Instituto Tecnológico de Ciudad Victoria

Maestría Profesionalizante en Sistemas Computacionales

Thesis protocol

Plataforma web para el análisis de la gesticulación facial

Jaime Jesús Delgado Meraz

Thesis Advisor:
Dr. Pedro Luis Sánchez Orellana

Thesis Co-advisor:
Dr. José Torres Jiménez

Scholarship:DGEST

Ciudad Victoria, Tamaulipas - October 16, 2012

Contents

1 Introduction

1.1 Justification and Motivation

Human interaction is very important in many different aspects in our daily life, such as business meeting, medical diagnostics, etc. However, thanks to the improvement of technology, these operations can be done remotely, which can be especially useful when the distance is considered a barrier. Nevertheless, this produces another problem which is the lack of physical gesture apreciation (product of non-presencial interaction), since gesture comunication is considered as an important part in human comunication. To solve this issue, we propose the use of a web service that allows to evaluate gestures remotely.

1.2 Problem Statement

Given a sequence of images from the face, taken in a client computer, where the illumination conditions and background motion are restricted, the problem of analyzing facial gestures on the web, could be stated as an architecture that allows transmission of compressed images through a web service, in order to produce an analysis of facial gestures (by the use of an existing algorithm). This algorithm focuses on the detection of symmetry of facial gesticulations and its goal is to help in the decision-making process. From this we state the following research questions:

- Considering the speed of internet connections in México, what techniques can be used to increase the speed of image transfer?
- Which type of web service is more suitable for image transmission/analysis?

1.3 Hypothesis

Nowadays, Internet technologies are optimized for low bandwith consumption; nonetheless, in order to obtain a more efficient analysis it is necessary to reduce the amount transfered information. At this point image compression techniques could increase the speed of the analysis by providing fast transmission of images through internet.

1.4 Objectives

1.4.1 General

Design a web architecture that allows the efficient transmission of facial image sequences to compute an analysis about the behaviour of facial gesticulations, in order to obtain symmetrical/asymmetrical motion patterns which might be used for the decision-making process.

1.4.2 Specifics

- Implement an image compression technique that allows fast Internet broadcasting.
- Design and implement a web architecture for the reception and decompression of compressed images.
- Adapt a given algorithm for the analysis of gesticulations in the transmitted images.
- Design a set of experiments to test the web architecture.

1.5 Research Scope and Constraints

The proposed architecture will focus only on the compression-transmission process of images, it will not consider the motion from other regions of the body. Moreover, the mechanisms for the analysis of the image sequences were generated in collaboration with researchers of the Universidad Politécnica de Victoria and will only be used, but not implemented.

2 State of the Art

2.1 Face Analysis

The analysis of facial expressions constitutes a critical and complex portion of our non-verbal social interactions. Therefore, over the past years the computer science community has developed different methodologies or strategies to automatically analyze the facial expression [13]. Facial expression analysis is one of the most difficult and interesting problems in computer vision due to variability in shape and texture of the face. These variations are caused by some factors such as pose, illumination changes and occlusion. It is important to consider that facial expression recognition is different from facial expression analysis; the first is focused on classification of the structure of the face into a set of general emotions, see Figure 1(A). The second focuses on meassuring how these emotions are produced in the face (mainly through facial muscles deformation analysis), see Figure 1(B).

The analysis of human facial expressions has huge implications in areas like psychology, since, according to Ekman [11], through this, it can not only be classified into one of the general seven emotions but also how they are generated. The corner stone for this is the symmetry of the facial muscles activity, which by its analysis it can be found whether an emotion is natural or not.

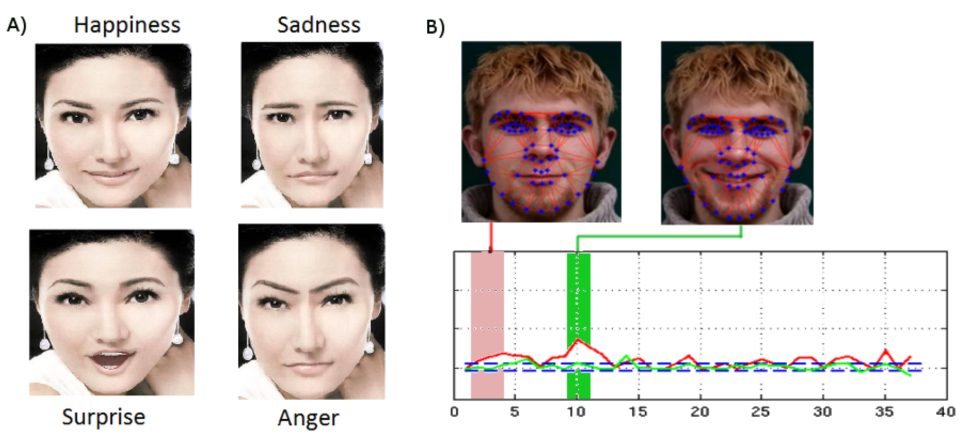


Figure 1: (A) Basic human emotions. (B) Facial gesture analysis. Emotion is a complex psychophysiological experience of an individual's state of mind as interacting with biochemical (internal) and environmental (external) influences

In computer vision and artificial intelligence (AI) there are many sub-problems related to facial emotion recognition such as: pose variations, people constantly moving their head in different angles; illumination changes, different environments with different illumination conditions; occlusion, which occurs when an object is in front of the face difficulting the analysis feature extraction task; background complexity, multiple objects in the environment with non-uniform background contrast; finally, image resolution which affects the accuracy of the tracking/recognition process. Besides, from the computational point of view, this problem implies the classification of facial motion or facial structure deformation into abstract representations completely based on visual information.

2.1.1 Basic concepts

Computational facial analysis methods can be classified in deformation extraction methods and motion extraction methods [13]:

Motion extraction: these approaches focus on facial changes presented as consequence of facial expressions.

Deformation extraction: these approaches work with a neutral face images or face models for extract relevant facial features not caused by wrinkles or accidents.

The main difference between them is that deformation-based methods can be applied to image sequences, in the second case, the analysis can be applied to both single images and image sequences by processing frames independently. However, deformation-based feature extractors miss low-level directional motion information, i.e. they cannot reconstruct pixel motion. Although face motion can be inferred by using facial feature models it requires extra computation time to achieve it.

In both cases, facial features may be processed holistically (the faces are analyzed as a whole) or locally (focusing on features from specific areas).These approaches are focused on the extraction and analysis of two types of facial features [13]:

Intransient facial features: always are in the face and could be deformed by facial expressions, for example, eyes, eyebrows and mouth.

Transient facial features: consider wrinkles and bulges in the face, for example, when the people open and close their eyes or mouth, face changes.

In both cases, the success of the technique depends on the capacity to segment the face from background. The correct segmentation allows an accurate extraction of points of interest from the faces. To do it, facial processing may take place either holistically, or locally, by focusing on facial features or areas that are prone to change with facial expressions. The task of separating the face from the background is done through segmentation, which allows the isolation of transient and intransient features within the face, that can be used to separate faces of interest from the background.

2.1.2 Facial gestures analysis methods

Facial gesture analysis is a challenging task and it has several application related to human interaction and computers. Traditionally, facial gestures recognition classifies the expressions into seven general emotions (anger, happiness, fear, sadness, contempt, disgust, surprise), in contrast, the facial expression analysis involves the reconstruction of facial motion, considering this way the facial muscle periods of motion between one gesture and another. In this cases it is necessary a robust classification and tracking technique of face gestures and head position [1].

There are three main steps for working in facial gestures recognition: face detection, extraction of facial gestures information and classification of the information extracted. As it was previously stated, the methods for facial gesture analysis can be grouped in deformation extraction and motion extraction.

