



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

Animación de un rostro tridimensional guiado por reconocimiento facial

NICOLÁS ESTEBAN GARRIDO SILVA

Profesor Guía: RODOLFO ALLENDES

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
12 de marzo de 2018

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Curicó, 2019

Dedicado a mi abuelo Osvaldo Garrido.

AGRADECIMIENTOS

Quisiera agradecer a mis padres por todo su apoyo, a mis hermanos del Hogar de Estudiantes Universitarios de Casa Rodríguez por levantarme el ánimo cuando estaba decaído, a mi amigo Victor por esas gratas e improvisadas conversaciones a las 3am, a “Los patos” y a Daniel Mella por darme asilo en los días de arduo trabajo, y por último a los muchachos de la “Sala de Estudio 1” donde lo último que se hacia era estudiar.

RESUMEN

A lo largo del tiempo la animación de objetos en tercera dimension a ido cobrando mayor importancia. Existen diversos métodos y herramientas que nos permiten lograr el movimiento en los modelos tridimensionales. Caso similar ocurre en el área del reconocimiento facial, que día tras día se van perfeccionando técnicas que ayudan a la detección de rostros.

En este trabajo se pretende desarrollar una prototipo funcional que realice la interacción de ambas técnicas, logrando que un modelo tridimensional imite los gestos que realiza el rostro de una persona.

Para realizar la aplicación propuesta, el proceso de desarrollo se divide en 3 etapas; la investigación de los algoritmos, la implementación y por último la evaluación de la aplicación.

En la primera etapa se buscan técnicas, métodos y algoritmos que ayuden a realizar (por se parado) la detección facial y la deformación del modelo.

La segunda etapa, para el desarrollo de la implementación, se codifico la aplicación con alguna de las técnicas investigadas anteriormente.

La tercera y ultima etapa, la de evaluación, se realizo por medio de una encuesta que mide que tan acertada es la gesticulación del modelo.

ABSTRACT

Over time, the animation of objects in the third dimension has become more important. There are several methods and tools that allow us to achieve movement in three-dimensional models. A similar case occurs in facial recognition, which day after day techniques are created that help the detection of faces.

In this work we try to develop a functional prototype that makes the interaction of both techniques, making a three-dimensional model imitate the gestures made by a person's face.

To make the proposed application, the development process is divided into 3 parts; the investigation of the algorithms, the implementation and finally the evaluation of the application.

In the first part we search techniques, methods and algorithms to help make (separately) facial detection and deformation of the model.

The second part, to develop the implementation, was codified the application with some of the previously investigated techniques.

The third and final part, the evaluation, a survey was made and that measure if the model mimics gestures correctly.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VI
Índice de Tablas	VIII
Resumen	IX
1. Introducción	10
1.1. Contexto del Proyecto	10
1.2. Definición del Problema	11
1.3. Hipótesis	13
1.4. Objetivos	13
1.5. Alcances	14
1.6. Herramientas Ocupadas en el Desarrollo	14
2. Estado del Arte	16
2.1. Trabajo Relacionado	16
2.1.1. Viola-Jones	16
2.1.2. Animación y Morphing de Expresiones Faciales con Métodos Locales de Deformación	17
2.2. Compañías Dedicadas a la Captura de Movimiento	17
2.2.1. BTS Bioengineering	17
2.2.2. Dynamixyz	19
2.2.3. Faceware Technologies INC	19

3. Introducción al Morphing	21
3.1. Qué es Morphing y un Poco de Historia	21
3.2. La Importancia de Morphing	23
3.3. Algunas Técnicas para Realizar Morphing	23
3.3.1. ¿Cómo se Logra la Deformación de la Imagen?	26
3.4. Modelo Utilizado	29
4. Reconocimiento Facial	33
4.1. Animación de un Rostro	34
4.2. Expresiones Faciales	35
4.3. Técnicas de Detección Facial	37
4.3.1. Técnica CLMtracking	37
4.3.2. Técnica de Viola-Jones con CLM	43
5. Diseño y Solución del Problema	47
5.1. Proceso de unión de los algoritmos	47
6. Evaluación de Resultados Obtenidos	58
6.1. Descripción del Método de Evaluación	58
6.2. Resultados de la Encuesta	60
7. Conclusiones y Trabajo Futuro	67
7.1. Análisis de Resultados	67
7.2. Algoritmos Ocupados	69
7.3. Trabajo a Futuro	70
Bibliografía	71
Anexos	
A: Interfaz	75
A.1. Imagenes del Proyecto Finalizado	75

