

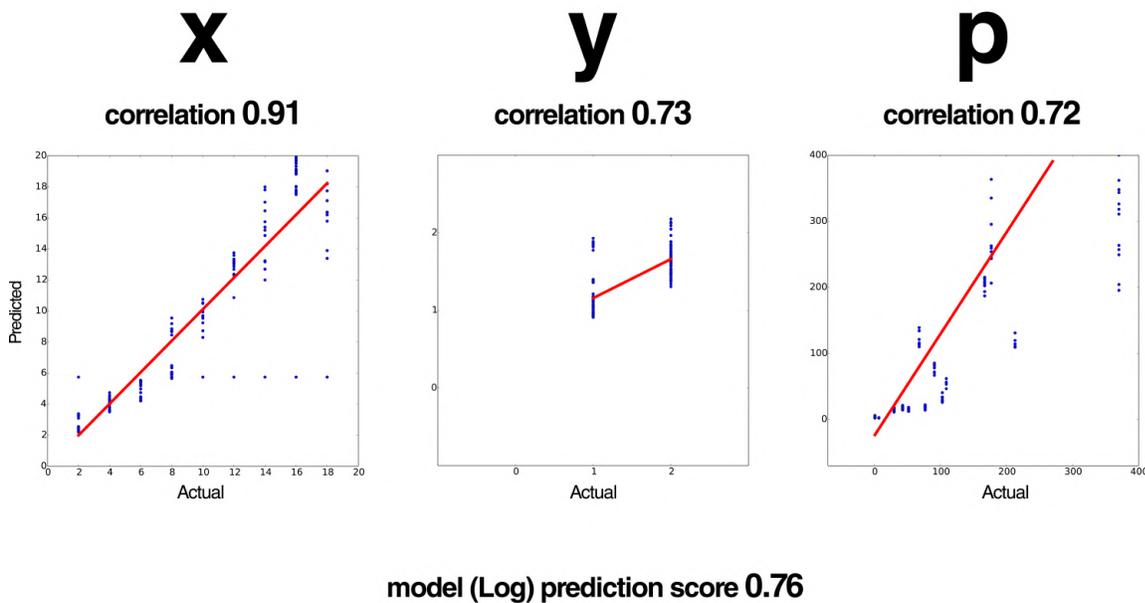


Figure 44: Results from a multivariate linear regression model using the logarithm of both dependent and independent variables.

As a first step to optimizing the model prediction accuracy, further models were created. Figure 45 shows the results obtained from five regression models built with each single sensor reading as a single predictor. This allows us to visualize the prediction power of all 5 sensor readings relative to the predicted values of x , y and p . We notice here that models that were fit using "s4" and "s5" show good accuracy in predicting x . On the other hand, models fit using "s1" and "s2" show good accuracy in predicting p . Models fit using "s3" shows slight accuracy in predicting y . These are most likely caused by sensitivities in the physical specimen having been assembled manually. In a more controlled experimental environment, this variation in accuracy across predictors is crucial in future optimization efforts. Instead of using one model fit with all five sensor readings, multiple models could be used and fit with a different combination of sensor readings to give the absolute best prediction accuracy. This could be accomplished by a brute-force search where models are fit with every possible combination of predictors allowing an exhaustive systematic search of prediction space. It was decided to utilize the single model fit using all five predictors Figure 43 while identifying potential evaluation and optimization possibilities.

Given the complexity of the prediction process, it is crucial to demonstrate the workflow through an example Figure 46. One of the data points within the training set recorded the loadcase $x=14\text{cm}$, $y=1\text{cm}$ and $p=30\text{mV}$ together with its corresponding five sensor readings 3550,3488,3105,3415 and 3344mV. When the same set of sensor readings are passed on to the model, the model predicts the loadcase $x=14.5\text{cm}$, $y=0.8\text{cm}$ and

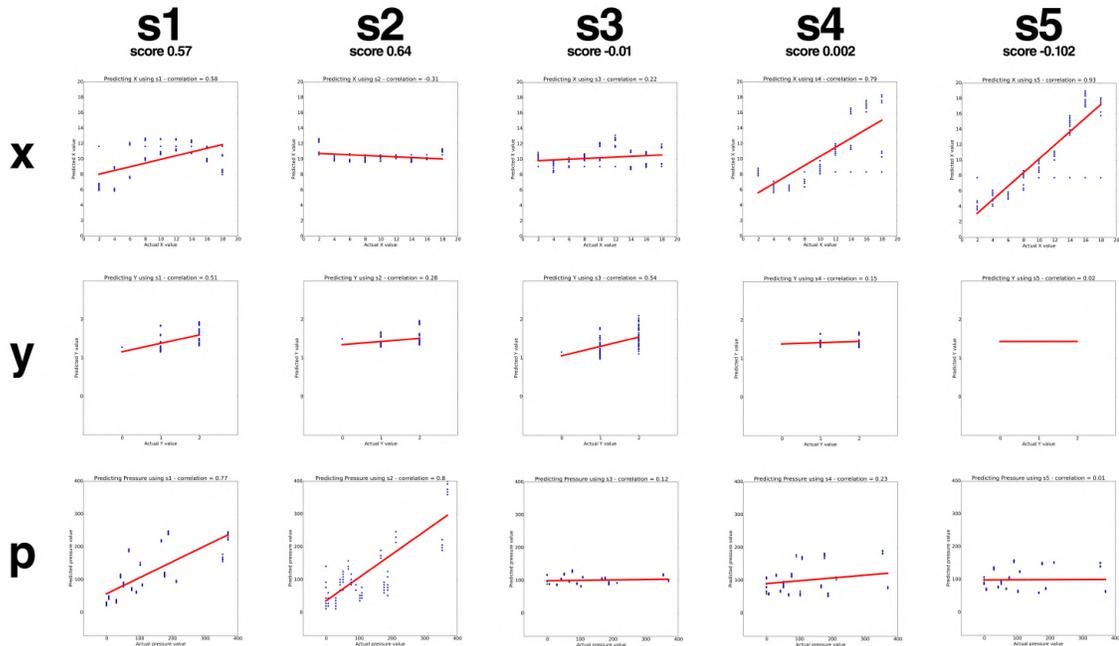


Figure 45: Results from five models created with each single sensor readings as a single predictor.

$p=47.4mV$. these are essential values at the regression line i.e. the best fit line. The predicted values are relatively close to the actual values using the same predictors.

	loadcase parameters				predictors				
	x_{cm}	y_{cm}	p_{mV}		$s1_{mV}$	$s2_{mV}$	$s3_{mV}$	$s4_{mV}$	$s5_{mV}$
actual	14	1	30	→					
predicted	14.5	0.8	47.4	←	3550	3488	3105	3415	3344

Figure 46: Example illustrating the regression model accuracy.

Now that a loadcase has been predicted, we move on to predicting the displacement field i.e. the deformation pattern of the beam under this specific loadcase. For this purpose, a displacement lookup table is utilized [Figure 47](#). This table is not based on experimental but numerical analysis data. This represents an instance of combining both numerical and experimental methods. As the predicted displacement coordinates x and y are unlikely to exist as is in the lookup table, we need to search for X closest matches in the table. When found, weights are applied to the matches based on their proximity to the predictions. Linear interpolation is then used where the matches' displacement fields are multiplied by their corresponding weights and added up to give a predicted displacement field. It is important to highlight that the numerical data is based on non-linear analysis. Performing linear interpolation on the data is thus questionable. Due to lack

of alternative options, this concern will not be addressed. This concludes the machine learning section which, given training data, is able to predict displacement coordinates, pressure and field.