2.1.3 Review of Deformation and Motion Extraction Methods

Over last years researchers have developed many approaches and models for face recognition and analysis with good results by using both deformation and motion extraction approaches. Although, the election of an specific technique relies on the characteristics of a certain problem. Deformation extraction have been successfully used to solve emotion recognition tasks, since they can be used to analyze single images. However, if the problem consists in analyzing a sequence of images to extract the *motion* (caused by muscles actions) of different regions in the face while making a gesture, the motion extraction techniques are ideal to this goal, since they have been applied to evaluate the motion produced in areas like eyes or mouth, by using a model generated a priori.

In general, motion techniques uses either 2D or 3D information. Typically, 2D based techniques are more used due to their fastness, product of a reduced dimensionality of the information. Nevertheless, a very large training data sets is needed to model the angular variations and illumination changes. In the case of 3D based approaches, the accuracy reached is higher than 2D based ones [12], since they are rotation invariant, although their sensitiveness to illumination changes remains.

In the case of facial expression analysis, precision is more important than processing speed, since the analysis can be carried on offline. Considering that, 3D based techniques are more relevant to solve the problem; especially those approaches with local oriented processing, since they allow analyzing information from specific points in the face (opposed points in the face are required in order to analyze the symmetry).

2.1.4 3D based local analysis techniques

As it was previously stated, the main difference about these techniques is the way they process information. Holistic approaches considers the whole image to extract motion vectors, which could be used for extracting general information about the motion of the face; still to increase the accuracy of facial expressions analysis requires the motion interpretation of specific regions in the face. On this line, there are several techniques in the state of the art (see Table 1) that focuses on specific regions in the face:

Reference	Sequences	Landmarks	Database	Expr. Types	R.P. (%)
Yin [46]	2D + 4D	64 semi-auto	BU-3DFE	6	80.2
Wang [43]	2D + 3D	58 auto	Private	4	83.0
Sun [39]	2D + 4D	83 auto	BU-4DFE	6	80.9
Rosato [32]	2D + 4D	22 auto	BU-3DFE, BU-4DFE	7	80.1
Zhao [47]	2D + 3D	19 manual	Bosphorus	7	94.2
Tsalakanidou [40]	2D + 4D	81 auto	Private	5	85.0

Table 1: Methods for $3D/4D$ Facial Expression Analysis (Headers: **Sequences** denotes the dimensions of the images where 2D denotes whether the method makes use of the 2D texture associated with the 3D data and 4D denotes whether the method uses temporal information from a sequence of 3D data; **Database** denotes the database from which a subset was selected; **R.P.** denotes reported performance, which is the average recognition rates).

The approaches described previously have the ability to perform facial expressions analysis considering 3D information. These techniques can be classified into two main groups, depending on the labelling mechanism: automatic and manual. In the case of the automatic labelling mechanisms, authors have used automatic fitting techniques to extract different numbers of facial regions of interest (22- [32] and 58 [43]). In the case of the manual labelling mechanisms, the existing techniques require off line characterization of the facial regions of interest. As it can be seen in Table 1, there is a difference on the percentage of recognition achieved by techniques in these two categories: methodologies designed to manual labelling reached as top a 94.2% compared to 85.0%. Also from the table 1, there is no clear difference about the number of points to be used (but the higher the number of points is the higher computational complexity as well as a higher recognition rate). Although it remains to review, automatic labelling techniques are more desirable for facial expression analysis problems, since they reduce the dependance from user interaction.

2.2 Neural Networks

The study of facial gestures has acquired an increasing interest since it has a huge application area. Facial gestures can be defined as a movement produced in the face by muscles while contracting (see Figure 2) with the objective to transmit thoughts or emotions, for example when a person communicates with others. The interpretation of non-verbal human communication, like facial gesticulation during a talk, is a very common activity and is done intrinsically. Equally important is the interpretation of gestures, through this, people can improve their interpersonal understanding. For example, by analyzing the mood of a speaker it is possible to confirm the intention of a message.

Traditionally, computational applications have tried to interpret emotion by associating a facial image to one of the seven general expressions (such as fear, anger, happiness, etc) but they are too general and they do not evaluate the symmetry of the emotion. This can be solved considering the psychological perspective of the emotions. According to Ekman [11] the symmetry of a gesticulation can describe the veracity of a expression. The bases for this are some studies of the amygdala [35], which is a region in the brain (located in the medial temporal lobes) that performs the analysis of expressions.

However, this task is complex, since involves temporal processing of muscles contractions. To approach this problem, one can look at biological systems that perform this analysis accurately. An example of this is the human brain, which can perform the analysis even under varying environment conditions like: illumination changes, occlusion and pose changes. This is a motivation for the developing of bio-inspired systems based on the behavior of the amygdala, more precisely modeling the neuron interaction for analysis facial motion produced by the gestures.

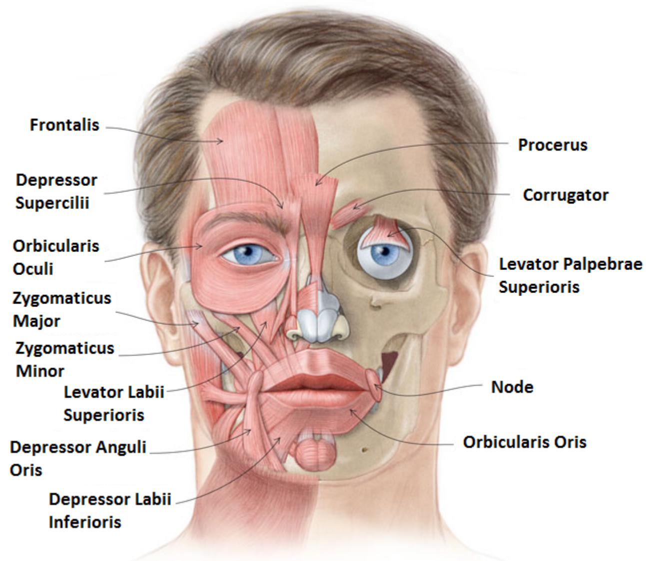


Figure 2: Facial muscles

From all the involved parts gesture eye gaze is the most important element since it express sincerity and credibility. Together, all the elements of the facial muscles can transmit the feelings of the speaker, from happiness, to depth concern. This was previously affirmed by Darwin [9], who stated that facial gesticulation for emotion transmition is innate and very similar for all people, since we evolved very similarly. His theory was not accepted before due to some detractors like Bruner et al. [5]. Later, Ekman et al. [11] solved this issue definitively by pointing out methodological problems that had confused other researchers. They showed that observers could agree on how to label both posed and spontaneous facial expressions in terms of either emotional categories or emotional dimensions. Much evidence, including reanalysis of negative studies, indicated that facial expressions can provide accurate information about emotion.

Therefore the analysis of facial expressions constitutes a critical and complex portion of our non-verbal social interactions. Over the past years there has been an increasing interest on developing automated computational tools for facial expression analysis.

To address this analysis it is necessary to perform two main tasks: face acquisition and facial feature extraction. In the first one, techniques have been developed to deal with the extraction and normalization of the face, considering variations in the pose and illumination. However, most of the effort has been focused on the feature extraction issue [14], since techniques like appearance-based model try to deal with significant variations in the acquired face images without relying on normalization.

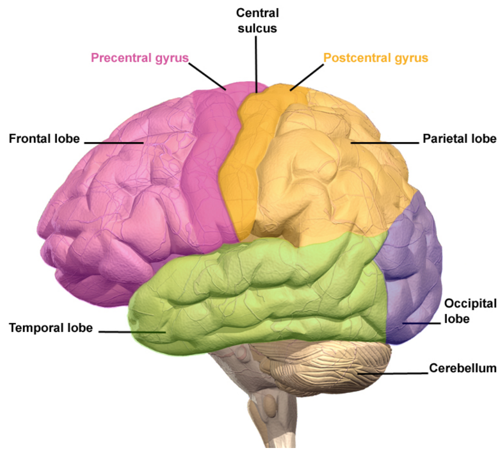


Figure 3: Brain anatomy

The analysis problem has been approached from two main streams [14]: facial deformation extraction models and facial motion extraction models. Motion extraction approaches directly focus on facial changes produced by facial expressions, whereas deformation-based methods contrast the visual information against face models to extract features product of expressions and not by age deformation like wrinkles.

