Title: Design and User Experience in Software for Medical and Psychological Research

Russell Pensyl, Professor (ret.)
pensyl.com | Northeastern University | Boston, MA, USA

Keywords:
Vision/sensor; fusion, facial recognition; biometric data capture; emotion response detection; ultrasonic tracking for positioning and localization; mixed reality; physical computing.

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems –Artificial, augmented, and virtual realities; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual reality

This paper draws from content in the upcoming text, "interAct - history of interaction design, experience and interactive art."

1. Introduction:
Leveraging on several years of research and development in facial recognition, emotion detection, sensors, and vision systems, we produced a series of works that merge art, science, technology, and devices. Over the last eight years, our teams have designed software and mediated interactive experiences for research in psychology, and in medical research. This paper presents observations on the stakeholder interactions in planning of these software and hardware systems that may prove useful for medical and scientific researchers and developers of systems used in medicine.

First, we will present the work the led up to the creation of the software, Emota v4.0, and the considerations of how the design, specification, planning and implementation of this software occurred working with software developers and psychologists. Second, we will present the design of a new software used in medical research using vision and sensor systems to collect biometric data that is used in scientific and medical research.

2. Background
When considering the design of software for medical devices for both research & medical treatment, the stakeholders associated with development include individuals with diverse backgrounds & knowledge. Inevitably, knowledge & experience of these groups have only partial overlap. This paper is written to raise awareness of the interface design, user experience, & software design & development stakeholder's contribution & what area of expertise, they bring to a group of experts in medicine, psychology, & other participants in the projects.

My work as a software design & device development in the creation of gesture recognition, gait recognition, & facial recognition, emotion detection software, specifically Emota v4.0. Other earlier work includes positioning and localization using vision and ultrasonic sensing, vision and gyroscopic accelerometer sensor tracking of humans in real-time real-world environments for mixed and augmented reality, and vison detection of gesture recognition. Second, we will present the design of a new software used in medical research using vision & sensor systems to collect biometric data that is used in scientific & medical research. The images in Figure 1 depict a prototype electro-stimulation headset designed for a start-up company with researchers from MIT, Harvard & Boston Hospitals. This project requires, among other things, ergonomics, physiology & as well as software design for control & management of the electro-stimulation as well as data collection post priori to ascertain treatment effectiveness.

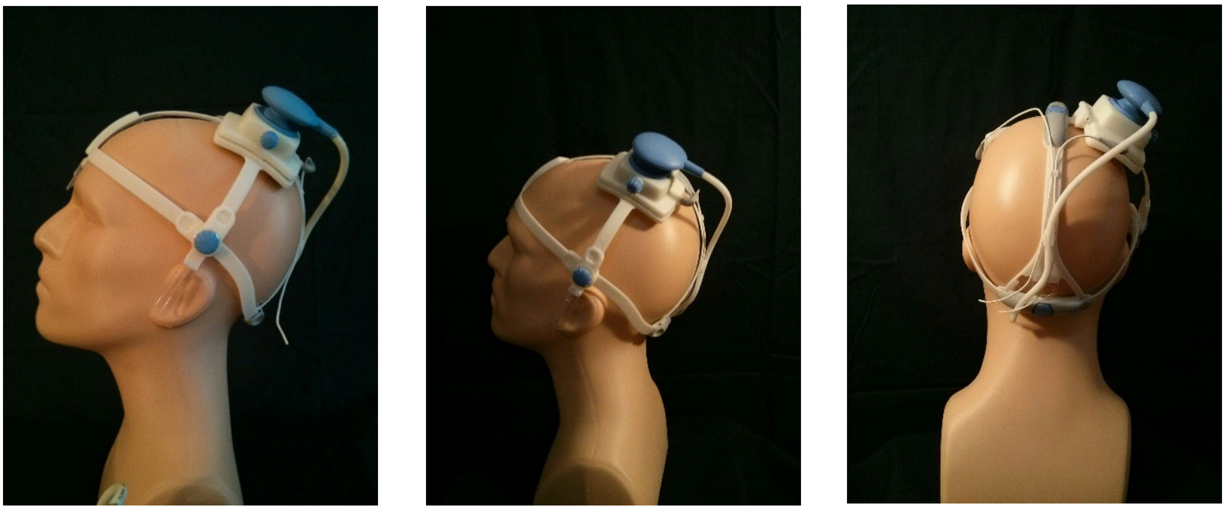
 [1]

Figure 1: Prototype 3D printed electro-stimulation headset, Highland Instruments.

3. Analysis Methods in Interaction Strategy and User Interface

One of the considerations in design of these systems is how the specification, planning & implementation occurs with collaboration with software developers, medical research, practitioners & psychologists. There are a few basic assumptions that can be made in regard to all digital media artifacts, past, present, and future. There are stakeholders in the development of any artifacts. Any enterprise creating an app, device or service has a basic frame that is related to the primary goals of the enterprise. In capitalist societies, this will be profit and return on investment. While this area for framing can be overlooked in the analysis, doing so leaves a very large hole in the understanding of the artifact – since the artifact exists specifically in the culture and society.

For user experience designers to effectively contribute to functioning software or systems, they will employ a set of techniques that are analytical & process oriented. For analysis, the techniques of Constraints & Affordances Analysis, Frame Analysis, Formal Analysis, and Organizational Culture as tools are often employed. For stakeholders in medical practice, research, data analysis experts, & software engineers, they should understand terms & methods employed. However, the author contends that Design Process as an Analysis Tool may prove equally effective in framing phasic segment of specification, design, testing and implementation of interactive artifacts in psychological research, medical practice and research. Design process has been worked and experimented with to maximize user participation and engagement. In fact, user experience and experience design has resulted from efforts by luminaries in Human Centered Design such as Don Norman and Tim Brown of Stanford University's D-School. Brown, et al, describes this as "Design Thinking." These scholars, among many others, sought to find ways to increase product engagement through understanding just what the consumer is seeking to achieve and what the desires and goals are. In relation to the digital artifacts, one aspect of the work in the field of "Human-Computer Interaction" blended cognitive psychology with engineering resulting in effective interface and user interaction strategies.

To become experts in the usage required specific knowledge and mannerisms that were idiosyncratic to that specific device. Different manufacturers employed different methods, and while the end result may appear the same another artifact, a printed document used in a business or corporate environments the methods of creation depended on how that tool was developed, and the underlying computational structures of the assembly language and machine code. These depended on the architectures of the chipsets in the device.

Erving Goffman, credited with coining the term in his book Frame Analysis,[2] understood the idea of the "frame" to mean the culturally determined definitions that allow people to make sense of objects and events. For Frame Analysis to be an element of ethnographic research, data is collected in identifiable chunks of social behaviour, in order to understand the frames that participants use to make sense of the behaviour. Frame analysis can be adapted to specification and parameters for interactive tools for medical, by maintaining a focus on the research mission and determining specifications within this strict purview. Each

---

[1] Figure 1: Images: courtesy the author, Electro-Stimulation device, Highland Instruments.

[2] Erving Goffman, Bennett Berger Frame Analysis: An Essay on the Organization of Experience, 1986, Northeastern University Press; ISBN-10 : 093035091X

tool used, while affording other potential uses, when used for medical or psychological data collection, is most likely a solution to prior diagnostic or treatment practices. This provide the frame of reference for analysis that includes device behavior, as well as behaviors of practitioner that employs the device. This can be extended to physical cues and behavior of patients as well.