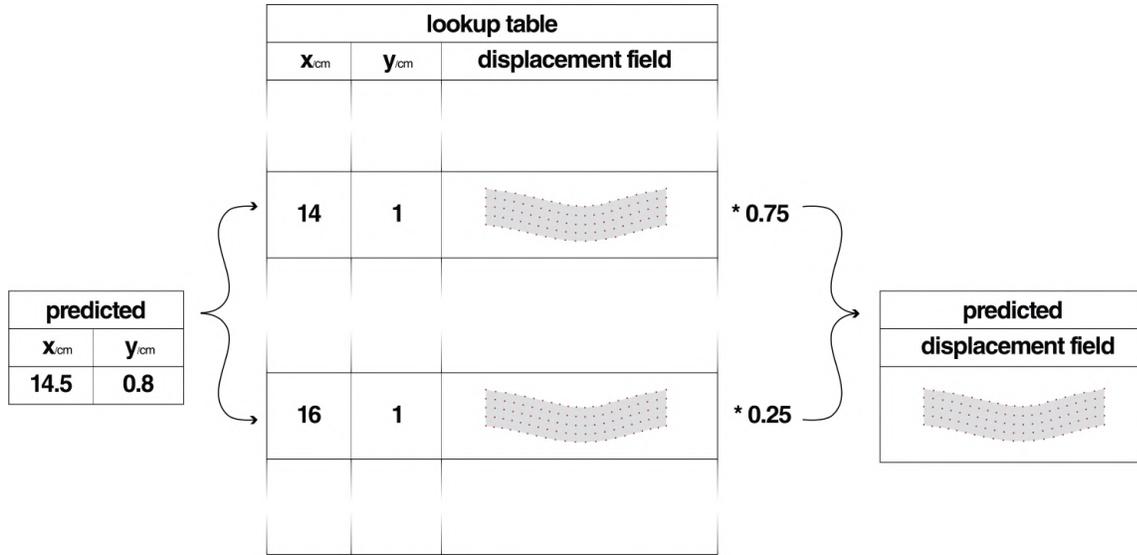


Figure 47: Predicting displacement field from lookup table

4.4 SYSTEM ARCHITECTURE & DATA MINING

Now that a methodology for material analytics has been defined, this section will attempt to construct a framework where this methodology can be applied. Figure 48 illustrates a simplified version of such framework. It is organized around a central cloud-based prediction application. Cloud computing is defined by the NIST as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services)[11] Economies of scale are thus achieved through the sharing of resources, very similar to an electricity grid where you only pay for what you use. The cloud thus allows the system to be fully scalable as the number of connected objects grows or shrinks. Additionally, it provides a one stop access point to the prediction algorithm i.e. the prediction does not happen on the clients but on a server. This allows further work on improving prediction accuracy to happen on the cloud, and on the cloud only. It also enables the framework to connect to different types of clients making it cross-platform. Therefore, regardless of the object’s embedded electronics, its ability to push sensor readings to the cloud is the only prerequisite. Each of these communication strings are explained in detail in this section.

4.4.1 Sensor-embedded objects

In addition to sensors, these deployed objects are also loaded with wifi enabled micro-processors that allow pushing sensor data to the cloud at regular intervals (currently set at 500ms). That is, these electronics are in themselves cloud-based; Connecting objects to

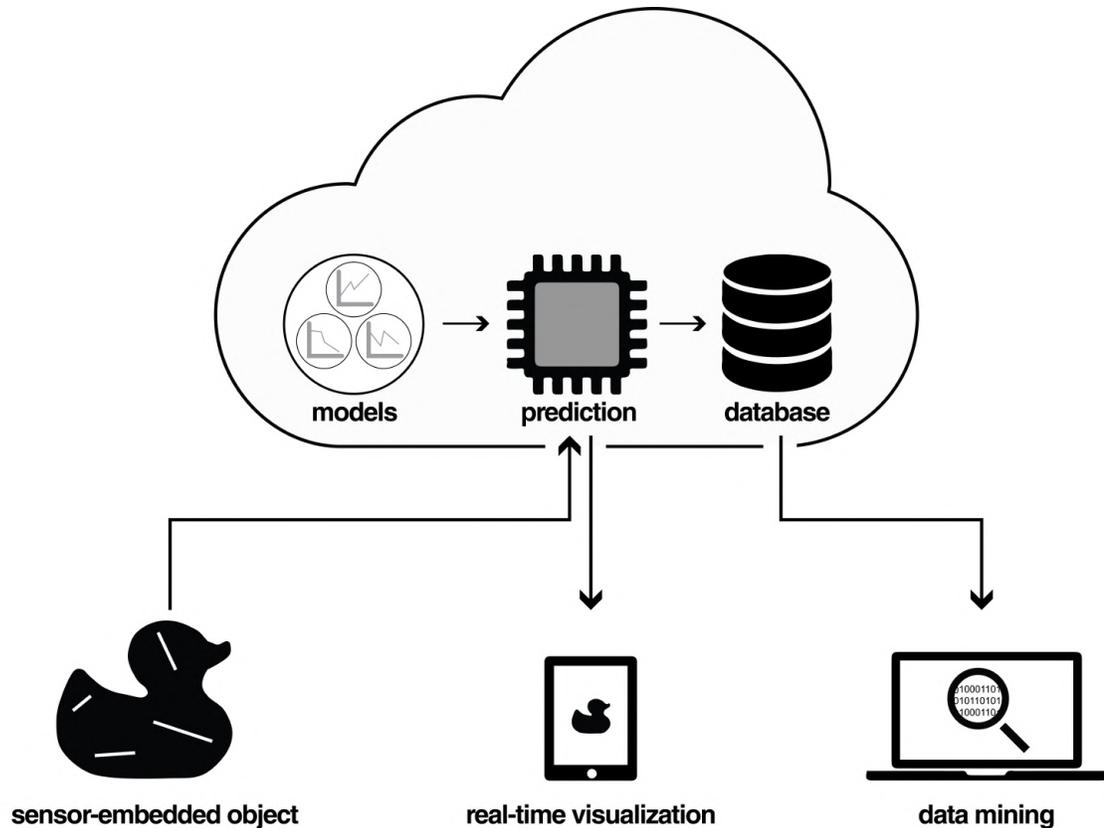


Figure 48: Diagram illustrating the system architecture.

such electronics automatically deploys the objects online. The cloud can then be accessed through an HTTP address, similar to accessing a web page. Appendix item 6 represents a JavaScript object notation (JSON) that is returned whenever the address is called. JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is used primarily to transmit data between a server and web application [19]. While this JSON might contain many microprocessor-specific attributes, that main interest lies in three attributes and their corresponding values: **deviceID** which returns a unique object identifier, **lastHeard** which returns a time stamp at which the sensor readings were taken and **result** which returns the five sensor readings as a list.

4.4.2 Cloud prediction & database

The Material Analytics cloud is preloaded with object-specific prediction models. It is assumed that objects are trained and their training data is loaded onto the cloud prior to the objects being deployed in real-life interaction environments. As objects differ in material and geometry, their individual training data is unique. Two identical chairs would use the same prediction model but a chair and a stool would use different models. Hence, the deviceID is crucial as it identifies which model is to be used. After the

appropriate model is selected, the sensor readings are supplied as predictors and a prediction is made. Appendix item 9 represents the JSON returned when a call is made to the prediction cloud. The **predicted** attribute contain nested attributes within including: **predictedX** and **predictedY** representing the predicted x and y coordinates of the displacement pin and **predictedP** representing the predicted pressure value at the displacement location. These predictions are a result of the multivariate linear regression model outlined in Section 4.3. Other attributes including the **predictedXDisp** and **predictedYDisp** represent a vector of nodal displacements in the beam mesh across the x and y axes respectively. These are generated through the displacement lookup table illustrated in Figure 47. Due to the processing time needed to preform predictions, a new JSON is generated every 5 seconds. That is, new sensor readings are logged in every 500ms and a new loadcase prediction is made every 5 seconds. Further development is to be made on improving the prediction speed and attempting to match its resolution to that of the raw sensor readings data.