Another approach to solve this problem, is by looking at biological systems that perform accurately this task. The human brain can analyze the information from the face using an area called amygdala which receives connections from other areas like visual cortex. How exactly the amygdala works remains unknown; however, it is know that this area sends impulses to the hypothalamus for activation of the sympathetic nervous system, to the thalamic reticular nucleus for increased reflexes, to the nuclei of the trigeminal nerve and the facial nerve, and to the ventral tegmental area, locus coeruleus, and laterodorsal tegmental nucleus for activation of dopamine, norepinephrine and epinephrine.

The cortical nucleus is involved in the sense of smell and pheromone-processing. It receives input from the olfactory bulb and olfactory cortex. The lateral amygdala, which send impulses to the rest of the basolateral complexes and to the centromedial nuclei, receive input from the sensory systems. The centromedial nuclei are the main outputs for the basolateral complexes, and are involved in emotional arousal in rats and cats. Several studies confirm that the amygdala receives inputs from visual cortex when it comes to emotion processing. Lateralization of processing visual information in the amygdala might indicate symmetry analysis activity which was first found by Ekman et al [11]. In their studies they found that emotion understanding is directly related to symmetrical analysis of opposed facial regions.

This information is relevant since provides hints on the role of the amygdala during the expression of emotions and visual interpretation of them. How the visual information is processed remains unclear, but it is known that the information is processed laterally (information is processed in one hemisphere). This can be used as a source of inspiration to build a neural architecture that allows processing opposed facial regions to interpretate emotions. Previously some researchers have used traditional neural networks (NN) to recognize emotions from face images[23]; however, NN in those works are used for recognition purposes though symmetry understanding in facial gestures requires analyzing the evolution of muscles during gesticulation rather than comparing gesture moments. This means that not all neural networks models are adequate for this task, in fact NN approaches can be classified in 3 generations, the two first are usually related to recognition tasks while the third also allows temporal information processing.

2.2.1 Generations of Neural Networks

Over the past hundred years, research has accumulated knowledge about the structure and function of the brain. The elementary processing units in the central nervous system are neurons which are connected to each other in an intricate pattern. Examples of this types of neurons are sketched in Figure 4 which shows a drawing by Ramón y Cajal, one of the pioneers of neuroscience around 1900. In his work, he distinguished several types of neurons, with different physical structures like triangular or circular. This picture gives a glimpse of the network of neurons in the cortex. In fact, cortical neurons and their connections are packed into a dense network with more than 104 cell bodies and several kilometers of “wires” per cubic millimeter. In all areas, however, neurons of different sizes and shapes form the basic elements. Wolfgang Maass outlined past and current artificial neural network research into three generations and made the following observations [26].

A typical neuron can be divided into three functional parts called: dendrites, soma, and axon (see Figure 5). Roughly speaking, the dendrites play the role of the “input device” that collects signals from other neurons and transmits them to the soma. The soma is the “central processing unit” that performs an important non-linear processing step: If the total input exceeds a certain threshold, then an output signal is generated. The output signal is taken over by the “output device”, the axon, which delivers the signal to other neurons.

The junction between two neurons is called a synapse. Let us suppose that a neuron sends a signal across a synapse, the sender is called presynaptic cell and the receiver is called postsynaptic cell. A single neuron in vertebrate cortex often connects to more than

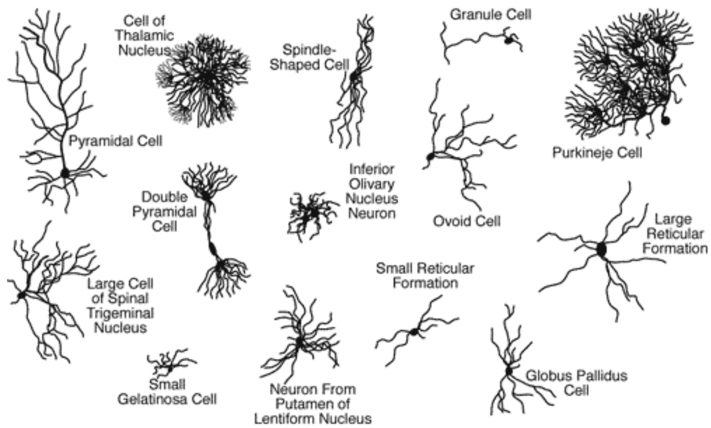


Figure 4: There are three kinds of neurons: motor neurons (for conveying motor information), sensory neurons (for conveying sensory information), and interneurons (which convey information between different types of neurons). The image identifies how neurons come in various shapes and sizes.

10^4 postsynaptic neurons. Many of its axonal branches end in the direct neighborhood of the neuron, but the axon can also stretch over several centimeters so as to reach to neurons in other areas of the brain.

In the literature there are different computational models that intent to simulate the behavior of these neurons communities. Roughly, they can be divided in three generations, considering their chronological appearance order. Our attention is centered in the third generation since it allows temporal processing of input patterns.

The **first generation** is based on the McCulloch-Pitts neuron (also known as a perceptron or a threshold-gate) as the basic computation unit. Models of the first generation, such as the multi-layer perceptron, use digital input and output, usually binary or bipolar. The perceptron is a type of artificial neural network (ANN) invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt [33].

The **second generation** is based on computation units (neurons) that use an activation function of a continuous set of possible output values.This generation, like first one, can compute arbitrary boolean functions (after using a threshold). Also, a concept of hidden layer appeared, through it more complex functions can be approximated [34]. Important to many implementations is the fact that second generation networks support learning algorithms based on gradient descent, such as error back-propagation.

The **third generation** is known as Spiking Neural Networks (SNN) model. They

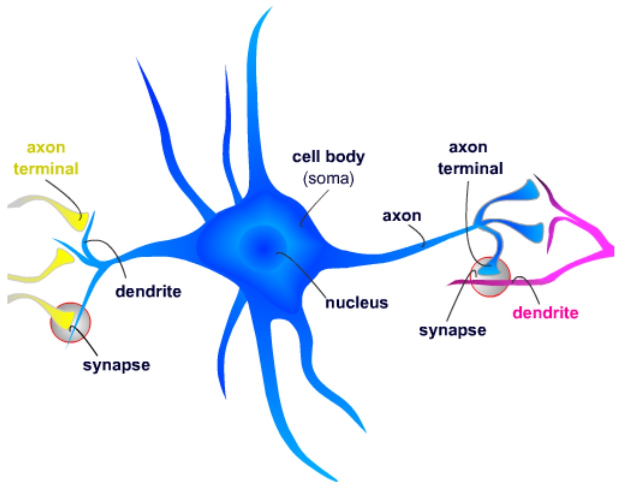


Figure 5: Structure of a neuron

increase the level of realism in a neural simulation, in addition to neuronal and synaptic states. The SNNs also incorporate the concept of time into their operating model, the idea is that neurons in the SNN model do not fire at each propagation cycle (as it happens with typical multi-layer perceptron networks), but rather fire only when a membrane potential[1] reaches a threshold. When a neuron fires generates a signal that travels to other neurons that are stimulated (excited or inhibited) by this signal. In the context of spiking neural networks, the current activation level (modeled as some differential equation) is normally considered to be the state of the neuron, with incoming spikes pushing this value higher, and then either firing or decaying over time. Various coding methods exist for interpreting the outgoing spike train as a real-value number, either relying on the frequency of spikes, or the timing between spikes, to encode information.