Design Process, for example, asks the user experience & interface design team to follow a set of stages that inform the design of the software coding as well as the interaction strategies of the users of such software & systems. One of these stages is also a frame analysis. Typically, the design process stages are:

- Frame the problem.
- Understand genre features.
- Refine familiar elements.
- Define usability, durability, aesthetics, both visual & tactile.
- Reframing to adjust behaviors of devices, tools, methods to fit with established understanding.[3,4]

Framing and Re-Framing

Reframing takes commonly understood objects and forms, using a new technology, allows the same conceptual framework, or "mental model" to be applied. This maintains a consistent and accessible mode of action for goals achievement. The use of the "tabs" in the browser, stacking multiple webpages carries the same cognitive model from the file folder. The cognitive model, or pattern is re-framed to allow the participant to undertake commonly understood actions and practices within the virtual environment of digital media and computing technology. Almost every software developed has been the solution to an earlier problem or to make some common tasks easier to undertake. In this each software is re-framing the actions, the process and possibly the results to enable the user to be more efficient and achieve more in the work at hand.

In a survey of how interface designers work, there will be some consistencies in the phases of their process. One is to start by considering the specifications for the work at hand. Following this will be an open-ended period of free-association, or possible ontological mapping of similar concepts, words, phrases, ideas, products, and other cultural artifacts. In the case of medical devices, there are often long used devices and practices that frame the work, the tasks and whatever analysis may result. The "framing" is much closer to machine design and engineering than social media to other interactive applications used today particularly those in social media or phone apps.

Design process:		Analysis:
▪ Frame the problem. ▪ Understand genre features of similar artifacts. ▪ Refine familiar elements. Usability, durability, aesthetics, both visual and tactile. ▪ Reframing.	⇨	▪ What problems did the artifact attempt to solve? ▪ Understand genre features of similar artifacts. ▪ What are the familiar elements? Usability, durability, aesthetics, visual and tactile. ▪ What earlier practices were reframed.[5]

Table 1: Reframing design process as an analysis tool

Analysis of Constraints and Affordances

Using the example of the earliest artifacts created by early hominids such as flaked stones used for cutting, include first the recognition of the affordances of the object. Affordances are found by understanding the material of the object. A stone tool used for cracking nuts, or bones to get to the food inside indicates knowledge of the comparative hardness of the material of the stone versus that of the nut, or the bone. A flaked stone has a sharp edge, that sharp edge in use affords cutting and scraping of skin and flesh. These artifacts can be first assessed by the materials – hardness, durability, portability, sharpness of flake edges. After this assessment of the formal materials, there are the formal aspects of the tools use. To understand

[3] Kelly, Nick; Gero, John (2021). "Design Thinking and Computational Thinking: a dual process model for addressing design problems". Design Science. 7: e8. doi:10.1017/dsj.2021.7. S2CID 233317330

[4] Cross, Nigel.2001, Design Cognition: Results from Protocol and other Empirical Studies of Design Activity, in C. Eastman, M. McCracken and W. Newstatter (eds.) Design Knowing and Learning: Cognition in Design Education, Elsevier, Oxford, pp. 79-103. ISBN 0 08 043868 7

[5] Pensyl, W.R. "interAct" - history of interaction design, experience and interactive art, Design Process as an analysis method.

both the materials and use of a tool, or device, one needs to analyze the "constraints" of the materials, constraints of the methods of production, constraints of the possible uses. In an effective analysis of any artifacts, these constraints will inform the understanding of the object and assist in making conclusions of the cultural usage or its "form."

Design of specific tools are constrained by material properties that enable what kinds of tool can be created. The afore mention knapped stone tools in basalt are still known today as providing a keener edge than the finest steel. Production of a tool is also constrained by the materials and the methods needed for its production. This is a technological constraint. Other constraints such environmental constraints are those where the use and behaviors may be constrained by external factors. The use of some tools is impossible in dry conditions, and possible in rainy or wet conditions. There are constraints that are cultural. Formal behaviors within culturally codified activity are constrained by accepted behaviors.

There are volumes written on the subject of affordances and the study of how one understand items in an interface, in a tool, in environmental conditions, that allow the human to just how it is that the encountered items can be used. J. J. Gibson in The Ecological Approach to Visual Perception states, "The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill." [6]Norman and Kress stress that beyond material and technological constraints, both constraints and affordances have social and cultural potentialities. As Kress stated, there are 'modal affordances' that are often multi-layered and 'affordances' are not a matter of perception, but rather refers to the "materially, culturally, socially and historically developed ways in which meaning is made with particular semiotic resources."[7]

Assuming that specific tools and methods in the medical field are exempted from notion of constraints and affordances, it so miss a critical aspect of how humans understand, perceive the world and operate within it. Humans operate by pattern recognition, however faulty that recognition is. An accepted pattern of behavior is a cultural constraint. In the case of navigation of large and dangerous machines, the use of a cultural constraint is imperative. Design researchers that explore medical environments, and interpersonal relationships between practitioners, treatment facilities and patients, such as Miso Kim, who takes a humanistic approach to design. Her holistic approach looks at more of the intangible elements in relationships, and this branch of design is referred to as service design. Kim's practice in service design focused on enhancing the experience of patient dignity. It is implicit that both societal and cultural constraints are in the practitioner/patient relationship.[8]

Case: Highland Instruments Electro-Stimulation Device
In the case of the prototype electro-stimulation device, and other that are ground-breaking technology and tools, the framing for these devices may be difficult, as well as determining constraints other than material and technological. However, the affordances arise from physics and physiological and sometimes environmental factors. In the design of this device, researchers studied and tested many iterations of sensor, and electronic beam angles and placements. The device is similar to how an x-ray is used in diagnostics, or radiation bombardment is used in treatment. Thus, there are contextual and environmental constraints that must be considered. Coupling the electro-stimulation with sensor to detect immediate physiological responses or in subsequent treatment regimens enabled the practitioner to use the device as a tool with specific intent.in this case the device control system is where the application of user experience and interface design are employed. Custom software was developed to allow the clinician to use the tool in treatment, the communication infrastructure, could be managed in a wireless method, but in this case, a wired connection via RS-232, and USB proved more effective and reliable. When considering the interface, factors such as expert knowledge and adaptability to the control system are imperative. Some clinicians prove to be more adept at operating the device, under direct and even indirect supervision of a medical doctor. This is not dissimilar to technicians operating data collection devices such as X-Ray, ECG/EEG/EMG, MRI, CAT and PET scanning.

To be blunt, much technology use medical diagnostic and treatment functions in a manner that is akin to machine control of earlier tools and technology. These devices are, as in the ultrasound device in Figures 2, 3 and 4,[9, 10] are complex and cannot simplified. This is the case with most of tool interaction and interfaces

[6] J. J. Gibson, The Ecological Approach to Visual Perception
[7] https://multimodalityglossary.wordpress.com/affordance/
[8] Miso Kim, An Inquiry into the Nature of Service: A Historical Overview, Design Issues (2018) 34 (2): 31–47.
[9] Figures 2, 3, Image credit, courtesy of the author.
[10] Figure 4, https://manufacturingsolutionsgroup.com/machining-processes-explained/

that designers are tasked to create. They are in many ways like a complex interface for controlling a CNC milling machine for mechanical parts – very powerful machines for dedicated tasks.