Eventually, this JSON constitutes an entry into the database, which is also cloud-based. The database utilized here is the NoSQL database MongoDB. A NoSQL database allows storage and retrieval of data that is modeled in means other than tabular relations used, for instance, in MS excel worksheets. This framework allows for a simpler schema and horizontal scaling. [20] Essentially, the database can be seen as a large empty canvas on top of which these JSON elements sit in no particular order. It is through their attributes that these JSON elements could be aggregated and insights from the data explored. Section 4.4.4 outlines the data mining process in detail.

4.4.3 Realtime visualization

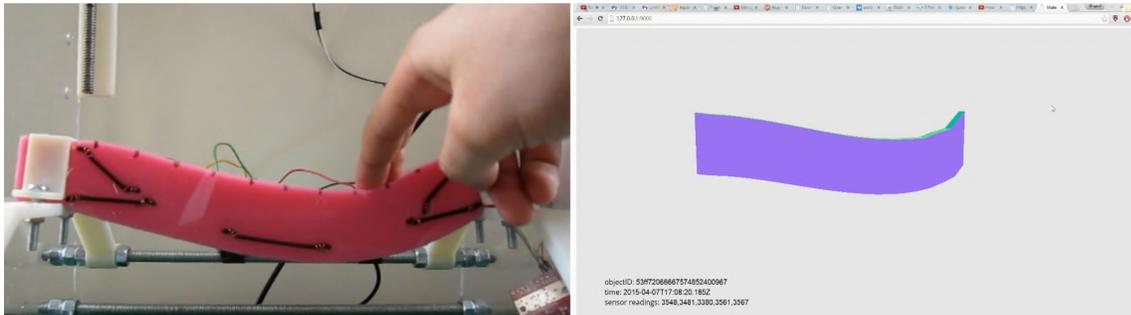


Figure 49: Real-time visualization of beam deformations on a web browser.

Direct connection to the cloud-based prediction algorithm also allows for real-time visualization of deformations. A node js environment is set with a server side application responsible for communication with the cloud. The Javascript library ThreeJS is utilized to display a digital model of the beam on a web browser. Once a prediction is made, the x and y coordinates of all nodes comprising the mesh are passed onto ThreeJS. The mesh is then updated to reflect the latest predicted deformation shape. It is crucial the node numbering sequence of incoming predictions (these originally match the Abaqus FEA mesh) match that of the ThreeJS mesh. Both loadcase predictions and mesh updates are in sync occurring every 5 seconds. The web interface also displays the the deviceID,

time stamp as well as sensor readings. As users would eventually be more interested in analytics regarding their use of daily objects, real-time visualization is expected to be complementary to that effort. The main goal behind developing this interface is to prove the accuracy of the prediction model. Around 70% of predictions were correct with some lag due to processing time needed to make the predictions. The remaining 30% were incorrect. It is also important to note that fact that the beam/sensors being symmetric introduced sensitivities into the prediction model. Thus incorrect predictions tend to be on the other side of the symmetry line relative to the actual applied displacements.

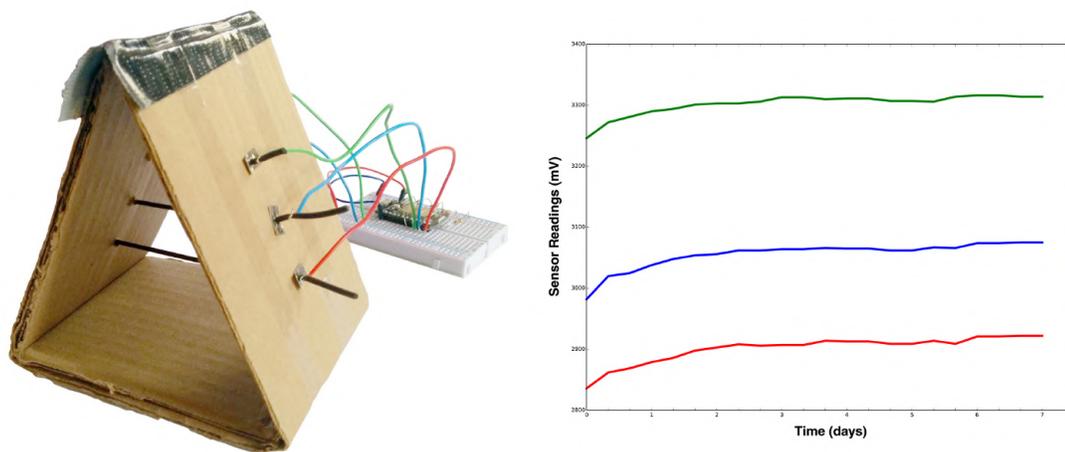


Figure 50: Readings from 3 sensors at rest across a period of 7 days.

As more time passed after training and visualizing, a decrease in prediction accuracy was noted. This then led to a series of experiments aiming to identify whether the sensor readings, over time, differed from those recorded at the time of training. The first experiment eliminated the rubber beam and tested three sensors of different lengths. The sensors were pre-stressed and mounted onto a jig. Their readings were recorded every few hours over a period of seven days. Figure 51 shows the experiment setup and a plot of the data collected during this period. Each color denotes a sensor with the red being of the longest sensor and the green being of the shortest. We notice that there is a slight increase in sensor readings over time. By the 5th day, the readings start to plateau and maintain a relatively constant value. This behavior is attributed to some micro tension being lost. As this tension is lost, carbon molecules become more closely packed without any change in the overall length of the sensor. This increases the conductivity and hence causes an increase in the readings. Therefore, any training on the beam should take place at least 5 days after installation.

The second experiment was conducted on the sensor-embedded beam itself. With the beam being at rest i.e. completely unloaded, readings were taken over a similar period of 7 days. Figure 51 illustrates the experiment setup and the five colors used in denoting the different sensors. The results show a steep decrease of values over time. This decrease is acceptable if the sensors maintain their relative differences. In other words, such hypothetical decrease would not affect the accuracy of the prediction model as the linear relation would simply be shifted down along the regression line. However, the actual

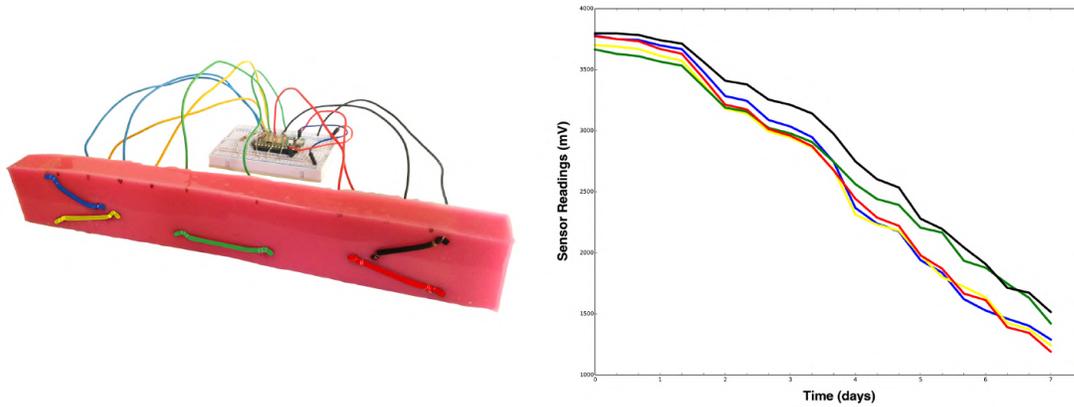


Figure 51: Readings from 5 embedded sensors at rest across a period of 7 days.

data shows the readings crossing one another meaning that in fact they do not maintain their relative readings. This renders the prediction model completely obsolete as retraining would be required before relatively accurate data is gathered. The plot shows relative data consistency and stability during the first few hours of day one right after training. This period has then been identified as suitable for data gathering.