These networks, like multi-layer perceptron networks, can approximate continuous arbitrary functions, as well as, temporal encoded inputs and outputs ([26]). This kind of neural network, in principle, can be used for information processing applications the same way as traditional artificial neural networks. However, due to their more realistic properties, they can also be used to study the operation of biological neural circuits. Some successful models of SNNs have been used to solve real-life problems, such as, navigation in mobile communities of robots [44].

[1]an intrinsic quality of the neuron related to its membrane electrical charge

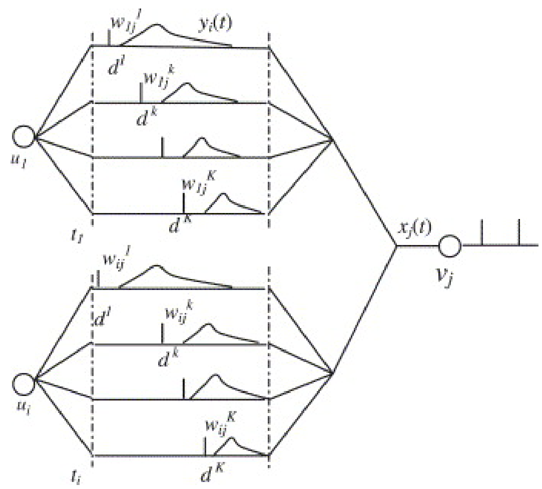


Figure 6: Spiking Neural Network

2.2.2 Spiking Neural Networks

In practice, there is a major difference between the theoretical power of spiking neural networks. Some large scale neural network models have been designed to take advantage of pulse coding found in spiking neural networks. One of the most exciting characteristics of spiking neural networks (with the potential to create a step-change in our knowledge of neural computation) is that they are innately embedded in time [26].

Spike latencies, axonal conduction delays, refractory periods, neuron resonance and network oscillations all give rise to an intrinsic ability to process time-varying data in a more natural and computationally powerful way than is available to 2nd generation models. Real brains are embedded in a time-varying environment; almost all real-world data and human or animal mental processing has a temporal dimension. Evidence is growing that rhythmic brain oscillations are strongly connected to cognitive processing.This type of processing is well suited to problems like face gesticulation characterization, since this problem implies the processing of spatial features over time periods to abstract their relations over time.

In the literature there are several SNN models that have different characteristics that make them desirable for specific types of problems. According to Izhikevich [24], to define the model that will be used to interpret facial motion it is necessary to consider two aspects about the selected SNN: biological plausibility, and computational time cost. A comparison between several SNN models is presented in Figure 7, where the zero shows those models means with the highest efficiency in terms of computational viability.

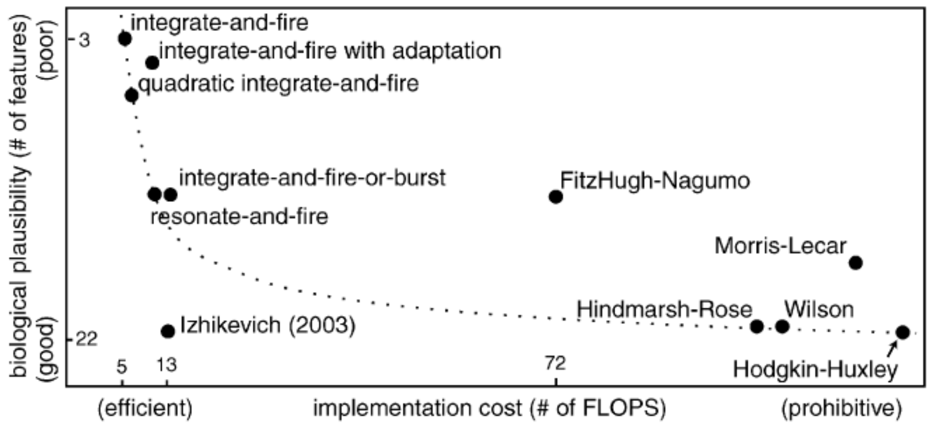


Figure 7: Comparison of spiking models: biological plausibility against implementation cost.

The gesture analysis methods, traditionally, can be used in situations like job interviews or business meetings where participants are in the same place, however, nowadays these events are carried out online, thereby decreasing the perception of those involved. Therefore arises the need of a tool or architecture that allows the use of gesture analysis methods online, but this task is not trivial, because it requires mechanisms for exchanging data between the client and the server. This issue could be solved by the use of web services, a technology that has been used for streaming video and audio.

However, given the characteristics of Internet services in the country, the transfer of information would not be very optimal, therefore a technique that allows the compression of images without loss of quality, is needed, in order to be able to perform the gesture analysis more efficiently.

2.3 Web Services

Since its inception, the Web has been an open frontier of exploration in software and network system design. New ideas were tried and tested first, but organized and standardized later, once they proved their utility. For example, HTTP[2], the transport protocol of the Web, had been in use for more than half a decade before its state of practice was written down as HTTP/1.0 [3] in May 1996. But the standardization process continued until 1999, when the final revision of HTTP/1.1 [16] standard was completed. The architectural

[2]Hypertext Transfer Protocol

principles behind HTTP and other Web standards were described by Fielding [17], thus completing the process. HTML[3] has followed a similar path. It started out with a simple set of tags for structuring text and graphics on Web pages. As the number of content types (new multimedia formats, more sophisticated ways of displaying text, interactive Web pages [18]) grew, the HTML tags were pressed into service of displaying them in various non-standard ways. After nearly two decades of this growth, new multimedia HTML tags were finally added and standardized by W3C[4] in HTML5, which was completed in 2012 [22].

A similar sequence of events – simple beginnings leading to an unruly explosion followed by some type of organization – can be observed in the realm of Web services. Web services concern the way in wich software communicates. Software can come in many forms from a simple script on a personal computer, to an application on a networked server, through to a large operational support system running in a mainframe computer.These scripts, applications and software systems can be viewed as software components, considering the following characteristics of such a software component:

- It can **describe** itself – so that other components can understand the functionality it offers and how to access that functionality,
- it can allow other components to **locate** it – so it can be used when required,
- it can be readily **invoked** whenever another component wishes to use its functions.

If such a software component were installed in a network, its services could be used by other software components. These components are said to be providing a web service. A *Web service* then, is an interface that describes a collection of operations that are network accesible through standardized messaging and that performs an specific task or set of tasks[20, 27].

The use of web services on the World Wide Web is expanding rapidly as the need for application-to-application communication and interoperability grows. These services provide a standard mean of communication among different software applications involved in presenting dynamic context-driven information to the user[2]. An example is a web service that returns a credit rating when provided with user's ID, or an order handling system that provides user and usage data to a billing web service wich then returns the user's bill. Web services can also be integrated together to provide greater value-add. For

[3]Hypertext Markup Language
[4]World Wide Web Consortium

example, a travel management web service may make use of the capabilities of the web services providing car hire, hotel bookings and flight reservations. Or a video-on-demand web service may make use of a communication service that delivers the video stream.

The first Web services were built for passing remote procedure calls (RPCs) over the Web. The idea took off quickly and resulted in a large collection of standards . Surprisingly, these standards were defined with little consideration to the contemporary practice; sometimes before there were any implementations to standardize. The end result of this premature standardization was confusion, rather than order that standards usually bring. In response, an alternative style of Web services, built according to the rules of the Web, began to appear. These (so-called RESTful) Web services are maturing, or, more precisely: people are re-learning to use the tried-and-true standards of the Web and applying them when building Web services.