Figure 2, 3, 4: Complex Machine Interfaces

It should be noted that software systems that control complex machinery and process generally have complexity of the interface corresponding to the power and range of possible actions and resulting work. In graphic arts, media CAD, and animation software, the learning curve is extremely steep, and the time to master such software is gauged in years, not months or weeks. Even with sophisticated user interfaces the processes being controlled are even more complex that is visible on the surface of the interface. In the paper Big Data: A Classification Of Acquisition And Generation Methods, Nanjappan, et al list, some examples of post-WIMP interaction styles: virtual, mixed and augmented reality, tangible interaction, ubiquitous and pervasive computing, context aware computing, handheld, or mobile interaction, perceptual and affective computing as well as lightweight, tacit or passive interaction.[11]

Of these, tangible interaction, context aware computing and perceptual and affective computing are directly related to device interfaces and interaction strategies used for biometric and medical data capture. Nanjappan, et al further state, “We believe that all of these new interaction styles draw strength by building on users’ pre-existing knowledge of the everyday, non-digital world to a much greater extent than before. They employ themes of reality such as users’ understanding of naïve physics, their own bodies, the surrounding environment, and other people. Nanjappan, et al give these interfaces a classification of Reality Based Interfaces. Interaction strategies that mimic reality alone is not enough. For Nanjappan, et al, Reality Based Interfaces principles should be traded against other considerations. These include:

- Expressive Power: i.e., perform a variety of tasks within the application domain
- Efficiency: perform a task rapidly
- Versatility: perform many tasks from different application domains
- Ergonomics: perform a task without physical injury or fatigue
- Accessibility: users with a variety of abilities can perform a task
- Practicality: practical to develop and produce

Anecdotally, the author contends that aside from widely used software applications and digital computing devices, these interface and interaction strategies indeed, also represent the “state of the art” in devices and systems in medical diagnosis and treatment. Additionally, Nanjappan, et al, characterize date collected using the various data type and sensor data collection as either structured or unstructured.

Structured data are usually defined with fixed attributes, type, and format—for example, records in a relational database are generated according to a predefined schema. Compared to unstructured or semi-structured data, processing of structured data is relatively simpler and more straightforward. This type of data can be generated by people, machines, and sensors.

> (1) Human-generated structured data: the data are created under explicit human involvement using some interaction mechanisms, e.g., data generated through human-machine interface devices like mouse input data and click-streams.

[11] Big Data: A Classification Of Acquisition And Generation Methods Vijayakumar Nanjappan, Hai-Ning Liang, Wei Wang, Ka L. Man

(2) Machine-generated structured data: the data are created automatically by a computing device without explicit human interaction, e.g., Web log data.

(3) Sensor-generated structured data: the data are generated by embedded fixed or moveable sensors, e.g., sensor data from smartphones and smart meters.[12]

Unstructured data are the opposite of structured data, without a predefined data model. Some common examples include text, images, audio, video, and streaming sensor data. Unstructured data are one primary source of big data and are much more challenging to process compared to structured data.

4. Facial recognition and data collection.

In the case of the software that the author has designed and implemented, we collect a combination of structured and non-structured data. Facial recognition and emotion detection uses what the author characterizes as "small data," that results in a small range numerical classification – biological sex, some personal attributes, and classification of facial expression in the range of seven classical emotion states, as define by Paul Eckman.[13] These results are used to invoke modifications media, or environmental spaces. In fact, in the iterations of the software, HiPOP and Emota, there is no actual interface at all for participants. The software simply collects the data and invokes the response without the knowledge or actions of the audience or viewer. There is a more complex control system behind that which is seen by the audience. This can be complex but is used only to instantiate the software and system that then runs autonomously. These software systems are fully described further on.

Examples of how structured data may be arrived in facial recognition and emotion detection can be found in our software Emota v3.0. This software techniques relies first on Viola and Jones's Viola and Jones Open CV Haar-like features application[14, 15, 16]and a "feret" database[17] of facial image and support vector machine (LibSVM)[18]. pre-trained Haar Classifiers to locate facial features. Before feature extraction, lumen normalization was adopted to detect facial part of the image such that light conditions have less effect to the feature extraction process. Numeric features are extracted through convolutions with a set of pre-calculated Gabor filters called Gabor Bank. Gabor filters are implemented to derive orientations of features in the captured image using pattern analysis, directionality distribution of the features. Using Gabor filters increases accuracy of the anchor points derived in the elastic bunch graph matching.

Operations of elastic bunching graph include graph matching, graph pruning, and adding sub-graphs from either an image or an xml file. Elastic bunching graphs applies convolutions of certain areas of images using all filters in the Gabor Bank. This results in a list of anchor information for all anchor points, where each anchor information contains a list of all convolution results.[19] This work was tested in the creation of a generative musical composition where the detection of emotion state via facial expressions alters tempo, timbre and pitch in a live unscripted musical stream.[20]

The author's teams went through two earlier iterations of emotion detection, with successful results. One main consideration has been a "universal" method that does not require specific training on a single face and will still provide accuracy that is usable in a practical application. The tradeoff is manly this, accuracy vs universality. In the case of the first facial recognition software the author developed, HiPOP,[21] we successfully detect biological sex based on cues in facial features, the tradeoff was accuracy vs speed. The longer time to scan and compare features results in higher accuracy. In Emota v4.0 we explored using a local algorithm and the Microsoft cloud based FaceAPI in a custom media control system running on a Raspberry Pi, 5 MCP23017 GPIO extender ICs, 5 sixteen bank relay clusters, to control patterns in an array of twenty-four traffic lights.[22] Microsoft regrettably limited access to Azure AI Vision API to selected partners.

---

[12] Ibid. Big Data: A Classification Of Acquisition And Generation Methods Vijayakumar Nanjappan, Hai-Ning Liang, Wei Wang, Ka L. Man

[13] Ekman, P., (1999), "Basic Emotions", in Dalgleish, T; Power, M, Handbook of Cognition and Emotion, Sussex, UK: John Wiley & Sons,

[14]. Viola, P., & Jones, M. (2001). Robust real-time object detection. Paper presented at the Second International Workshop on Theories of Visual Modelling Learning, Computing, and Sampling

[15] Bradski, G. and Kaehler, A., (2008). Learning OpenCV. OReilly.

[16] Burges, C. J.C., (1998) A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121-167

[17] http://www.nist.gov/huma nid/colorferet

[18] Ibid, Burges, C. J.C., (1998) A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121-167

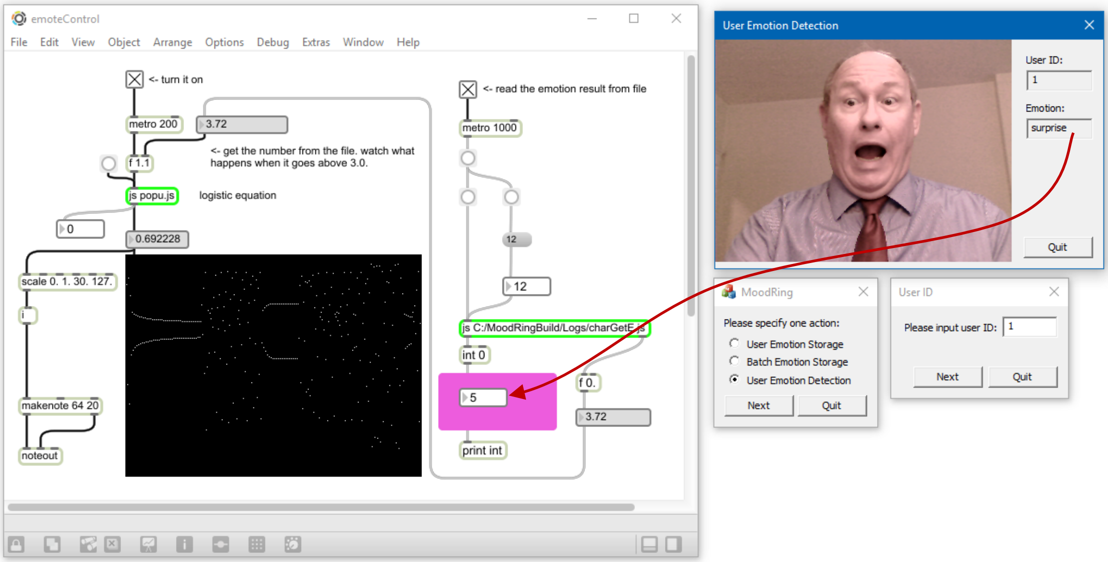
[19] 2015 Pensyl W., Min, X., Song S., Facial Recognition and Emotion Detection in Environmental Installation and Social Media Applications Encyclopedia of Computer Graphics and Games https://doi.org/10.1007/978-3-319-08234-9 = 978-3-319-08234-9

[20] Cycling74 MAX/MSP

21 file:///D:/pensyl/p51Hi.html

22 file:///D:/pensyl/p51RGW.html

Currently we use the fall back method that is a less sophisticated and less accurate custom face/emotion detection that runs entirely on the Raspberry Pi.