B: Encuesta	80
B.1. Saludo e instrucciones iniciales	80
B.2. Encuesta realizada	87

ÍNDICE DE FIGURAS

	página
1.1. Ejemplo de un mocap-suit	12
2.1. Sistemas optoelectrónicos SMART-DX.	18
2.2. Cámara GoPro utilizada por Dynamixys.	19
2.3. Algunas cámaras ocupadas por Faceware	20
3.1. Deformación simultánea y disolución de una imagen.	22
3.2. Proceso de distorsión de una imagen.	22
3.3. Ilustración de Forward mapping.	24
3.4. Ilustración del algoritmo de Beier-Neely.	25
3.5. Ejemplo de Cross-dissolve.	25
3.6. Una forma de distorsión del morph con interpolación.	26
3.7. Ejemplo de triángulos cocirculares.	28
3.8. Ejemplo de coordenadas baricéntricas.	28
3.9. Malla triangular de una esfera en tercera dimensión.	29
3.10. Modelo Ginger.	31
3.11. Warp-point en un rostro.	32
4.1. Ejemplo de lo que podrían ser los pasos a seguir para el reconocimiento de una cara.	34
4.2. Cara Neutra	36
4.3. Cara Alegre	36
4.4. Cara enojada	36
4.5. Cara sorprendida	36
4.6. Un rostro anotado de la base de datos MUCT.	38
4.7. Entrenando un clasificador con parches de las caras anotadas	39
4.8. El parche recortado y la “respuesta” para el ojo izquierdo	39
4.9. Detecta la cara.	41
4.10. Cuadro delimitador de los ojos y nariz.	41

4.11. Posición de los ojos y nariz.	41
4.12. Coloca el modelo facial.	41
4.13. Ajuste con procrustes.	42
4.14. Arquitectura del sistema.	43
5.1. Máscara con los puntos de warp.	48
5.2. Resultado final.	57
6.1. Gráfico de los porcentajes máximos de cada pregunta.	60
6.2. Porcentaje de Acierto para la emoción de Alegría.	61
6.3. Porcentaje de Acierto para la emoción de Sorpresa.	62
6.4. Porcentaje de Acierto para la emoción de Enojo.	63
A.1. Cara neutra.	75
A.2. Cara enojada.	76
A.3. Cara enojada con ojos cerrados.	77
A.4. Cara feliz.	78
A.5. Cara feliz con boca abierta.	79

ÍNDICE DE TABLAS

	página
6.1. Porcentajes según respuesta.	64
6.2. Resultados esperados con sus respectivos porcentajes.	65

1. Introducción

Este informe presenta el tema desarrollado como proyecto de titulación. Acá se indica la problemática que se desea solucionar con este proyecto. También se plantean las hipótesis, los objetivos y la metodología.

1.1. Contexto del Proyecto

Dada una imagen ¿Cómo podemos encontrar un rostro, ya sea sus ojos, nariz, boca, etc? Por supuesto, hay muchas técnicas que tratan de explicar este proceso. Una de ellas es la exploración rápida de toda la imagen para determinar qué parte de la imagen se ve como una cara, luego concentrarse en esa parte para examinarla más de cerca y así encontrar dónde está la nariz, los ojos y la boca.

En este proyecto de titulación se propone trabajar con el modelamiento de objetos en tercera dimensión, logrando realizar la interacción entre el algoritmo de detección de rostros y el modelo tridimensional, donde este último tiene que imitar los gestos que realiza la persona que fue detectada. Esta aplicación esta pensada para uso libre y en un entorno web, orientada a los desarrolladores que desean incursionar en el mundo de la animación y captura de movimiento facial.

A menudo nos vamos a encontrar con el término morphing. Este ha sido definido como la transformación de formas vía técnicas digitales [12]. Con el morphing podemos realizar la distorsión o transformación de una imagen. Un tipo de Morphing es el Warping. Este consiste en la disposición de una única imagen, la cual es comprimida

y/o estirada pareciendo que tenga características de elasticidad, lo que da el efecto de movimiento que buscamos en este proyecto.