4.4.4 Data mining & design recommendations

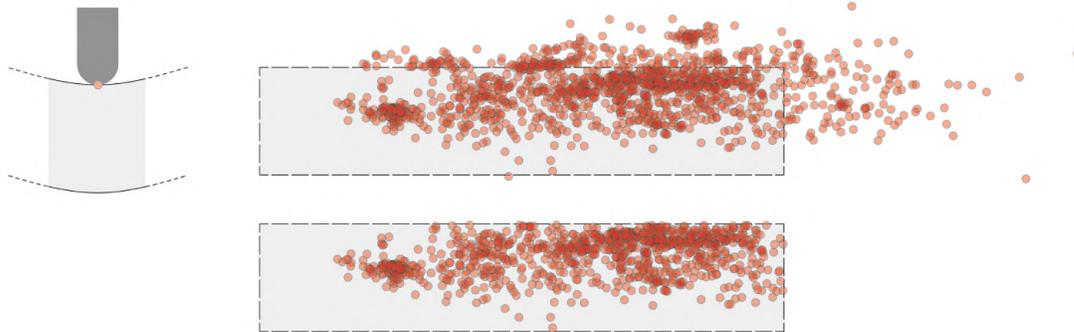


Figure 52: 1500 predicted displacement coordinates (top) and cleaned version eliminating predictions outside beam boundary (bottom).

With results from previous experiments in mind, data gathering was performed by manually applying displacements by hand onto the beam for a period of two hours right after training. This generated around 1500 database entries at a time spacing of 5 seconds. These entries were downloaded from the cloud-based database in JSON format for offline mining exercises. Figure 52 shows a plot of the gathered data. Each red marker represents the x and y coordinates of a predicted pin location. More manual displacements were deliberately applied towards the right side of the beam to showcase variation in the data. The accumulation of predicted points towards the right reflects this. Another observation

is the predictions that fall beyond the boundary of the beam, to the top and to the right. It seems that the prediction algorithm had a tendency to "over-predict" values along both the x and y axes. Hence, the first operation performed on the data is a simple cleanup eliminating all predictions beyond the beam boundary. Around 300 entries were deleted.

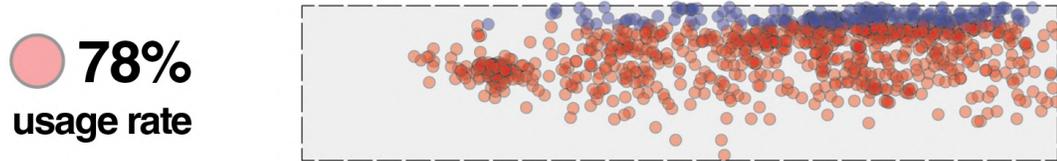


Figure 53: Object in use at red markers and object at rest at blue markers.

Due to inaccuracies and noise in the predictions, we can assume that those prediction close to the contact surface could be regarded as non-contact loadcases i.e. the beam was unloaded and at rest at these moments. A threshold of 5mm below the contact surface was defined as illustrated in figure [Figure 53](#). Predictions in blue markers fall within that threshold and those in red fall beyond it. This thus allows us to develop a usage rate metric. This value amounted to a 78% usage rate for this specific dataset, that is the beam has been interacted with 78% of the total data gathering time period. We often buy items that we end up not using. This is either due to the fact that we did not need these items initially or they did not function as intended. This usage rate metric could thus provide us indirect user feedback. The amount of time users interact with objects could potentially translate into a metric gauging their success. For users, this metric could then be translated into a dollar value. Users could then understand how the value of objects depreciates the more they use/interact with it. The dollar value could also help them on, for instance, deciding when to sell or replace an item when its value per unit time has dropped below a certain amount.

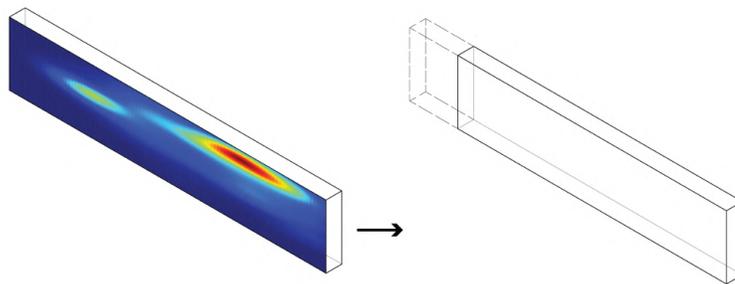


Figure 54: Lengthening by interaction density.

[Figure 54](#) (right) illustrates a color map of predicted displacements across one side of the beam. Perhaps the first observation is the lack of predictions towards the left end of the beam where a dark blue area dominates the map. The most intuitive recommendation

is thus to shorten the beam from the side that did not experience any interaction. This could be seen as an optimization opportunity where material is discarded when not in need. However, this recommendation will rely heavily on what this beam actually is and its role in a larger assembly. Does it make sense to shorten it? A second layer of data interpretation becomes crucial in translating these recommendations into viable design iterations.

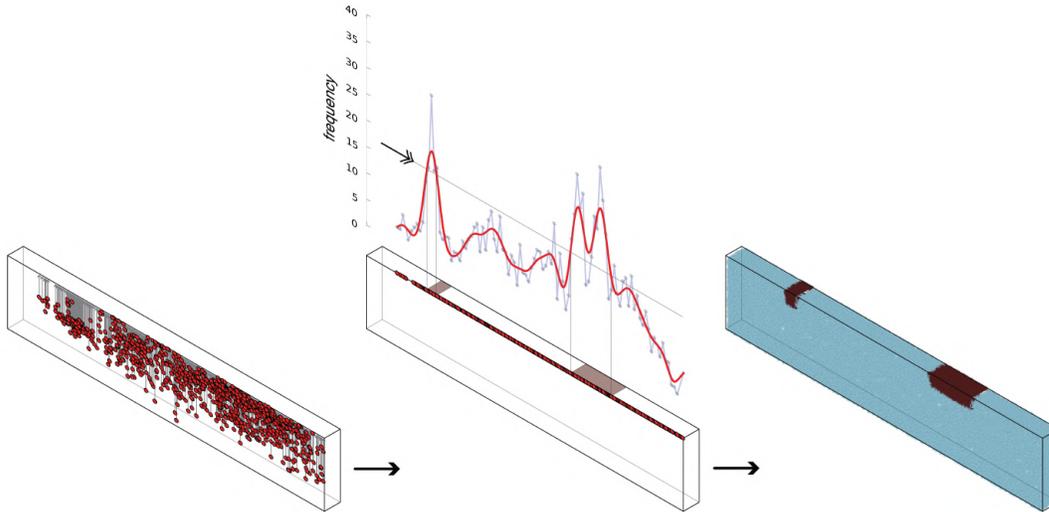


Figure 55: Design for durability (harden by frequency of contact).

The following data mining operations involve projecting all the predicted interactions up towards the contact surface. We can then plot a histogram of frequencies [Figure 55](#) along the entire length of the beam. This helps us visualize how the number of interactions vary as a function of beam length. The histogram plot is smoothed using a Butterworth filter that captures outlier frequencies in the dataset. Subsequently, a safety threshold could be set, above which frequencies are expected to cause damage to the material. Another design recommendation here would be to use a more durable material as surface treatments at these "high risk" areas. One could also plot pressure as a function of beam length as in [Figure 56](#). It could be assumed that the more pressure is experienced to move material, the less comfortable the process is. It could therefore be suggested that softer material be introduced along the contact surface, the thickness of which is directly proportional to accumulated pressure. To map out the pressure values, a best-fit curve is drawn through the data points (shown in red). This curve is then mirrored around the horizontal axis. Material above this curve is to be softer or more pleasant to touch than material below the curve. It may be argued that some fixed thickness of this soft material could be applied along the entire interaction surface without the need for data. However, this strategy allows a better and more optimized distribution of this potentially costly material. Additionally, it produces an aesthetic that emerges out of a purely data-centric design approach.