[23]

Figure 5: Screen shot of Emota v3.0 applied to musical composition in Cycling74 MAX/MSP

One of the most advanced systems for emotion detection is develop by a team of psychologists and computer scientists at the Machine Perception Lab at University of California, San Diego. Emotient was bought out by Apple in late 2016 and much of the history and development can only be found in the "wayback machine" archives.[24] The core software use similar techniques in facial recognition and emotion detection the author's Emota v3.0 software solution but is far more adept at the recognition response. As stated in the description of "Auto FACS Coding," the output of the face detector is fed directly to the facial expression analysis system (see Figure 1). First the face image is passed through a bank of Gabor filters at 8 orientations and 9 scales (2-32 pixels/cycle at 0.5 octave steps).


[25]

Figure 6: Screenshot of the Emotient FACET SDK Demo Video

The filter bank representations are channeled to a statistical classifier to code the image in terms of a set of expression dimensions. We conducted a comparison of classifiers, including support vector machines (SVM's), Adaboost (Freund & Shapire, 1996), and Linear Discriminant Analysis (Littlewort et al., in press, Bartlett et al., 2003). SVM's were found to be very effective for classifying facial expressions. Recent research at our lab has demonstrated that both speed and accuracy are enhanced by performing feature selection on the Gabor filters prior to classification (e.g. Bartlett et al., 2003). However, the methods used affords detection of what Bartlett describes as "micro-emotions." This is an extremely sophisticated

[23] https://pensyl.com/p51Emot.html | Cycling74 MAX/MSP
[24] https://web.archive.org/web/20151116125123/http://www.emotient.com/ | https://web.archive.org/web/20151219062637/http://www.emotient.com/products
[25] https://www.youtube.com/watch?v=9DmiqU8Cxjs&ab_channel=Emotient

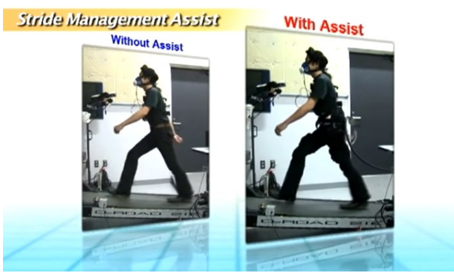
algorithmic approach that enable researchers to assess possible emotion states even in subjects that may attempt to hide their emotions. (sic) [26]

At the time of writing, likely, the most sophisticated emotion detection today is found in Microsoft's Azure AI Vision API. This is a cloud-based software solution that allows participants to upload images and video stream and the results are delivered back to the participant software application. Microsoft's earlier iteration, Emotion API takes a facial expression in an image as an input and returns the confidence across a set of emotions for each face in the image, as well as bounding box for the face, using the Face API.[27] The confidence is given back in a numerical scale, that could be classified in one of Eckman's, classic emotion states.[28] Earlier iterations of the Face API have been scrubbed from Microsoft Azure site, but can be found on the waybackmachine.[29]

The design of such software systems and the user interaction strategies may have serious implications for researchers. The goals of exactly what the expected outcomes in data collection require stringent specification. The are limitations to every sensing system that include frequency of data point collection. In terms of facial recognition, the speed of typical video capture speed of 30 or 60 frames per second is likely adequate for assessment of facial expression, body positioning. However, is researchers need to couple other data capture, such as EEG, or other sensing devices, some limitations of the computer system and operating system may cause issues in data capture and any subsequent analysis. If software based real-time categorization of information collect the timing and synchronization of the data points become critical. Given advances in artificial intelligence agents designed to process and provide analysis, these concerns will be important considerations. Timing and synchronization of various sensing devices will be presented in depth in a later case study presented in this paper.

In terms of the user interface, there is little concern as the data capture can may affected by user performance or expertise. In many cases, the user interface for such systems is a simple instantiation of the system. This is particularly the case with only one or two sensing devices. If multiple sensing devices are used, then some considerations of user interface structure, affordances and action flow need to be addressed in the user interaction specifications.

The ideal circumstance for such system may be as what Janet Murray refers to the computing functioning as a passive companion[30] that can collect data unobtrusively. We can see such systems being commercially employed by enterprises such the shopping experience recently rolled out at Amazon's Whole Foods when the costumer scans their palm on entry, and each item selected is automatically scanned and catalogued in the "electronic shopping cart" and payment is automatically processed without any over actions on exit. If one considers the use of RFID sensors embedded in consumers arms, use for bar tabs, and direct billing as has been around for over a decade in some European cities, the possibility of using real-time tracking and data capture is obvious.[31] Another interesting example of such systems was deployed by Sergio Albeic, sponsored by Lexus where automobile sensor data was captured in real-time to paint a generative portrait of the driver on the tablet screen.[32] The characteristics of the image reflected the manner of the drivers' actions. Aggressive driving results in a cruder image, while gentler and steady driving results in softer imagery.



Extrapolating the possible applications of data collection, the methods of data collection, and analysis easily leads to solutions for behavior assessment – and on a darker side behavior modification. Medical applications for muscle, brain, and nervous system stimulation are becoming common. Much of this was foreshadowed by artists such as Stelarc experimenting with embedding electrodes in the body[33] to allow remote "tele-operation" by persons in far-away locations to force involuntary muscle movements in artistic/theatrical performance art. *Fig7*****

[26] Bartlett, M.S., Littlewort G., Movellan,J. Frank. M.S. | https://inc.ucsd.edu/mplab/79/
[27] https://web.archive.org/web/20190110164611/https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview
[28] Ekman, P., (1999), "Basic Emotions", in Dalgleish, T; Power, M, Handbook of Cognition and Emotion, Sussex, UK: John Wiley & Sons,
[29] Ibid. https://web.archive.org/web/20190110164611/https://docs.microsoft.com/en-us/azure/cognitive-services/face/overview
[30] Janet Murray, Inventing the Medium - Principles of Interaction Design as a Cultural Practice Chapter 12, The MIT Press
[31] https://pourmybeer.com/beverage-wall/
[32] https://www.sergioalbiac.com/wall/the-generative-identity-of-walter-vanhaerents.html
[33] http://stelarc.org/_.php#page/7 | http://stelarc.org/_.php#page/8