Estos métodos de morphing son guiados por los movimientos que hace el rostro de un ser humano que se muestra mediante una cámara web. En él se realiza la gesticulación facial, y para ello necesitaremos realizar el reconocimiento del rostro.

Existen algunos algoritmos de reconocimiento facial que son en tiempo real. Esto quiere decir que son algoritmos que interactúan activamente y procesan información con un entorno. En este contexto la palabra “tiempo” significa que el correcto funcionamiento de un sistema depende no sólo del resultado lógico que devuelve la computadora, si no que también depende del tiempo en que se produce ese resultado. La palabra “real” quiere decir que la reacción a eventos externos debe ocurrir durante su evolución.

1.2. Definición del Problema

Este proyecto de titulación consiste en el modelamiento de un objeto tridimensional que reproduzca los gestos faciales que realiza una persona por medio de una cámara web [28]. Lo que se desea lograr es la interacción entre el algoritmo de detección de rostros con el algoritmo de deformación que realiza la distorsión del objeto tridimensional, de manera que la gesticulación del objeto en tercera dimension sea lo más fluida y lo más parecido posible a los gestos que se están realizando.

Para poder llevar a cabo este proyecto lo primero que se debe realizar es la detección del rostro que se esta proyectando, para que el objeto tridimensional pueda imitarlo cuando se realicen los movimientos correspondientes. El modelamiento se realiza con un rostro tridimensional y con la ayuda de los algoritmos de morphing. Este algoritmo es el que realiza la distorsión de una imagen logrando dar el efecto de movimiento que necesitamos. Volker Blanz y Thomas Vetter crean algo similar en su trabajo [25].

Existen varios algoritmos para realizar la detección de caras. Acá se espera que el objeto tridimensional imite todos los gestos que realiza el rostro real detectado. El problema que existe en este proyecto radica en cómo realizar la interacción de

estos algoritmos de detección de rostro (real) con el algoritmo encargado de realizar la distorsión del objeto tridimensional (cara virtual).

La deformación de un modelo tridimensional (en término de expresiones faciales) se aplica por lo general en el área de la animación y desempeña también un papel importante en la infografía. Las técnicas de deformación facial podrían aplicarse a diversas aplicaciones (tales como el reconocimiento de rostros, la autenticación, imágenes dinámicas, animación semántica, construcción de la realidad virtual, y la interfaz hombre-máquina inteligente). En este caso es la distorsión o animación de un modelo tridimensional que se representa en una cara virtual.

Hoy en día existen métodos que ayudan a la animación de objetos tridimensionales, pero estos necesitan de la ayuda de un “mocap suit”, Figura 1.1, el cual es un traje de cuerpo completo que tiene una serie de sensores con forma de esferas. La cámara captura la ubicación de estas esferas y al moverse el sujeto que tiene el traje puesto se genera la animación.



Figura 1.1: Ejemplo de un mocap-suit

Con este proyecto se pretende realizar la captura del movimiento de los gestos faciales sin la ayuda de ningún traje o programa especial, de manera que cualquier persona pueda realizar la animación de un objeto tridimensional, sin mencionar que el costo monetario no sería elevado. Es por ello que una vez terminado el proyecto se espera lograr tener una aplicación web de libre uso que ayude a facilitar y agilizar

el proceso de gesticulación de las animaciones en tercera dimensión, con el fin de ayudar a las personas que quieren incursionar en el mundo de la animación pero no tienen los medios para poder hacerlo.

1.3. Hipótesis

- El objeto virtual puede realizar los mismos gestos de un rostro real.
- El objeto tridimensional puede realizar los gestos fluidamente y sin desfases.
- El uso de los algoritmos de morphing facilita la distorsión del objeto 3d.
- El algoritmo de detección de rostros Viola-Jones ayuda a la realización de este proyecto.

1.4. Objetivos

Objetivo General

- Implementar un prototipo funcional que integre el algoritmo de detección de rostros con el modelo tridimensional, logrando que el modelo tridimensional imite los gestos que realiza el rostro de la persona.

Objetivos Específicos

- Especificar el algoritmo que se necesita para realizar la detección de caras.
- Especificar el algoritmo de Morphing que realiza la distorsión del modelo tridimensional.
- Realizar la integración entre los 2 algoritmos mencionados anteriormente.
- Codificar la aplicación.
- Evaluar la aplicación.