As the machine learning algorithm also outputs a predicted displacement field, the VonMises stresses can be calculated. This involves numerical computing where the Jaco-

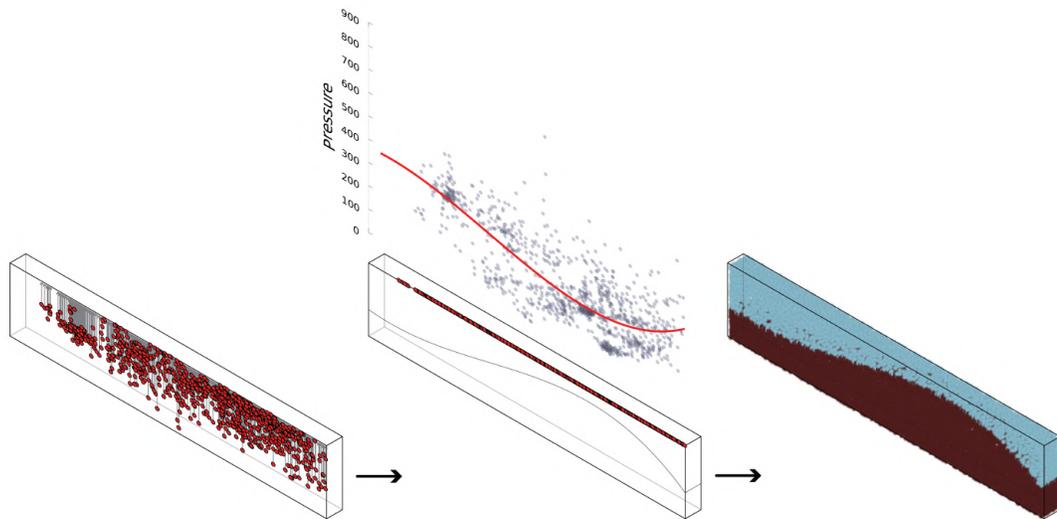


Figure 56: Design for comfort (soften by amplitude of contact).

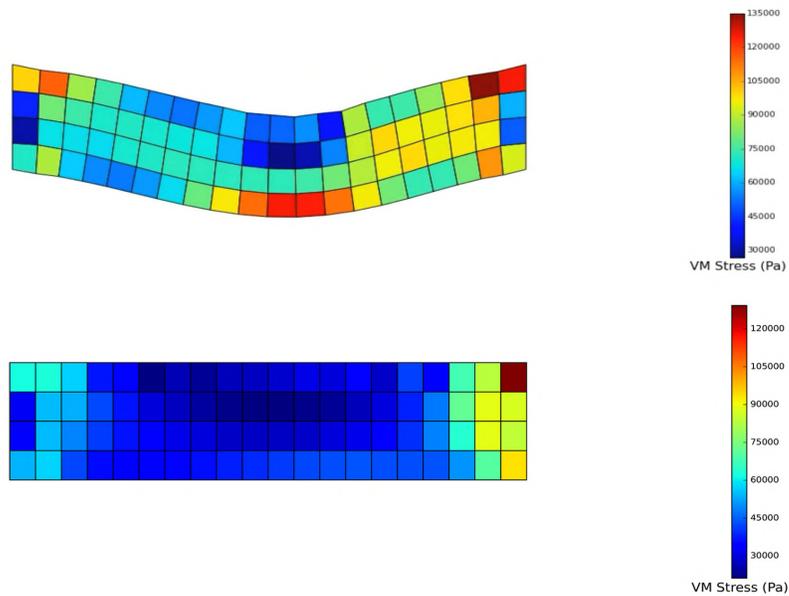


Figure 57: VonMises stress distribution from single loadcase (top) and average VonMises stress distribution from composite loadcases (bottom).

bian matrix and its inverse are calculated. This is used to compute the B matrix. Strains and stresses are then calculated as follows:

$$\epsilon = B^e * d^e$$

where d is the displacement vector

(6)

$$\sigma = D * \epsilon$$

where D is selected for plane stress problems

(7)

The scalar quantity VonMises stress is hence calculated from the stress tensor components. Figure 57 (top) illustrates the VonMises stress distribution on one of the 1500 predicted loadcases. These results could then be averaged out across all loadcases resulting in a composite loadcase giving an overview of stresses across the entire data gathering period Figure 57 (bottom). Instead of being generated from hypothetical numerically simulated data, these results are based on real-life conditions. Hence, the attractiveness of the method is highly evident in such averaged results. With this information in hand, multiple design recommendations could then be made. A thickening by stress approach could be suggested Figure 58. The constant thickness of the beam could be varied as a function of stresses. As the VonMises stress calculations are element-based, they are remapped onto the mesh nodes. The nodes are then displaced relative to their corresponding stress values; A higher stress value causes a larger nodal displacement and thus a locally thicker beam. Other approaches could include perforation by stress where parametrically-sized holes are introduced in areas of low stress concentration as a means to optimize material distribution and mass. Another bi-material approach would involve aggregating the averaged stress results into two groups based on some cut-off threshold. These two groups could be translated into two different materials of varying moduli so as to address the required material strength locally.

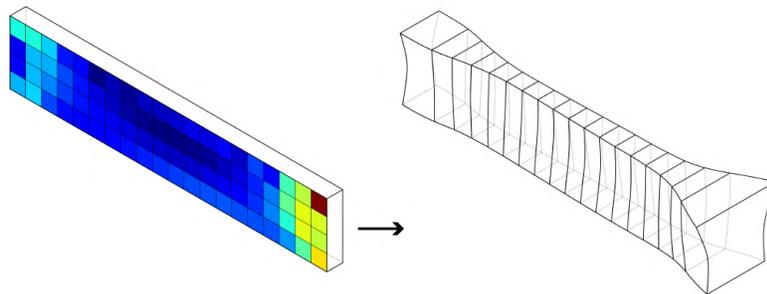


Figure 58: Thickening by Stress.

4.4.5 Fatigue prediction

Fatigue is the weakening of a material caused by repeatedly applied loads that are below the yield point. It is the progressive and localized structural damage that occurs when a material is subjected to cyclic loading. [16]. Parameters affecting fatigue are also inherently linked to the type of material subjected to these loads. It thus becomes crucial to define fatigue in terms of rubber-like materials, rubber being the material used to cast the subject specimen. The four major categories of factors contributing to fatigue in rubber are: the effects of mechanical loading history, environmental effects, effects of rubber formulation, and effects due to dissipative aspects of the constitutive response of rubber.[10] This research is solely concerned with mechanical loading history. Additional inputs, such as temperature and humidity, could be added as predictors enabling the study of how these correlate to fatigue and hence crack growth.

The quantification of fatigue needs to happen on the element level rather than on a loadcase by loadcase basis. For that, it was important to pivot the data allowing us to access stress quantities at the element level. Figure 59 illustrates a plot of stress across the data gathering period on a single element. Due to the high noise levels in the data, the readings were smoothed using the Butterworth filter allowing a better reading of loading cycles. With 80 elements in the beam mesh, 80 stress/time plots were generated.