2.3.1 SOAP

In 1998, a new remote procedure call protocol was defined with the ability to go through firewalls, this new protocol was named as Simple Object Access Protocol (SOAP), it was designed to be a platform and language-neutral alternative to previous middleware technologies like CORBA and DCOM. Its first public appearance was an Internet public draft (submitted to the IETF[5]) in 1999; shortly thereafter, in December of 1999, SOAP 1.0 was released. In May of 2000 the 1.1 version was submitted to the W3C where it formed the heart of the emerging Web Services technologies. The current version is 1.2, finalized in 2005. From its definition up to its standardization, SOAP has evolved quite a lot to become more flexible in terms of the data to be transferred and also protocol agnostic (supporting different bindings of WSDL[6])[28].

Together with WSDL and XML Schema, SOAP has become the standard for exchanging XML-based messages. It was also designed from the ground up to be extensible, so that other standards could be integrated into it –and there have been many, often collectively referred to as WS-*[7]. Hence much of the perceived complexity of SOAP, comes from the multitude of standards which have evolved around it[38].

[5]Internet Enginnering Task Force

[6]Web Services Description Language

[7]The term WS-* is used to refer to services based on the SOAP standard, and other WS-* standards (e.g WS-Addressing, WS-Security) defined specifically for Web services.

2.3.2 REST

Much in the way that Ruby on Rails was a reaction to more complex web application architectures, the emergence of the RESTful style of web services was a reaction to the more heavy-weight SOAP-based standards. In RESTful web services, the emphasis is on simple point-to-point communication over HTTP using plain old XML (POX)[38].

The origin of the term “REST” comes from the doctoral thesis from Roy Fielding[17] (who also participated in developing HTTP 1.0 and HTTP 1.1), where he described the concept of *Representative State Transfer* (REST). He saw REST as a way to help communicate the basic concepts underlying the Web. In order to understand REST, it is necessary to understand the definition of resource, representation and state, in the context of web services.

Resource Can be anything, wether it be a physical object or an abstract concept. As long as something is important enough to be referenced as a thing itself, it can be exposed as a resource. Usually a resource is something that can be stored in computer an represented as a stream of bits.

Representation Is any useful information about the state of a resource. A resource may have multiple different representations.

State In REST there are two types of state, one is resource state wich is information about a resource, and the other is application state, wich is information about the path the client has taken through the application. Resource state stays on the server and application state only lives in the client.

REST provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components, and intermediary components to reduce interaction latency, enforce security, and encapsulate legacy systems[17].

2.3.3 Principles of Web services

Roy Fielding documented REST based on the principles that emerged as the Web evolved. He noticed that Web servers, clients, and intermediaries shared some principles that gave them extensibility to work on the large-scale of the Internet. He identified four principles of REST (which he called constraints):

- Identification of resources.
- Manipulation of resources through representations.
- Self-descriptive messages.
- Hypermedia as the engine of application state (abbreviated HATEOAS).

These principles combine into a short and consistent metaphor of systems and interactions that make up the Web. The building blocks of the Web are called resources. A resource is anything that can be named as a target of hypertext (e.g. a file, a script, a collection of resources). In response to a request for a resource, the client receives a representation of that resource, which may have a different format than the resource owned by the server. Resources are manipulated via messages that have standard meanings; on the Web, these messages are the HTTP methods. The fourth principle means that the state of any client-server interaction is kept in the hypermedia they exchange, i.e., links, or URIs. Any state information is passed between the client and the server in each message, thus keeping them both stateless.

It's easy to evaluate any design with such a simple metaphor. Any discrepancies will be easy to identify. However this simplicity is deceptive, if one tries to simplify it even more, bad things happen.

WS-* services do not have a single metaphor. Web Services Architecture document[4] from W3C describes four architectural models of WS-*, but does not explain how they relate. One of the models is the Resource Oriented Model (which would imply REST), but as their definition of Web services suggests, the systems they consider are limited to various standards: SOAP, WSDL, and others. New capabilities are added in the form of new standards. There is no general description of the relationship between WS-* standards. Their definitions are constrained only by the compliance with SOAP, WSDL, and the XML schema for defining additional "stickers" in the SOAP envelope.

2.3.4 Comparison between REST and WS-* Principles

Choosing the best option when developing a web service is not a matter of luck, but the result of analyzing the advantages and disadvantages of each of the different models, and choosing one that fits the needs of the problem to solve. Two attempts to compare REST and WS-* services at the abstract level are described below:

Pautasso et al. study In the most comprehensive comparison to date, Pautasso et al.[29] compare RESTful and WS-* services on 3 levels: 1) architectural principles, 2) conceptual decisions, and 3) technology decisions.

On the level of *architectural principles*, Pautasso et al. analyze 3 principles (protocol layering, dealing with heterogeneity, and loose coupling) and note that both styles support these 3 principles. However, they can identify only one aspect common to both styles – loose coupling to location (or dynamic late binding). Consequently, they conclude that it's not possible to make a decision at this level and proceed with more detailed analysis. At the level of *conceptual decisions*, they compare 9 different decisions and find that RESTful services require the designer to make 8 of them, versus only 5 for WS-*. However, WS-* have many more alternatives than RESTful services. Finally, in the *technology* comparison, they identify 10 technologies that are relevant to both styles. In this comparison, WS-* once again offer many more alternatives than their RESTful counterparts.

Based on these results, the authors recommend using REST for ad hoc integration and using WS-* for enterprise-level application integration where transactions, reliability, and message-level security are critical. This study illustrates two key difficulties of performing convincing comparisons of broad ideas, such as Web service styles. First, it's difficult to select the most relevant principles to compare. Second, once the principles are selected, it's difficult to identify choices that are shared by the competing ideas.

Pautasso et al do not explain why they selected protocol layering, dealing with heterogeneity, and loose coupling as the only architectural principles to compare. However, in their analysis, *"key -ilities"* (security, reliability) are only mentioned at lowest level of comparison, the technology decisions. Moreover, they shy away from comparing concepts that are relevant at the enterprise level (transactions, reliability, message-level security), even though they cite these very concepts in their concluding recommendation.

The actual comparison has two problems. First, they use the *numbers* of architectural decisions and available alternatives to choose which style is better. But counting is hardly the right metric – not every decision point has the same weight. Second, most decision points on every level have 2 options, 1 for each style, indicating that they actually have nothing in common. Only in a few cases do both styles require a decision on the same question.

Nevertheless, this paper is the best-conducted comparison of principles available today. It's unbiased, thoroughly researched, and it examines multiple points of view.

Richardson and Ruby Study A second comparison of note is presented in the book,

> "RESTful Web Services"[31]. The authors, Richardson and Ruby, discuss the principles that are relevant to all systems available on the Web. Even though their book is biased toward RESTful Web services, the principles they discuss would be a better starting point for making a fair comparison between the two styles.
>
> They identify four system properties of RESTful services: 1) uniform interface, 2) addressability, 3) statelessness, and 4) connectedness. In RESTful Web services, these properties are embodied in resources, URIs, representations, and the links between them. On the contrary, WS-* services exhibits three of these four properties.
>
> Addressability and some form of connectedness are embedded in the WSDL definition of bindings and ports. Many WS-* services are stateless (although it is not an explicit requirement). Having a uniform interface shared by all services is the only property not supported by WS-*. Since these properties are relevant to both, they are a good choice for comparison.
>
> Richardson and Ruby use a similar approach to evaluate how RESTful Web services offer capabilities which are important for enterprise-level integration. They show how to implement transactions, reliability and message-level security (concepts that Pautasso et al mention, but do not discuss) using REST.

From the previous comparisons, it's notable that both styles of Web services possess certain characteristics that guide their design and development, although they are defined in ways that make it difficult to compare them side-by-side, therefore it is important to take into consideration the common characteristics since the election of the correct style, will be fundamental in the performance of the web service.