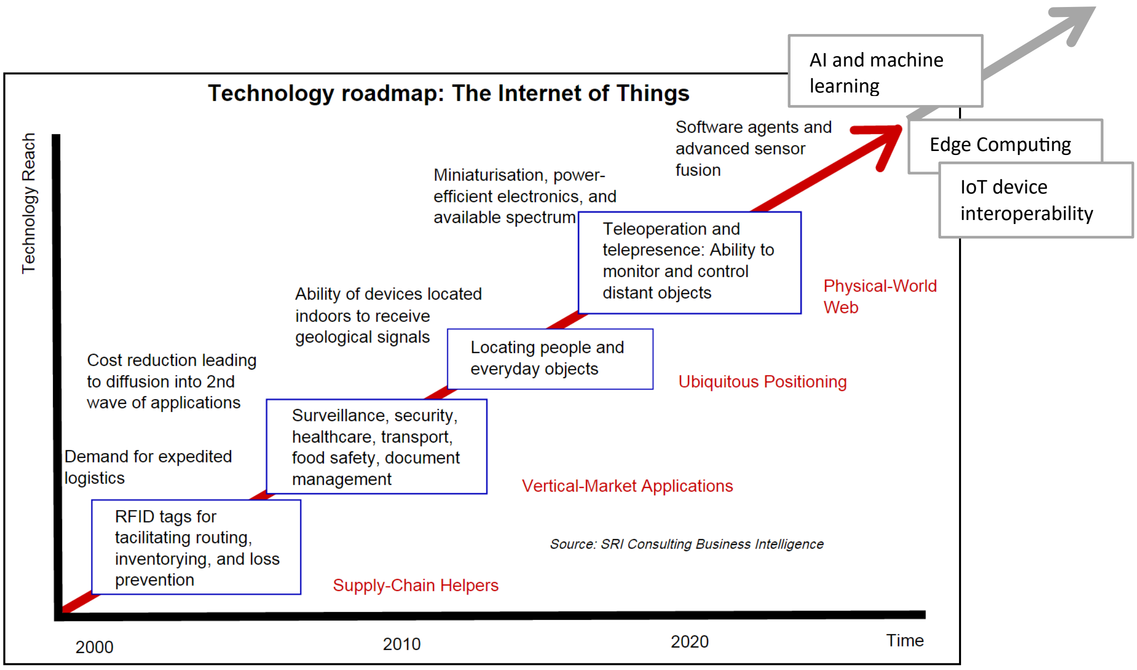


Figure 8: The state of the Internet of Things circa 2025

Given the current state of the "internet of things," we are well past software agents and advance sensor fusion as indicated in Figure 8. [34] The author AI search agent, gives an indication of the next few years:

- Edge computing where data collection, processing and analysis occurs closer to the source, reducing latency and improving real-time decision-making
- AI and machine learning enable software agents to predict maintenance, anomaly detection, and improved decision-making. The implication is that this can happen without human intervention or even cooperation.
- The Internet of Things Consortium and the Open Group are developing standards and guidelines for IoT device interoperability – similar to what Berners-Lee did for sharing of data content via a standardized mark up, hypertext transfer protocol and uniform resource location.

In terms of ease of use, creation of standardized protocols, and interface-less software agents is perhaps the best for the average consumer, where an intelligent system collects data and provides a meaningful passive action. However, as the tools used become more complex and powerful, complexity of interface interaction increases correspondingly to the power of the machine, or system. The expectation of expert knowledge of the operator to successfully managed processes for the intended result is still here, and in some cases, increases the learning curve and specialization of that expert knowledge. For advanced software in media – editing 3D modeling, animation, image editing, and in CAD/CAM, or mathematics and science the complexity of interface has not become simpler over time. In most software applications, the complexity of continues to increase.

This leaves researchers in many fields in situations where vast knowledge software engineering advanced coding and hardware specifications is required - or - to work in teams with experts in other domains. The crux of this paper is aimed at cases where there are several stakeholders in the development of software tools that use a variety of sensing and data collection for research.

Complex mathematical formulas inevitably require breaking the formula in the sections that can be processed in the native code of the API. Each language has specific syntax and structure. A formula in C+ has to be rewritten in C# or Python. These idiosyncratic methods become more complex with the number of parameters needed to gather useful information for research purposes. Working in team-based environments, communication on what the expected outcomes of the coding and processing of raw data is sometimes confusing. A software engineer may prefer to write in Python. However, the development of a system requires the use of devices where the Application Programming interface (API) is written on C#. it is unlikely that the device manufacturer will rewrite the API to suit the application a researcher is developing.

[34] https://en.wikipedia.org/wiki/File:Internet_of_Things.svg | Public Domain, Title 17, Chapter 1, Section 105 of the US Code

Even a simple set of mathematical functions, such as for calculating angles in a trapezoid, seen in Figure 9[35] that can be easily processed by hand, becomes 80 lines of code in C# as seen in Figure no 10.[36] It is not a straightforward simple translation, due to the syntax and idiosyncrasies of the software language. The formula used in our Emota v3.0 is seen in Figure 11.[37]

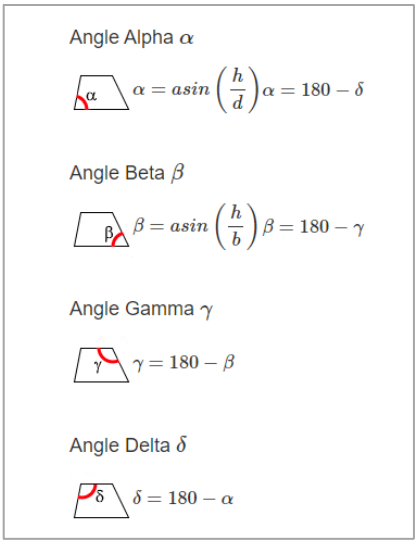


Figure 9: Function for calculating the trapezoid angles.

Anecdotally, in a project where the author developed a large number of interactive tutorials for a population college level textbook in Chemistry, illustrates perhaps the worst case scenario. The consulting professor working on the book, stated something to this effect: "here is the formula for the "Equilibrium of NO_2 and N_2O_4" the module required to graph the resulting change concentration of these chemical agents over time. [38]

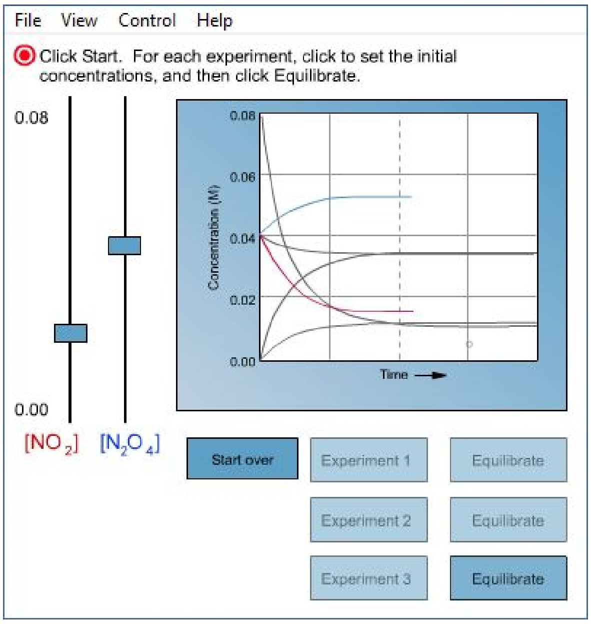


Figure 12: Equilibrium of NO_2 and N_2O_4, courtesy the author.

```
/// <trapezoid>
/// The following class represents simple functionality of the
trapezoid.
/// </summary>
using System;

namespace MathClassCS
{
    class MathTrapezoidSample
    {
        private double m_longBase;
        private double m_shortBase;
        private double m_leftLeg;
        private double m_rightLeg;

        public MathTrapezoidSample(double longbase, double
        shortbase, double leftLeg, double rightLeg)
        {
            m_longBase = Math.Abs(longbase);
            m_shortBase = Math.Abs(shortbase);
            m_leftLeg = Math.Abs(leftLeg);
            m_rightLeg = Math.Abs(rightLeg);
        }

        private double GetRightSmallBase()
        {
            return (Math.Pow(m_rightLeg,2.0) - Math.Pow(
            m_leftLeg,2.0) + Math.Pow(m_longBase,2.0) +
            Math.Pow(m_shortBase,2.0) - 2* m_shortBase *
            m_longBase)/ (2*(m_longBase - m_shortBase));
        }

        public double GetHeight()
        {
            double x = GetRightSmallBase();
            return Math.Sqrt(Math.Pow(m_rightLeg,2.0) -
            Math.Pow(x,2.0));
        }

        public double GetSquare()
        {
            return GetHeight() * m_longBase / 2.0;
        }

        public double GetLeftBaseRadianAngle()
        {
            double sinX = GetHeight()/m_leftLeg;
            return Math.Round(Math.Asin(sinX),2);
        }

        public double GetRightBaseRadianAngle()
        {
            double x = GetRightSmallBase();
            double cosX = (Math.Pow(m_rightLeg,2.0) + Math.Pow(
            x,2.0) - Math.Pow(
            GetHeight(),2.0))/(2*x*m_rightLeg);
            return Math.Round(Math.Acos(cosX),2);
        }
        public double GetLeftBaseDegreeAngle()
        {
            double x = GetLeftBaseRadianAngle() * 180/ Math.PI;
            return Math.Round(x,2);
        }

        public double GetRightBaseDegreeAngle()
        {
            double x = GetRightBaseRadianAngle() * 180/
            Math.PI;
            return Math.Round(x,2);
        }

        static void Main(string[] args)
        {
            MathTrapezoidSample trpz = new MathTrapezoidSample(
            20.0, 10.0, 8.0, 6.0);
            Console.WriteLine("The trapezoid's bases are 20.0
            and 10.0, the trapezoid's legs are 8.0 and 6.0");
            double h = trpz.GetHeight();
            Console.WriteLine("Trapezoid height is: " +
            h.ToString());
            double dxR = trpz.GetLeftBaseRadianAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dxR.ToString() + " Radians");
            double dyR = trpz.GetRightBaseRadianAngle();
            Console.WriteLine("Trapezoid right base angle is:
            " + dyR.ToString() + " Radians");
            double dxD = trpz.GetLeftBaseDegreeAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dxD.ToString() + " Degrees");
            double dyD = trpz.GetRightBaseDegreeAngle();
            Console.WriteLine("Trapezoid left base angle is: "
            + dyD.ToString() + " Degrees");
        }
    }
}
```