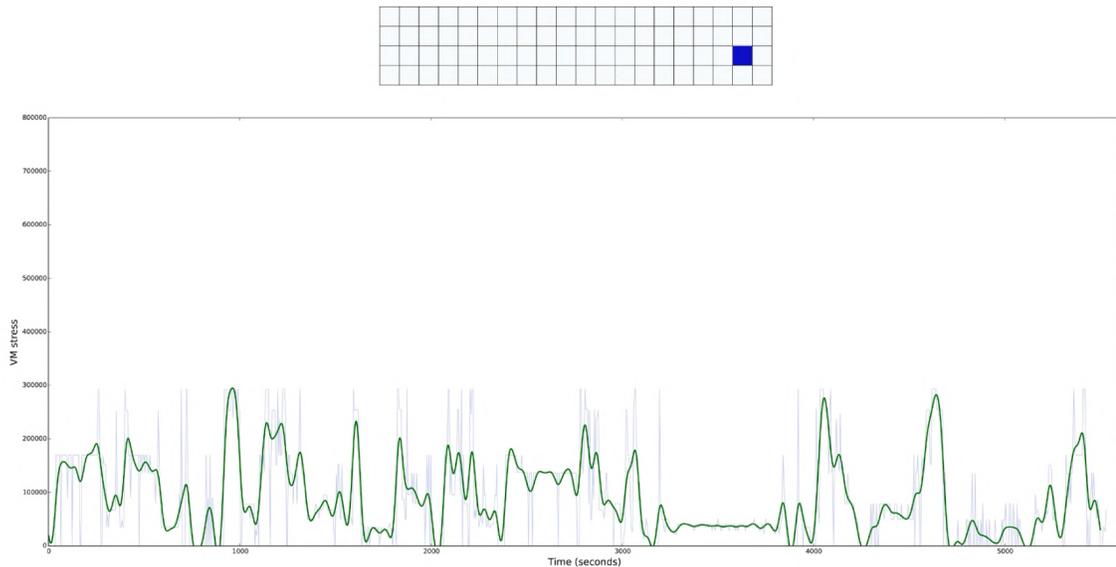


Figure 59: A plot of VonMises stress values across time for a single element.

Further analysis on these plots is material specific. What induces fatigue in one material might not have the same effect in another material. Under mechanical loading history, Mars et al. outlined three factors contributing to fatigue in rubber and these are hence extracted from the plots Figure 60 :

- **Mean Load:** This represents the average stresses acting on the element and are extracted by calculating the area under the stress/time curve.

- **Loading Frequency:** This represents the number of loading cycles and are calculated through either the number of peaks or valleys in the data. The effect of loading frequency depends on polymer type. For rubbers that exhibit strain crystallization, frequency does not have much effect on fatigue life. For amorphous rubbers, frequency has a larger effect [10].
- **Statically Strained Rest Period:** This represents the time at which the beam was under static loading as a percentage of total time i.e. time at which applied pin displacements remained static. These are extracted from flat portions of the curve with a slope less than a specified threshold.

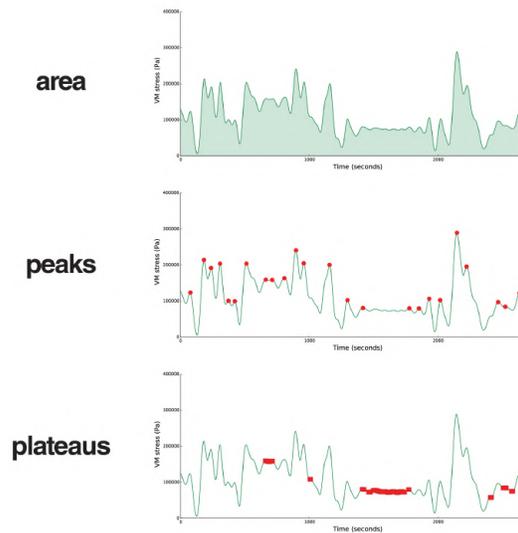


Figure 60: Fatigue prediction parameters of a single element.

By performing the above analysis on all elements, the distribution of each of these parameters can be visualized across the entire beam [Figure 61](#). The plots representing area under curve and plateau periods are somewhat the reciprocal of one another. Mapping out the number of cycles per element reveals an interesting and unpredictable pattern. These three parameters could then be aggregated to produce a hypothetical fatigue prediction metric. In doing so, different weights are applied to each of these metrics relative to their contribution to fatigue development. As a showcase, mean loading was given a weight of 0.65, loading frequency a weight of 0.25 and strained rest periods a weight of 0.10. The fine tuning and adjustment of these weights is crucial to improving the accuracy of the fatigue prediction metric. One potential method to achieve that is to allow for a feedback mechanism whereby out-of-commission beams are collected and examined. Real fatigue features could then be compared to predicted fatigue both in terms of location as well as severity. This closed loop would then allow re-calibrating the weights based on real observation data. The result of this weighted operation [Figure 61](#) (right) is a prediction map of elements with high susceptibility to fatigue in dark red and others with a low susceptibility in dark blue. Reinforcement, for instance, could then be proposed at amounts proportional to the susceptibility rate; More reinforcement at elements

with high risk of fatigue and so on. It is important to note that this prediction map is not comprehensive. It does not take into account environmental and chemical factors that might have a greater effect on fatigue life than mechanical loading. It is only with the addition of extra sensors such as temperature and humidity sensors that the effect of these other parameters could be taken into account.

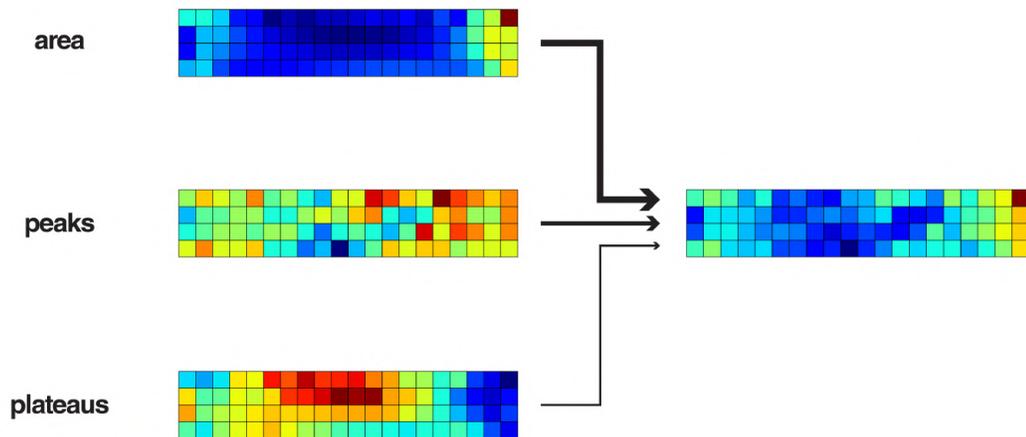


Figure 61: Fatigue prediction parameters across entire beam (left) and combination of weighted parameters (right).