RESTful Web services (and Web services in general) pose the first big test of the principles of REST, as identified by Fielding. Even though RESTful services might look like typical Web system, they are not, and WS-* services are clearly different. Up until a few years ago, there was a simple dichotomy between REST and WS-*. RESTful services were used only for simple, public services. In contrast, enterprise standards, tools vendors, and the research community were only concerned with WS-* services, but this is no longer the case, and nowadays both styles are being used in all domains.

The latter is caused mainly because REST web services have evolve to become an alternative to RPC based web services. According to Feng et. al.[15], RESTful web services architecture is better in scalability, coupling and performance, and is likely to become the mainstream technology of web services. Considering this, and based on the problem to be solved, REST web services would be the best option.

2.4 Image Compression

2.4.1 Basic Concepts

An image is essentially a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form. A digital image, then, is basically a 2-Dimensional array of pixels.

Images form the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. The use and dependence on information and computers continue to grow, and so does the need for efficient ways of storing and transmitting large amounts of data.

Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements. Compression is achieved by the removal of one or more of the three basic data redundancies: coding redundancy, interpixel redundancy and psychovisual redundancy[37].

Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. An inverse process called decompression (decoding) is applied to the compressed data to get the reconstructed image. The objective of compression is to reduce the number of bits as much as possible, while keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible. Image compression systems are composed of two distinct structural blocks : an encoder and a decoder.

Some benefits of using compression on images are:

- It provides a potential cost savings associated with sending less data over switched telephone network where cost of call is really usually based upon its duration.
- It not only reduces storage requirements but also overall execution time.
- It also reduces the probability of transmission errors since fewer bits are transferred.
- It also provides a level of security against illicit monitoring.

2.4.2 Types of Redundancy

As stated before, the compression is based on the removal on one or more of the following redundancies:

Coding Redundancy is present when less than optimal code words are used. A code is a system of symbols (letters, numbers, bits, and the like) used to represent a body of information or set of events. Each piece of information or events is assigned to a sequence of code symbols, called a code word. The number of symbols in each code word is its length. The 8-bit codes that are used to represent the intensities in the most 2-D intensity arrays contain more bits than are needed to represent the intensities.

Interpixel Redundancy Also known as spatial or temporal Redundancy, is based on the fact that, as the pixels of most 2-D intensity arrays are correlated spatially, information is unnecessarily replicated in the representations of the correlated pixels. In a video sequence, temporally correlated pixels also duplicate information.

Psychovisual Redundancy Most 2-D intensity arrays contain information that is ignored by the human visual system and extraneous to the intended use of the image. It is redundant in the sense that it is not used. Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible.

2.4.3 Image Compression Techniques

Image compression is applied to reduce the amount of data required to represent a digital image. Image compression is broadly classified into lossless and lossy image compression[42].

Lossless Compression Tecniques The feature of the lossless compression technique is that the original image can be perfectly recovered from the compressed image. It is also known as entropy coding since it use decomposition techniques to eliminate or minimize redundancy[19]. Lossless compression is mainly used for applications like medical imaging, where the quality of image is important. The following are the methods that fall under lossless compression:

- Run length encoding.
- Huffman encoding.

- LZW coding.
- Area coding.

Lossy Compression Techniques Lossy compression technique provides higher compression ratio than lossless compression. In this method, the compression ratio is high; the decompressed image is not exactly identical to the original image, but close to it. Different types of lossy compression techniques are widely used, characterized by the quality of the reconstructed images and its adequacy for applications. The quantization process applied in lossy compression technique results in loss of information. After quantization, entropy coding is done like lossless compression. The decoding is a reverse process. The entropy decoding is applied to compressed data to get the quantized data. Dequantization is applied to it and finally the inverse transformation is performed to get the reconstructed image. The methods that fall under lossy compression technique are listed below:

- Vector quantization.
- Fractal coding.
- Block truncation coding.
- Sub band coding.
- Transformation coding.

2.4.4 Lossless Compression VS Lossy Compression

In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Under normal viewing conditions, no visible loss is perceived (visually lossless)[10].

2.5 Experimental Designs

2.5.1 Introduction

The experimental design is a key piece in the software development process given that it allows identifying failures in the software before it begins to operate. A good strategy to test a software involves the generation of the whole set of cases that participate in its operation. However, testing all the possible cases of a program requires a great amount of time, which can be inadmissible even for small programs[8]. For example, in the scenario of a system with 10 parameters having 5 different values there are 9,765,625 different cases. In general, the number of cases is exponential in the number of parameters.

An alternative strategy to test software is the use of Covering Arrays (CAs). A Covering Array (CA) is a combinatorial object that, with a small number of cases, covers a certain level of interaction of a set of parameters. The meaning of a level of interaction relates any subset of t parameters of a matrix to the set of the v^t different combinatios derived from v different values. The confidence level of the testing using combinatorial objects as CA increases with the interaction level involveld [25].

2.5.2 Covering Arrays

The Covering Arrays (CAs) are mathematical objects with minimal coverage and maximum cardinality that are a good tool for the design of experiments. A covering array is an $N \times k$ matrix over an alphabet v each $N \times k$ subset contains at least one time each combination from $0, 1, \ldots, v - 1^t$, given a positive integer value t.

Covering Arrays have been an object of study and application in different research areas. Cawse [7] used CAs in the material design, Hedayat et al. [21] used them in medicine and agriculture; in biology and industrial processes have also been used by Shasha et al. [36] and Phadke[30]. CAs have been used in hardware testing [41], but significantly the area with the major application of these objects is in software testing[6, 45].

Let N, t, k, and v be positive integers with $t \leq k$. A covering array denoted by CA(N; t,k,v), of alphabet v and strength t is an array $\mathcal{M}$ of size $N \times k$, where each element $m_{i,j}$ takes values from the set $S = \{0, 1, 2, \ldots, v - 1\}$ and each subset of $\mathcal{M}$ of size $N \times t$ contains all the possible combinations derived from $\{0, 1, \ldots, v - 1\}^t$ symbols. A subset of size $N\times$ is known as a t-tuple.

To illustrate the CA approach applied to the design of software testing, consider the Web-based system example shown in Table 2, the example involves four parameters each with three possible values.

	Browser	**OS**	**DBMS**	**Connections**
0	Firefox	Windows	MySQL	ISDN
1	Chromium	Ubuntu	PostgreSQL	ADSL
2	Netscape	Red Hat	MaxDB	Cable

Table 2: Parameters of Web-based system example

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 1 & 0 & 2 \\ 2 & 2 & 1 & 0 \end{pmatrix}$$

Figure 8: Test-suite covering all 2-way interactions, CA(9; 2, 4, 3)

A full experimental design ($t = 4$) should cover $3^4 = 81$ possibilities, however, if the interaction is relaxed to $t = 2$ (pair-wise), then the number of possible combinations is reduced to 9 test cases (which represents a reduction of the 90 percent in the number of cases required). Figure 8 shows the CA corresponding to CA(9; 2, 4, 3); given that its strength and alphabet are $t = 2$ and $v = 3$, respectively, the combinations that must appear at least once in each subset of size $N \times 2$ are {{0, 0}, {0, 1}, {0, 2}, {1, 0}, {1, 1}, {1, 2}, {2, 0}, {2, 1}, {2, 2}}.

Finally, to make the mapping between the CA and the Web-based system, every possible value of each parameter in Table 2 is labeled by the row number. Table 3 shows the corresponding pair-wise test suite; each of its nine experiments is analogous to one row of the CA shown in Fig. 8.