Figure 10: Coded C++ math functions to calculate the inner angles of a trapezoid.

[35] https://www.redcrab-software.com/en/Calculator/Trapezoid-Angle
[36] https://learn.microsoft.com/en-us/dotnet/api/system.math?view=net-8.0#code-try-2
[37] 2015 Pensyl W., Min, X., Song S., Facial Recognition and Emotion Detection in Environmental Installation and Social Media Applications Encyclopedia of Computer Graphics and Games https://doi.org/10.1007/978-3-319-08234-9 = 978-3-319-08234-9
[38] Pensyl. W.R. Interactive Tutorial for Bruice, McMurry and Faye Chemistry e2. Prentice Hall/Pearson 1998

However, in the now deprecated Flash development software, a math function library was non-existent. One had to write some arithmetic, feed the result into a variable, and then pass the variable to the next piece of arithmetic. Explaining this to the consulting professor demonstrates communication issues in team based work. This has decreased since digital literacy and advances in programming environments that include libraries such as a math library. A library can be called at the head of the code to import the appropriate function. Comparing solving a math equation on paper – the process must be adjusted to fit within the syntax of the programming language. Depending on the devices being used the API may be written in C#, C++ and an efficient algorithm may become cumbersome requiring hours of debugging and testing.

An example from the afore-mention software, Emota v3.0, required the filtering of inconsistencies in a captured image that compensated for shadows. If one is image processing in real-time, then considerations of optimization of the process and code is warranted. (There is a cost to accuracy in "compute cycles." Generally speaking, high accuracy in facial detection will result in slower processing times.) As we stated in our paper, "a vague shadow will not heavily affect Haar classifier performance, and hard shadow edges can be weakened. Thus, instead of complex shadow removal algorithm, we adopt following operations to concentrate effective image information so that Haar classifiers can find target more easily." [39] The formula used is seen in Figure 12. Writing this in the most recent version of Python where a math function library can be called, results in this code seen in Figure 13.

$$f(x)=\begin{cases}\alpha K\log(x)+(1-\alpha)x & ,\ if\ x<127\\ \alpha[255-K\log(x)]+(1-\alpha)x & ,\ if\ x\geq 127\end{cases}$$

Figure 12: Simple shadow removal formula for Haar Classification

shadowAdjust.cs

```csharp
using System;
using System.Math;

public class Program
{
    private static readonly double ALPHA = 0.64;
    private static readonly double K = 256;
    private static double varAdjust = 1;

    public static double Function(double x)
    {
        if (x >= 127)
        {
            return ALPHA * K * Math.Log10(x) + (1 - ALPHA) * x;
        }
        else
        {
            return ALPHA * (255 - K * Math.Log10(x)) + (1 - ALPHA) * x;
        }
    }

    public static void Main()
    {
        double x = 254;
        double functionX = Function(x);
        varAdjust = (double)x + ", f(x) = " + functionX;
        Console.WriteLine($"Input: x: {x}, f(x) = {functionX}");
    }
}
```

shadowAdjust.py

```python
from math import log10

ALPHA = 0.64
K = 256
varAdjust = 1

def function(x: float):
    if x >= 127:
        return ALPHA * K * log10(x) + (1 - ALPHA) * x
    else:
        return ALPHA * (255 - K * log10(x)) + (1 - ALPHA) * x

x = 254
function_x = function(x)
varAdjust = (f"Input: x: {x}, f(x) = {function_x}")
```

Figure 13, 14: The shadow removal formula in C# and in Python Script using math library.

What these instances reveal is the importance of clearly defined specifications in what is input what are the expected outcomes. TWhese are paramount concerns in scientific or medical research and other considerations are subservient. Software engineers, and user interface designers speak in different languages, almost. Much software engineering operates in Murray refers to as the "black box"[40] for others in the team. Yet, with clear specifications for how the data is captured and the expected results are clearly defined, the stakeholders can work effectively together. For a single user, or researcher that writes their own code, the interface can be crude and as arcane as possible. Since the person that is operating the software knows exactly what each element does, the need for an "interface" is limited or even non-existent. The image in Figure 14 is developed in Cycling 74's MAX MSP for detection and tracking of position of persons or objects using a webcam for control a stepper motor or other devices in an interactive media installation. MAX MSP is a software development platform used primarily by music composers and artists. It has very idiosyncratic programming method. In the workspace, the operator sets a series of various modules where

[39] Ibid Pensyl W., Facial Recognition and Emotion Detection in Environmental Installation and Social Media Applications
[40] Ibid, Janet Murray, Inventing the Medium - Principles of Interaction Design as a Cultural Practice Chapter 12, The MIT Press

each represent discrete piece of code, and feeds information between these. The operator does not need an interface that make it easy to use as the requirement of functionality overrides the need for an interface. Figure 15 shows a "presentation" mode of the above-mentioned tracking software,[41] where the functions are "cleaned up" and structured into a rudimentary interface. Even this kind of interface is complex. Working with student usability testing this tracking software still required training. In other words, that interface need more affordance!

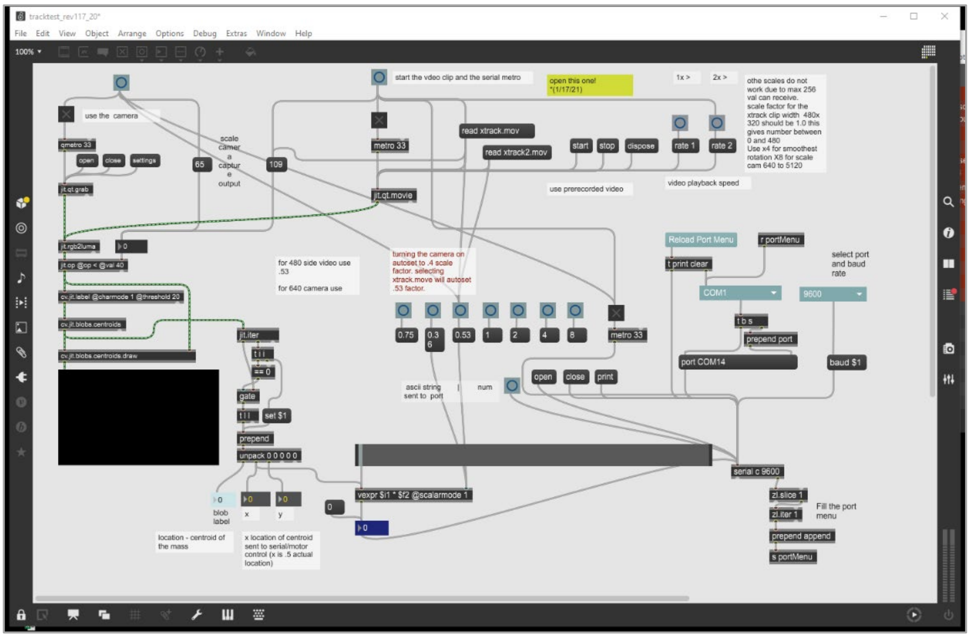


Figure 13: Motion tracker software workspace in MAX MSP, courtesy of the author.