DISCUSSION

5.1 IDENTIFYING APPLICATION SCENARIOS

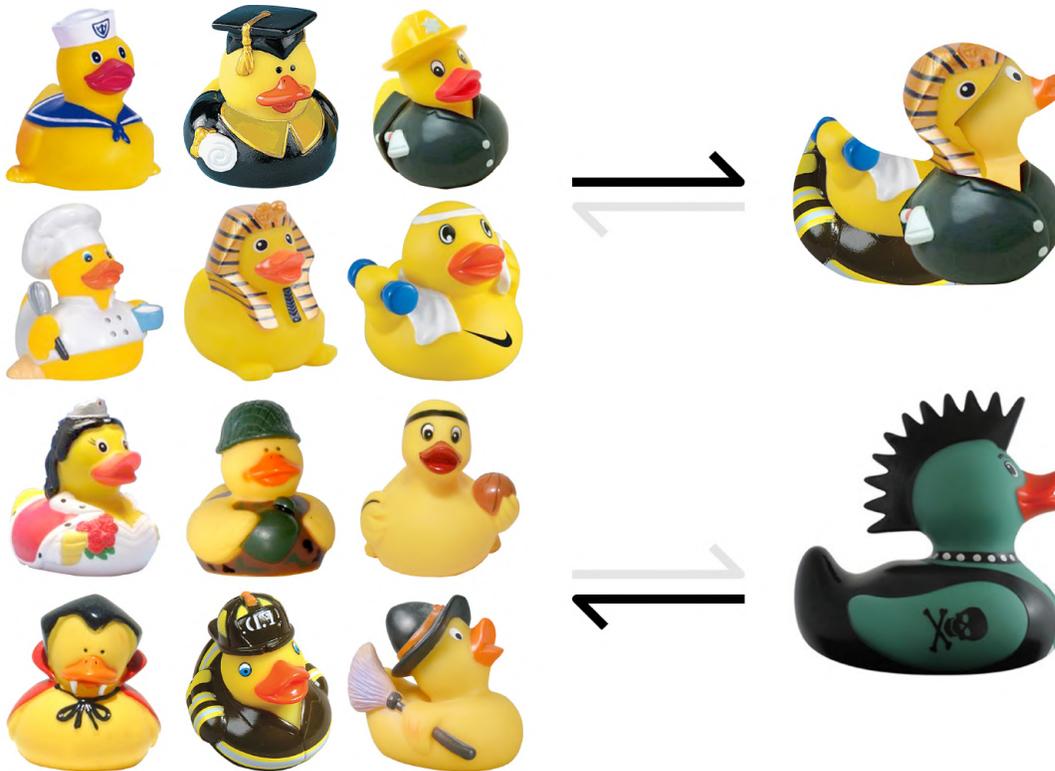


Figure 62: Hypothetical family of sensor-embedded ducks and potential suggested designs.

If we were to imagine a hypothetical family of sensor-embedded ducks [Figure 62](#). These ducks have been deployed into real world scenarios i.e. they are products that consumers have bought and used. The data recording thus started at the time of purchase. Let's assume that we now have deformation data from each member of the family across a number of years. We would then be able to access every model independently and define how success translates into data. What portions of the data represent ideal interaction modes? Which design data conveys minimum pressure readings and hence maximum comfort? Which predicts the least fatigue probabilities? Which has been interacted with and used the most? By identifying successful portions of each model, we can mix and match to mix and match eventually putting together a new generation of rubber ducks of superior performance. But this process is also reversible. We can propose a new radical disruptive rubber duck and, based on abstractions of geometry and material from existing data, we would be able to predict how this new design would

perform. This is one approach to creating a user-object interaction model. By allowing designers to provide design proposals as input, this model is then able to simulate interactions with the design based on historical data. Ultimately, the model would also autonomously generate design recommendations based on parameters set by users.

Potential applications for this methodology could be divided by results; whether they are immediate or require aggregation over a longer period of time:

- Immediate applications:

Quantified self applications: The emergence of wearables aims to quantify every single aspect of our lives from heart rate to number of steps taken [Figure 63](#). This requires the development of specialized sensors for each metric. A potential application would introduce the monitoring of material mechanical behavior to wearable technologies. A rubber band with sensitive strain sensors worn around the wrist could potentially replace an electronic heart rate sensor. Smart seats could detect unhealthy postures and alert user to correct it.



Figure 63: Early prototype of "Quantimetric Self-Sensing" apparatus, 1996 (body sensing apparatus with Digital Eye Glass for realtime display of ECG, EEG, EVG, and other body sensing apparatus output [9])

Smart user-aware objects: A material-specific Internet of Things framework that, based on interaction patterns or habits, would be able to identify users. For instance, sensor-embedded beds would learn the sleeping habits of users and would then refer back to it when attempting to identify the current user.

- Applications requiring aggregation over time:

User-object interaction models: Developments in the human-computer interaction and human factors and ergonomics are both human-centric. They map human interaction modes to computer software/hardware or to physical objects respectively. However they do not address human interaction in terms of object's mechanical behavior as ergonomics mainly focuses on how user's feel as oppose to how object's feel.

Physics-based deformation models A video game aiming to animate a real-world effects such as buckling could potentially utilize such a model. These deformations models can also be used in computer graphics and animation and augmented reality applications. The opens up the possibility of creating deformation models that help simulate the behavior of materials rather than relying on analysis systems based on approximation. simulations can now be more accurate relying on real data rather than based on approximations of elasticity moduli and other preset FEA parameters.

Real-time fatigue monitoring quantifying fatigue levels through accumulated stress. The aim is to propose an accelerated approach to understanding material failure. While traditional workflows rely on forensic evidence in assessing how and where materials fail - through physical cracking , real time monitoring of material behavior offers an immediate insight into potential failure scenarios. Other metrics could include humidity levels in materials, water content, temperature and exposure to light for light-sensitive materials.

Spatial Studies Analyzing objects to understand space. While translating from micro to macro, local behaviors are studied and global patterns are thus extracted i.e. aggregation of local behaviors to understand space use.

Within the health monitoring arena, a specific use case is identified for material analytics. This acts as an exemplary case study for problems which the internet of materials framework can potentially provide solutions for.

Prerequisites:

- A medical device, prosthesis or orthopedic device that patients interact with physically on a regular basis. This could be anything from beds, wheelchairs, splints for broken wrists to shoes or pillows used to treat specific disorders.
- A device where monitoring its mechanical behavior (deformations and strains) is beneficial to understanding its functionality.
- Data collected (and aggregated over a period of time) from monitoring this device is potentially useful for the medical practitioner, patient and device manufacturer alike.
- A device that experiences large deformations i.e. is made of rubber or some kind of material able to undergo large elastic deformations

Requirements:

- An existing design, preferably in any digital 3d format.
- A set of load cases based on initial guesses on how the patient will interact with the device.

Usage scenario:

A prosthesis embedded with strain sensors. Readings are logged at equal time intervals on the cloud and the equivalent deformation field is predicted and stored in a database. The medical practitioner can monitor the data and push notifications to the patient if the prosthesis is not being utilized appropriately. If abnormal data is discovered, this could trigger an emergency alert. The patient herself/himself can use the data in addition to continuously updated advice/pointers by the medical practitioner to assess their personal progress. This follows along with personalized health-care efforts and could also tie in the popular notion of "quantified self". The manufacturers of the prosthesis can get an accelerated understanding of fatigue - a load case that can't be predicted and hence modeled. They can then use the data to design improvements or even recall the prosthesis if its functionality exceeds a specific safety factor.

5.2 DATA-DRIVEN DESIGN VS. DESIGN INTUITION

The three major contributions of this research maybe be summarized as follows: **A.** A method for identifying the most optimum sensor placement locations given a designed artifact and expected loadcases **B.** A material analytics framework that is scalable, coupled with prediction power and plugs in to multiple sensor technologies and **C.** A new mode of design iteration that is data-informed. The heavy reliance on big data recently accompanied by the emergence of data science as a profession both triggered concerns towards the interpretation of data. How far should reliance on data go and does it restrict creativity? This is more of a design-biased concern. Other concerns, one might argue, pose a greater threat; Will data slowly erode science? It was back in 2008 that Wired magazine's editor-in-chief Chris Anderson predicted the end of theory and science in an article titled "The End of Theory: The Data Deluge Makes the Scientific Method Obsolete" [Figure 64](#).



Figure 64: Wired magazine cover, issue 16.07 [12].

Anderson argued that "our ability to capture, warehouse, and understand massive amounts of data is changing science, medicine, business, and technology" . And in fact it did. The scientific approach includes creating a hypothesis, testing it and thus validating or disproving it. With massive amounts of data, this approach is obsolete. The access to data replaces the need to prove a hypothesis. Instead of carrying out experiments, big data treats the entire world as one large experiment ground. And back to the design realm, can massive amounts of data combined with applied mathematics change the way we design? The fact is the day have, as demonstrated by software-based design fields such as web design and UI/UX. It was through this research that an attempt to replicate such data analytics practices into physical environments.