When a CA contains the minimum possible number of rows, it is optimal and its size is called the *Covering Array Number* (CAN). The CAN is defined according to

$$\mathrm{CAN}(t, k, v) = \min\{N : \exists \mathrm{CA}(N; t, k, v)\}$$

Experiments				
1	Firefox	Windows	MySQL	ISDN
2	Firefox	Ubuntu	PostgreSQL	ADSL
3	Firefox	Red Hat	MaxDB	Cable
4	Chromium	Windows	PostgreSQL	Cable
5	Chromium	Ubuntu	MaxDB	ISDN
6	Chromium	Red Hat	MySQL	ADSL
7	Netscape	Windows	MaxDB	ADSL
8	Netscape	Ubuntu	MySQL	Cable
9	Netscape	Red Hat	PostgreSQL	ISDN

Table 3: Mapping the CA(9; 2, 4, 3) to the corresponding pair-wise test suite

The trivial mathematical *lower bound* for a covering array is $v^t \leq \mathrm{CAN}(t, k, v)$, however, this number is rarely achieved. Therefore, determining achievable bounds is one of the main research lines for CAs. Given the values of t, k, and v, the optimal CA construction problem (CAC) consists in constructing a $\mathrm{CA}(N; t, k, v)$ such that the value of N is minimized.

Some of the algorithms used to solve the CAC problem are approximated, meaning that rather than constructing optimal CAs, they construct matrices of size close to that value. Some of these approximated strategies must verify that the matrix they are building is a CA. If the matrix is of size $N \times k$ and the interaction is t, there are $\binom{k}{t}$ different combinations which implies a cost of $O(N \times \binom{k}{t})$ for the verification.

For small values of t and v, the verfication of CAs is overcome through the use of sequential approaches; however, in the construction of CAs of moderate values of t, v and k, the time spent by those approaches is impractical. Then, the necessity of parallel or grid strategies to solve the verification of CAs appears.

3 Methodology

This proposal is basically divided into 3 stages:

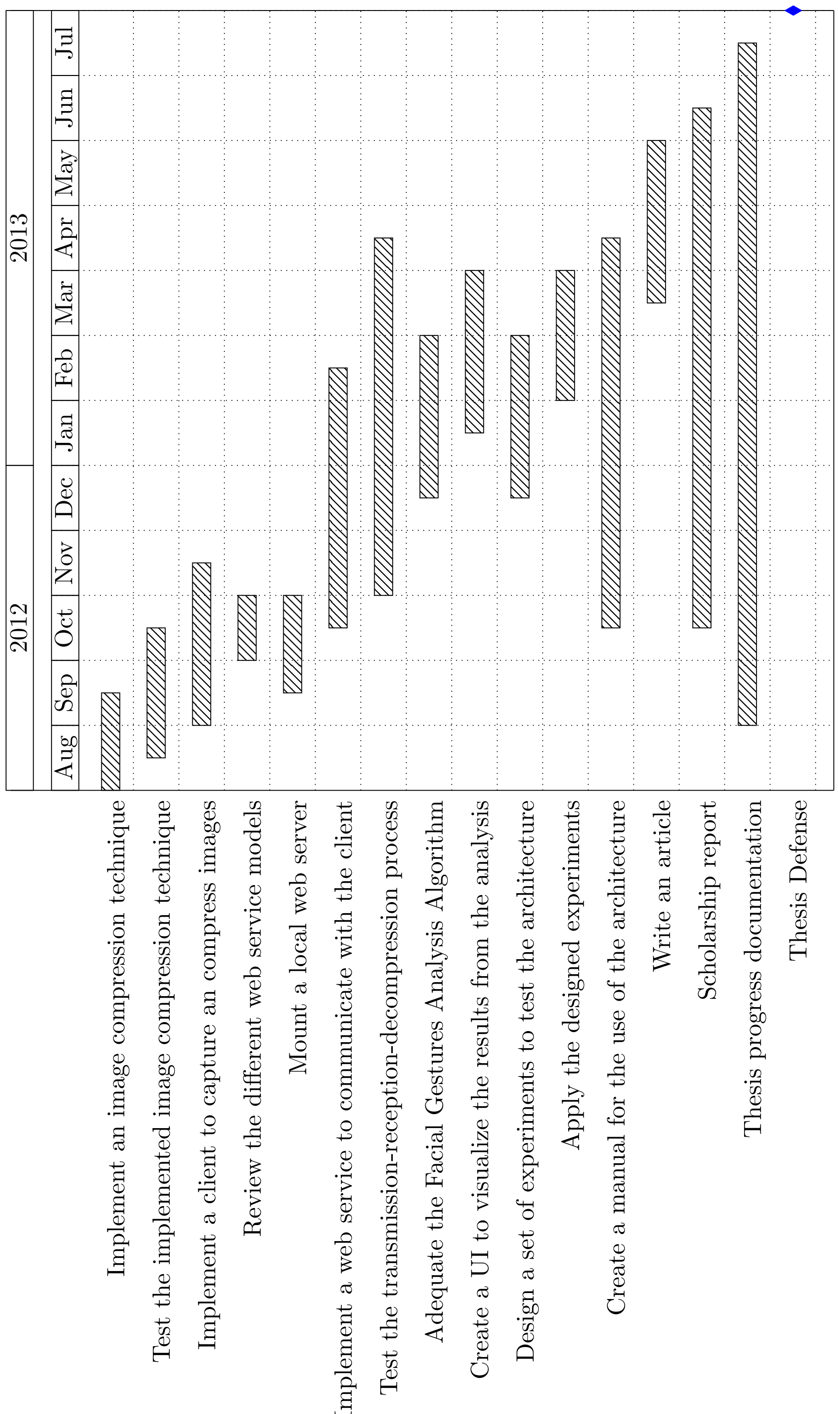
1st Stage In this stage, the different techniques for image compression/decompression will be studied, in order to choose a set of them to be implemented and tested. Once the implementations of the selected techniques have been made, they will be tested and compared to choose only one that will be integrated into a client application. This client application will be responsible of capturing the images for their compression by the selected technique.

2nd Stage In this stage, the different approaches for web services will be analized, to choose the most appropriate solution for the problem to be solved. Also, a local web server will be mounted, to implement the chosen web service model. Once the web service has been implemented, the decompression technique to reverse the technique selected on the previous stage, will be setup on the web service, this, in order to connect the client to the server to send the compressed images, and then testing the process of transmission-reception-decompression. Based on the results from the previous test, the necessary adjustments will be made to ensure both the proper functioning of the connection between the client and server, and the reception of the original image as captured by the client.

3rd Stage In this stage, the work will be focused on the adaptation of the algorithm for facial gestures analysis, since it was originally developed to work in a local environment. This adaptation will include a mechanism to interact with the compression/decompression technique previously developed, and a mechanism to retrieve and process the results from the facial gesture analysis algorithm, also, a user interface for displaying the results of the analysis will be created.

Finally, a set of experiments oriented to test the different components of the architecture (such as compression, quality of results, processing time, etc.) will be developed and applied. Also, a user's manual for the created architecture will be elaborated.

4 Schedule


2012
2013
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Jun
Jul
Implement an image compression technique
Test the implemented image compression technique
Implement a client to capture an compress images
Review the different web service models
Mount a local web server
Implement a web service to communicate with the client
Test the transmission-reception-decompression process
Adequate the Facial Gestures Analysis Algorithm
Create a UI to visualize the results from the analysis
Design a set of experiments to test the architecture
Apply the designed experiments
Create a manual for the use of the architecture
Write an article
Scholarship report
Thesis progress documentation
Thesis Defense

5 Infraestructure

As infrastructure for the development of this work are considered the following components:

- Laptop HP Pavilion DV5-1135la

  OS Ubuntu Linux 11.10 (32 bits).

  Microprocessor AMD Turion X2 Dual-Core Mobile Processor RM-72 (2.1 GHz).

  Memory 3072MB 800MHz DDR2 System Memory (2 Dimm).

  Video Graphics ATI Radeon HD 3450 Graphics (1533MB total graphics memory with 256MB dedicated).

  Hard Drive 320 GB (5400 RPM).

  Webcam Webcam HP Pavilion (VGA low-light)

    Resolution 648×480.

    FPS 30 fps.