Figure14: Motion tracker "presentation mode" MAX MSP executable interface. Courtesy of the author.

Even with such development platforms, it is still needed to write custom code in another high level programming language such as Javascript or C++. The earlier mentioned emotion detection controlled music app in MAX MSP, emoteControl, required custom Javascript code[42] to get the detected emotion facial expression from the Emota v3.0 executable.(Figure 5.) Communication carried out in the real-time to modify the music stream in the Javascript seen n Figure 16.

[41] Pensyl, W.R. Motion tracker executable compiled from Max MSP, Cycling 74
[42] Custom Javascript code in MAX MSP

In the afore-mention , the music application was developed in MAX MSP, while facial recognition and emotion detection was carried out in Emota v.3.0. These two software applications run simultaneously. The interface is less of a concern the closer and more expert the user is to the process. If the user is a chemistry student using an interactive tutorial that simulates a chemical reaction, then the structure of interface and the interaction strategy must provide appropriate affordances. These considerations are more important, depending on who is responsible for data collection in research environments. Graduate students may often be running experiments, and data collection sessions. In these cases, the need for clear and understandable interfaces and interactions is evident. Since this paper is intended for audiences in the research domain, we can focus primarily on systems that an academic researcher needs to put together for their data capture in their study. Devices that are already in the marketplace have undergone significant design and testing both for usability and utility. There are considerations that a UI designer and Software Developer can raise that will be valuable to these projects. As stated above, working in team-based projects or outsourcing design and coding requires some knowledge of what the other stakeholders are doing and what skill sets they bring and how to effectively communicate across the discipline divides. In the initial discussions, the UI designer will ask for clear directions on what the operator of the system needs to see and what they are expected to do. Most likely there will be requests for site visits and observation. Observation is expected as well in the testing and refinement phase.

However, development is much more complex as one needs to write fully code based applications using programming environments such as Visual Studio in a high-level programming language and a front-end interface built with Windows Presentation Foundation XAML UI design scripting, or Windows Console. (Universal Windows Platform is likely to be deprecated.) One can still use WinForm in Visual Basic, forms UI and Console UI are both ugly in the opinion of the author.

The benefit of WPF and XAML for developers and UI designers familiar with HTML/CSS/Javascript web development is that the syntax and functionality of XAML is similar enough to allow easy migration. Microsoft allow offer the GUI based XAML development package, Blend. However, it is most likely the UI designer will work with a dedicated software engineer for creation of content.

Constraints & Affordances, Frame Analysis, Formal Analysis, and Organizational Cultural, Design Process as an Analysis tool

Specifications phase for user interface:

User interface design and testing

Flow, affordances and sequences of actions

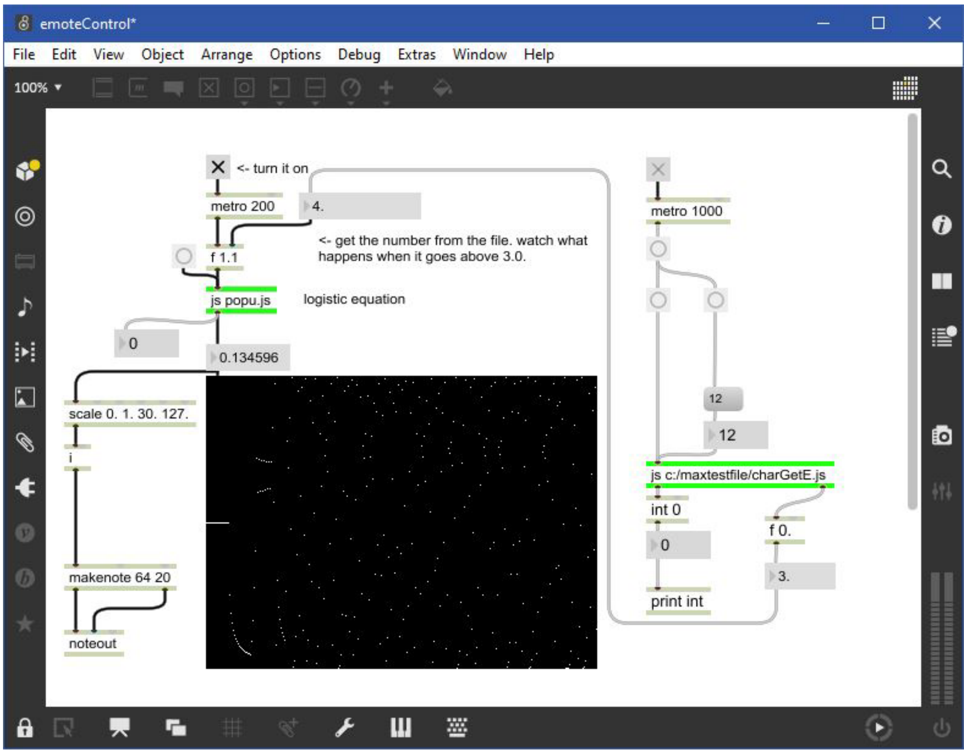
What data needs to need to be visible in capture time

Post process reliability and evidence in successful data capture in sessions - ??

Who are the users – MDs, expert lever technicians? Training of operators

Clarity of UI elements and affordances.

Observation in session, adjustment and refinement of UI and flow.



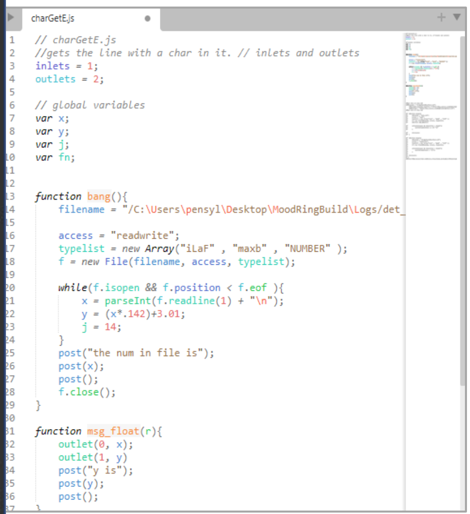
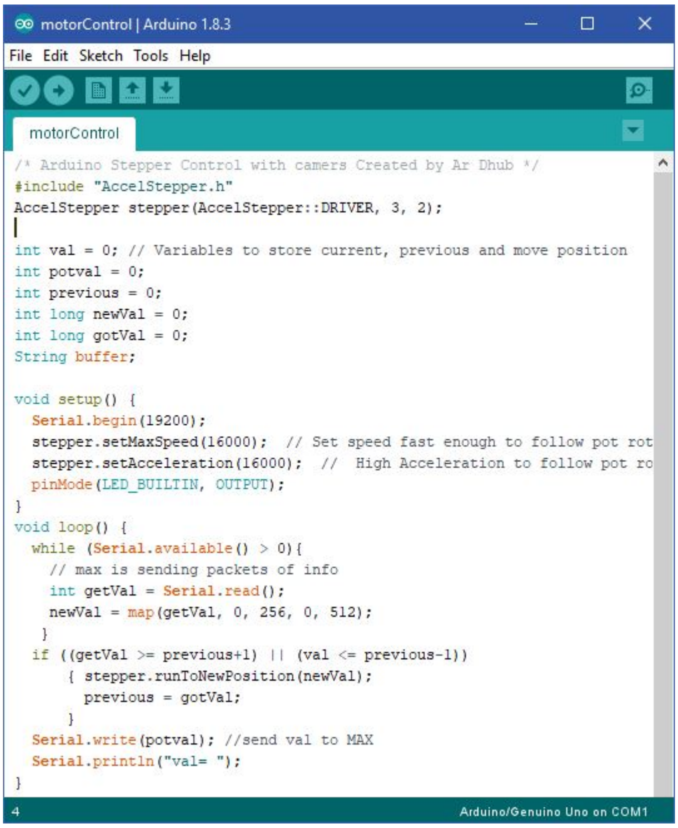


Figure15: Custom Javascript code affords communication between emoteControl and Emota v3.0.