Data-backed design starts to raise crucial issues, one of which is the dichotomy between data and instinct. This has also been described as the tension between the science of design and the art of design. While instinct should be backed by reason, the temptation to refer to "scientific" data that is hard to avoid. The is, and will continue to exist, a burden of proof in design. Gut feelings and intuition are an accumulation of what is perceived to be an existing behavior or pattern, and thus it might be argued that data and instinct might not be too far apart. the Google+ button is a good case study to addressing this tension. While all style sheets have been already created, the team realized from research that red buttons received many clicks. The color red, however, was not consistent with the design palette across the project. This led to a compromise where red was used at main locations only.

It is thus important to separate data-informed from data-driven. While data might help drive solution towards a local optimum, finding the global optimum is not as straight forward. The data scope might not be far enough as to capture this global optimum. Since it is very difficult to quantify data reach and depth, interpretation of data at hand is crucial in making this distinction between data- "backed" and "informed". Data is often bias and some metrics are qualitative making them impossible to quantify and collect. It is also important to note that design is not an optimization problem, and thus data needs several layers of abstraction before it could be seamlessly integrated into the design thinking processes.

The data is eventually giving use mere recommendations. Its up to the authors how to interpret such data, leverage data feedback in the design process and potentially use it to validate design intuition.

APPENDIX

Follows is the code work flow for the whole process:

1. 3d models for both the specimen and pin used to create the displacement in .igs format.
2. Abaqus python routines a.generate the load cases, b.visualize the deformations and c. extract node, element and stress information.
3. Python routine to read stress/element information. Here optimum sensor locations are set.
4. Python routine to read node information and create look up tables for predicting the displacement field.
5. Arduino microprocessor code used to drive the material training apparatus.
6. Spark microprocessor code used to record sensor readings and push them to the cloud. A ([URL link](#)) could be queried at any time producing a JSON object as follows:

```
1 {  
  "cmd": "VarReturn",  
  "name": "myVol",  
  "result": "344916237674180",  
  "coreInfo": {  
6    "last_app": "",  
    "last_heard": "2015-03-31T11:26:58.463Z",  
    "connected": true,  
    "last_handshake_at": "2015-03-31T10:53:03.958Z",  
    "deviceID": "53ff72066667574852400967"  
11  }  
}
```

7. Web interface code for visualizing sensor readings developed in javascript. It is also used to pack the training data into .csv files to be used by the machine learning framework.
8. Python machine learning code that utilizes a linear regression model to predict coordinates of applied displacements and loads.
9. The [python web application](#) places the machine learning code on line. It receives the sensor JSON object from the spark microprocessor and, based on the object logging its data, will chose the corresponding regression model to use. Hence all files used to create the models are also placed alongside the application. A prediction is made in the form of another JSON object:

```
{
  "objectID": "53ff72066667574852400967",
  "sensorReadings": [3550, 3489, 3104, 3417, 3345],
  "time": "2015-03-31T11:44:50.247Z",
  "predicted": {
    "predictedX": 14.58,
    "predictedY": 0.87,
    "predictedP": 38.96,
    "predictedXdisp": [-0.0, -0.0, 0.00965, 0.00965, 0.01918 ...],
    "predictedYdisp": [-0.0, -0.0, -0.00064, -0.00064, -0.00103 ...]
  }
}
```

10. Prior to visualizing the object within a web application, two steps are taken: **a.** the model is converted from .obj to .js allowing it to be read by THREE.js, a WebGL javascript library. The orientation of the .obj should match that of the .igs used initially as input to Abaqus. **b.** Matching the vertex numbers of both the original simulation model and the newly created .js. This is done using a python code and an ordered list is then copied into the Node.js web application explained hereafter.
11. The **Node.js web application** is responsible for saving the JSON object under item 9 onto a MongoDB database. It is also responsible for pinging the python web application every 5 seconds to receive new displacement predictions. These are then passed onto the THREE.js library updating the nodes of a digital mesh representing the beam.

BIBLIOGRAPHY

- [1] *Experimental stress analysis and measurement*. Charles E. Merrill Books Inc., Columbus, Ohio, 1964.
- [2] *Experimental stress analysis*. McGraw-Hill book company, 1965.
- [3] M. Alba, F. Roncoroni, L. Barazzetti, A. Giussani, and M. Scaioni. Monitoring of the main spire of the duomo di milano. Politecnico di Milano, Dept. of Building Environmental Science and Technology.
- [4] A. Ali, M. Hosseini, and B.B. Sahari. A review of constitutive models for rubber-like materials. *American J. of Engineering and Applied Sciences*, 3(1):232–239, 2010.
- [5] Bernd Bickel, Moritz Bacher, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. Capture and modeling of non-linear heterogeneous soft tissue. In *SIGGRAPH '09 ACM*.
- [6] Pedro Domingos. A few useful things to know about machine learning. Department of Computer Science and Engineering, University of Washington.
- [7] P. Ghosh, A. Saha, and R. Mukhopadhyay. Prediction of tyre rolling resistance using fea. *Constitutive Models for Rubber*, 3:141–145, 2003.
- [8] Can He, Jianchun Xing, Juelong Li, Qiliang Yang, Ronghao Wang, and Xun Zhang. A combined optimal sensor placement strategy for the structural health monitoring of bridge structures. *International Journal of Distributed Sensor Networks*, 2013(820694), 2013.
- [9] Steve Mann. Humanistic computing: "wearcomp" as a new framework and application for intelligent signal processing. *Proceedings of the IEEE*, 86(1), 1998.
- [10] W.V. Mars and A. Fatemi. Factors that affect the fatigue life of rubber: A literature survey. *Journal of Rubber Chemistry and Technology*, 77(3):391–412, 2004.
- [11] Peter Mell and Timothy Grance. The nist definition of cloud computing - recommendations of the national institute of standards and technology. Special Publication 800-145.
- [12] Misc. Wired magazine: Issue 16.07, 2007. URL <http://archive.wired.com/wired/issue/16-07>.
- [13] Misc. Amiens cathedral, 2015. URL http://en.wikipedia.org/wiki/Amiens_Cathedral.
- [14] Misc. Open cv template matching, 2015. URL http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

- [15] Misc. Understanding fatigue analysis, 2015. URL <http://www3.eng.cam.ac.uk/DesignOffice/cad/proewild3/usascii/proe/promec/online/fatigue.htm>.
- [16] Misc. Fatigue(material), 2015. URL [http://en.wikipedia.org/wiki/Fatigue_\(material\)](http://en.wikipedia.org/wiki/Fatigue_(material)).
- [17] Misc. The complete guide to a/b testing, 2015. URL <https://vwo.com/ab-testing/>.
- [18] Misc. Spy strategies for web design: Analytics, 2015. URL <http://www.dtelepathy.com/blog/design/analytics-for-web-design>.
- [19] Misc. Json, 2015. URL <http://en.wikipedia.org/wiki/JSON>.
- [20] Misc. Nosql, 2015. URL <http://en.wikipedia.org/wiki/NoSQL>.
- [21] Tom M. Mitchell. The discipline of machine learning. CMU-ML-06-108.
- [22] Stephen Murray, Robert Bork, and Robert Mark. The openwork flying buttresses of amiens cathedral. postmodern gothic and the limits of structural rationalism. *Journal of the Society of Architectural Historians*, 56(1):478–493, 1997.
- [23] G. Stoilov, V. Kavardzhikov, and D. Pashkouleva. A comparative study of random patterns for digital image correlation. *Journal of Theoretical and Applied Mechanics*, 2012.
- [24] Yuta Sugiura, Gota Kakehi, Anusha Withana, Calista Lee, Daisuke Sakamoto, Maki Sugimoto, Masahiko Inami, and Takeo Igarashi. Detecting shape deformation of soft objects using directional photorefectivity measurement. *UIST*, 2011.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX :

<http://code.google.com/p/classicthesis/>

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

<http://postcards.miede.de/>

Final Version as of May 15, 2015 (`classicthesis` version 2.4).