- Both the client and the server, will be developed on open source software.

References

[1] Akakin, H. C., and Sankur, B. Robust classification of face and head gestures in video. *Image and Vision Computing 29*, 7 (2011).

[2] Austin, D., Barbir, A., Ferris, C., and Garg, S. Web services architecture requirements. *World Wide Web Consortium Recommendation 2009*, 1/26/2009 (2004).

[3] Berners-Lee, T., Fielding, R., and Frystyk, H. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945 (Informational), May 1996.

[4] Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., and Orchard, D. Web services architecture : W3C working group note 11 february 2004, 2004.

[5] Bruner, J. S., and Tagiuri, R. *The perception of people. In G. Lindzey (Ed)*, vol. 2 of *Handbook of Social Psychology.* Addison-Wesley, 1954.

[6] Burr, K., and Young, W. Combinatorial test techniques: Table-based automation, test generation and code coverage. In *Proceedings of the Intl. Conf. on Software Testing Analysis and Review* (1998), West, pp. 503–513.

[7] Cawse, J. *Experimental design for combinatorial and high throughput materials development.* Wiley-Interscience, 2003.

[8] Cohen, M. B., Colbourn, C. J., and Ling, A. C. H. Augmenting simulated annealing to build interaction test suites. *Software Reliability Engineering, International Symposium on 0* (2003), 394.

[9] Darwin, C. *The expression of the emotions in man and animals.* New York. D. Appleton and Company., 1916. http://www.biodiversitylibrary.org/bibliography/4820.

[10] Dhawan, S. A Review of Image Compression and Comparison of its Algorithms. *International Journal of Electronics & Communication Technology 2*, 1 (March 2011), 22–26.

[11] Ekman, P., and Hager, J. C. Long-distance of transmission of facial affect signals. *Ethology and Sociobiology 1*, 1 (1979), 77–82.

[12] Fang, T., Zhao, X., Ocegueda, O., Shah, S. K., and Kakadiaris, I. A. 3D/4D facial expression analysis: An advanced annotated face model approach. *Image and Vision Computing* (2012).

[13] Fasel, B., and Luettin, J. Automatic facial expression analysis: A survey. *PATTERN RECOGNITION 36*, 1 (1999), 259–275.

[14] Fasel, B., and Luettin, J. Automatic facial expression analysis: a survey. *Pattern Recognition 36*, 1 (2003), 259–275.

[15] Feng, X., Shen, J., and Fan, Y. *REST: An alternative to RPC for Web services architecture*. Ieee, 2009, pp. 7–10.

[16] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. Hypertext transfer protocol – HTTP/1.1 (RFC 2616). Request For Comments, 1999. available at `http://www.ietf.org/rfc/rfc2616.txt`, accessed 7 July 2006.

[17] Fielding, R. T. *REST: Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

[18] Garrett, J. J. Ajax: A New Approach to Web Applications, 2005. Adaptive Path LLC, http://www.adaptivepath.com/publications/essays/archives/000385.php.

[19] González, R., and Woods, R. *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[20] Gottschalk, K., Graham, S., Kreger, H., and Snell, J. Introduction to web services architecture. *IBM Systems Journal 41*, 2 (2002), 170–177.

[21] Hedayat, A., Sloane, N., and Stufken, J. *Orthogonal Arrays: Theory and Applications*. Springer Series in Statistics. Springer, 1999.

[22] Hickson, I. HTML5, Oct. 2010. W3C Working Draft.

[23] Ioannou, S. V., Raouzaiou, A. T., Tzouvaras, V. A., Mailis, T. P., Karpouzis, K. C., and Kollias, S. D. Emotion recognition through facial expression analysis based on a neurofuzzy network. *Neural Networks 18*, 4 (2005), 423–435. Emotion and Brain.

[24] Izhikevich, E. M. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council 15*, 5 (Sept. 2004), 1063–1070.

[25] Kuhn, R., Lei, Y., and Kacker, R. Practical combinatorial testing: Beyond pairwise. *IT Professional 10* (2008), 19–23.

[26] Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks 10* (1996), 1659–1671.

[27] Muschamp, P. An introduction to web services. *BT Technology Journal 22* (2004), 9–18. 10.1023/B:BTTJ.0000015537.71918.c2.

[28] Pautasso, C. SOAP vs. REST - bringing the web back into web services. *Building* (2007), 1–19.

[29] Pautasso, C., and Leymann, F. RESTful web services vs . "big" web services : Making the right architectural decision categories and subject descriptors. *Technology 31*, 7 (2008), 805–814.

[30] Phadke, M. S. *Quality Engineering Using Robust Design*, 1st ed. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1995.

[31] Richardson, L., and Ruby, S. *RESTful Web Services*. O'Reilly Media, 2007.

[32] Rosato, M., Chen, X., and Yin, L. Automatic registration of vertex correspondences for 3D facial expression analysis. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on* (Oct. 2008), pp. 1–7.

[33] Rosenblatt, F. Perceptron simulation experiments. *Proceedings of the IRE 48*, 3 (1960), 301–309.

[34] Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, 1962.

[35] Sergerie, K., Chochol, C., and Armony, J. L. The role of the amygdala in emotional processing: A quantitative meta-analysis of functional neuroimaging studies. *Neuroscience and Biobehavioral Reviews 32*, 4 (2008), 811–830.

[36] Shasha, D. E., Kouranov, A. Y., Lejay, L. V., Chou, M. F., and Coruzzi, G. M. Using combinatorial design to study regulation by multiple input signals. a tool for parsimony in the post-genomics era. *Plant Physiology 127*, 4 (2001), 1590–1594.

[37] Sonal, and Kumar, D. A study of various image compression techniques. *Challenges & Opportunities in Information Technologies* (2007).

[38] Spies, B. Web services, part 1: SOAP vs. REST. *Ajaxonomy* (2008).

[39] Sun, Y., and Yin, L. Facial expression recognition based on 3D dynamic range model sequences. In *Proceedings of the 10th European Conference on Computer Vision: Part II* (Berlin, Heidelberg, 2008), ECCV '08, Springer-Verlag, pp. 58–71.

[40] Tsalakanidou, F., and Malassiotis, S. Real-time 2D+3D facial action and expression recognition. *Pattern Recogn. 43*, 5 (May 2010), 1763–1775.

[41] Vadde, K. K., and Syrotiuk, V. R. Factor interaction on service delivery in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications 22*, 7 (2004), 1335–1346.

[42] Vrindavanam, J., Chandran, S., and Mahanti, G. K. Article: A survey of image compression methods. *IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET 2012) icwet*, 1 (March 2012), 12–17. Published by Foundation of Computer Science, New York, USA.

[43] Wang, P., Köhler, C., Barrett, F., Gur, R. E., Gur, R. C., and Verma, R. Quantifying facial expression abnormality in schizophrenia by combining 2D and 3D features. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA* (2007), IEEE Computer Society.

[44] Wang, X., Hou, Z.-G., Zou, A., Tan, M., and Cheng, L. A behavior controller based on spiking neural networks for mobile robots. *Neurocomputing 71*, 4-6 (2008), 655–666.

[45] Yilmaz, C., Cohen, M. B., and Porter, A. Covering arrays for efficient fault characterization in complex configuration spaces. *SIGSOFT Softw. Eng. Notes 29*, 4 (July 2004), 45–54.

[46] Yin, L., Wei, X., Longo, P., and Bhuvanesh, A. Analyzing facial expressions using intensity-variant 3D data for human computer interaction. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 01* (2006), ICPR '06, IEEE Computer Society, pp. 1248–1251.

[47] Zhao, X., Huang, D., Dellandrea, E., and Chen, L. Automatic 3D facial expression recognition based on a bayesian belief net and a statistical facial feature model. *Pattern Recognition, International Conference on* (Aug. 2010), 3724–3727.