5. Considerations for team-based development of research tools.

Specifications phase.

Technology/sensor search and evaluation:

Testing of data collection, sampling frequencies, software API for the collection – written in which language (C+, C#, Python, etc.)

CPU, Architecture – Wireless/Bluetooth, USB bus speed and interface, system level interferences,

CPU – component communications and usage impact on performance. ex. using dual screen vs single screen

CPU – component communications and usage impact on performance. ex. using dual screen vs single screen

Graphics Cards – minimum specifications, throughput on PCI bus. Software interaction in APIs and SDKs for the graphics cards and any impacts on sampling frequency in data collection.

Security – WIFI, environments, Wireless encryption, wired internet connections, data collection – encryption needs?

Sampling frequencies of devices used for data collection. System clocks on devices – is there a clock? Can it be reliably used – data collection in wireless devices – probabilistic data transmission. Interference in the wireless environment. Power management in devices and its impact on data capture.

Synchronization and reconciliation of sampling frequency in real time vs post capture re-processing.

Operating systems – system clock in Window, vs Device clocks.

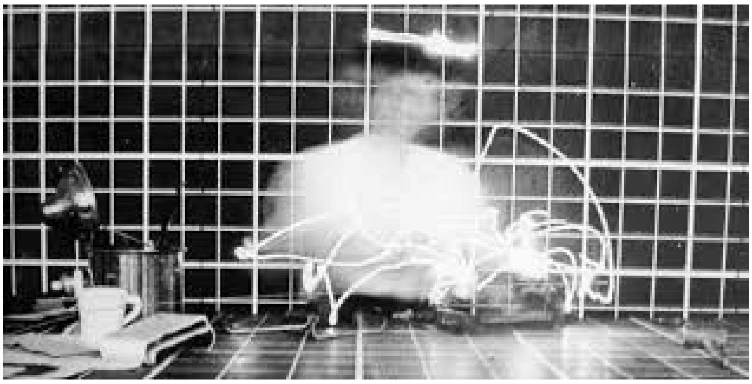
Multithread processing – Invoking commands in Windows, never happens simultaneously. Reconciliation of system clock time commands vs device clock responses.

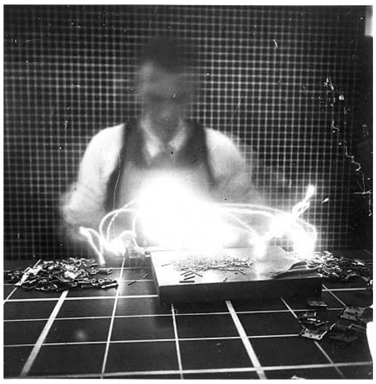
Processing time to post data – buffering and file dump on end of data capture.

GE EKG

60hz. 12 data points turn-key windows 10, dedicated card set for data collection.

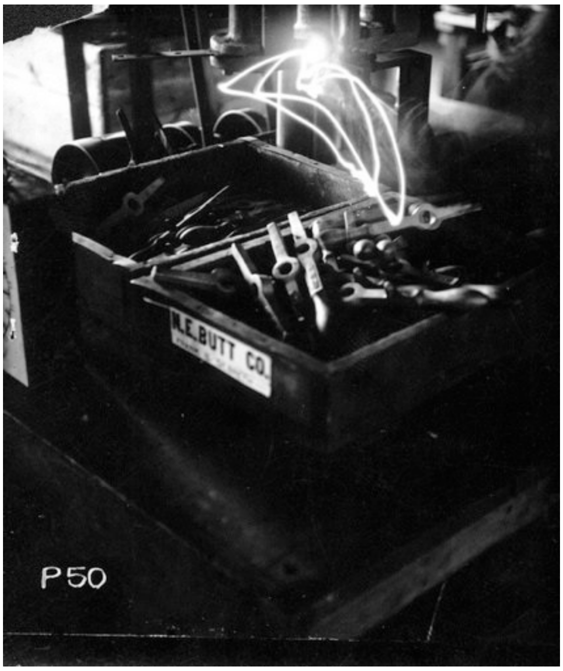
The systems designed to maximize effort and work can be analyzed for ways to increase usability, increase effectiveness, and to create more ergonomic and efficient actions. This is the beginning of the user centric design processes. However, the effort of early researchers looked to understand **how to better adapt human actions to the systematized processes** of the assembly line and other mechanized processes.





Gilbreth’s motion studies

https://en.wikipedia.org/wiki/Time_and_motion_study



In applying Motion Study method to work, Gilbreth and his successors found that the key to improving work efficiency was in reducing unnecessary motions. Not only were some motions unnecessary, but they caused employee fatigue. Their efforts to reduce fatigue included reduced motions, tool redesign, parts placement, and bench and seating height, for which they began to develop workplace standards. Gilbreths' work broke ground for contemporary understanding of ergonomics.

Part of this research studied how to have worker’s motions fit the process or mechanization.

$$g(x,y)=\frac{k^2}{\sigma^2}\cdot e^{-\frac{k^2(x^2+y^2)}{2\sigma^2}}\cdot(e^{ik\cdot\begin{bmatrix}x\\y\end{bmatrix}}-e^{-\frac{\sigma^2}{2}}),\ where\ k=\begin{bmatrix}k_v\cos\varphi\\k_v\sin\varphi\end{bmatrix},\ k_v=2^{-\frac{v+2}{2}}\pi$$

Figure 11: Formula for processing Gabor Bank filters for earlier mentioned Emota v3.0 software.

The formula below in Figure 11 processes for classification of Gabor filters for facial expressions resulted in a significant amount of code. Gabor filters are implemented to derive orientations of features in the captured image using pattern analysis. The Gabor filters enable a more accurate positioning of graph vectors in an "elastic bunch map graph" that is aligned with facial features.[43]

[43] 2015 Pensyl W., Min, X., Song S., Facial Recognition and Emotion Detection in Environmental Installation and Social Media Applications Encyclopedia of Computer Graphics and Games https://doi.org/10.1007/978-3-319-08234-9 = 978-3-319-08234-9

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented, and virtual realities; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Virtual reality

Biography

Russell Pensyl (MFA 88, BFA 85) is an American media artist and designer. Pensyl's current work includes the creation of location based entertainment several areas of technology in the application of content delivery in environmental spaces including emotion detection, facial recognition, positioning and localization, gesture recognition. Pensyl is a noted pioneer in mixed and augmented reality, creating one of the first integrations of mixed reality in theatrical a production. Pensyl was a finalist for the 2021 Premio Arte Laguna in Venice, Italy., and exhibited in the Heaven and Hell exhibition in Chicago IL, the 2020 The International Society of Experimental Art, Calgary, AB, Canada, the 2020 Altered States, National Exhibition of American Art, Pleiades Gallery in New York. In 2011, his installation "subtle presence" was included in the Sarajevo Winter Festival, In 2008,"The Long Bar" was Invited into the SIGGRAPH Asia Synthesis – Singapore. Pensyl's work has been included in the Shang Hai Biennial, SIGGRAPH USA, the Machida Museum of Art and many other exhibitions.