1.5. Alcances

- En este trabajo sólo se usarán los algoritmos de morphing existentes para hacer la distorsión del modelo.
- Se ocupan algoritmos de detección de rostros ya existentes.
- En este trabajo no se crean nuevos algoritmos pero puede que se tenga que alterar el algoritmo de detección de rostros o morphing para tener un mayor grado de precisión en el reconocimiento de los gestos faciales.

1.6. Herramientas Ocupadas en el Desarrollo

En el transcurso de este proyecto se trabaja con algunas herramientas de desarrollo. A continuación se dan a conocer las más importantes dentro de ellas.

WebGL: es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos de tres dimensiones en navegadores web. WebGL permite mostrar gráficos de tercera dimensión acelerados por hardware (GPU) en páginas web, sin la necesidad de plug-ins en cualquier plataforma que soporte OpenGL 2.0 u OpenGL ES 2.0. Técnicamente es un API para javascript que permite usar la implementación nativa de OpenGL ES 2.0 que viene incorporada en los navegadores. WebGL es gestionado por el consorcio de tecnología sin ánimo de lucro Khronos Group [11].

JavaScript: es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico, que se incorpora en una página web HTML. Un lenguaje interpretado significa que las instrucciones son analizadas y procesadas por el navegador en el momento en que deben ser ejecutadas [5].

Three.js: Es una biblioteca liviana escrita en JavaScript para crear y mostrar gráficos animados por ordenador en tercera dimensión en un navegador Web [21] y puede ser utilizada en conjunción con el elemento canvas de HTML5, SVG y WebGL.

jQuery: permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es la biblioteca de JavaScript más utilizada [16].

Resumen del Capítulo

La problemática que se plantea en este capítulo consiste en implementar un prototipo funcional que integre el algoritmo de detección de rostros con el modelo tridimensional, logrando que el modelo tridimensional imite los gestos que realiza el rostro de una persona.

La aplicación se realizó en un entorno web y el uso de este software será libre, ya que está pensado para personas que no tienen los medios para pagar programas más elaborados.

Los algoritmos que se integran en este proyecto no se modifican, pero lo más probable es que se tengan que adaptar para lograr la interacción entre el modelo tridimensional y el algoritmo de detección facial.

2. Estado del Arte

En este capítulo se describe brevemente algunos de los trabajos relacionados al proyecto desarrollado, además de las compañías que se dedican a la captura de movimientos faciales y las tecnologías que tienen cada una de ellas.

2.1. Trabajo Relacionado

Para poder realizar la detección de rostros se han creado varios algoritmos, cada uno de estos algoritmos está basado en alguna técnica conocida de reconocimiento de formas como, redes neuronales y vecino más próximo, por mencionar algunos. Cada detector tiene características que lo hacen diferente al resto, uno de los más conocidos es el detector de Viola-Jones.

2.1.1. Viola-Jones

El algoritmo de Viola-Jones es un detector muy rápido, pero no es capaz de calcular la orientación de la cara ni darnos una estimación de lo seguro que está de la detección. El algoritmo hace uso de una cascada de clasificadores muy sencillos que se van ejecutando secuencialmente. La cascada permite desechar gran parte de las regiones de la imagen y sólo concentrarse en las zonas que es más probable encontrar una cara, haciendo que la detección sea bastante rápida. Esto se puede apreciar en el trabajo de Joaquín Planells Lerma, que realiza la implementación del algoritmo de detección facial de Viola-Jones [17] el cual comprende los fundamentos del algoritmo

de detección así como los aportes de otros autores al trabajo de P. Viola y M.J. Jones. También implementa una plataforma configurable para la detección de rostros y devuelve una lista de coordenadas de las caras encontradas.

2.1.2. Animación y Morphing de Expresiones Faciales con Métodos Locales de Deformación

En inglés es “Facial Expression Morphing and Animation with Local Warping Methods”. Este trabajo fue realizado por Daw-Tung Lin y Han Huang [27], donde estudia la distorsión de una imagen con warping de manera local mediante dos métodos: Linear Piecewise Polynomial (LPP) y Radial basis functions (RBF).

2.2. Compañías Dedicadas a la Captura de Movimiento

Es importante mencionar la existencia de compañías que se dedican a la captura de movimientos, cada una de ellas con sus propias técnicas y herramientas tecnológicas.

2.2.1. BTS Bioengineering

BTS Bioengineering es una compañía que cuenta con un sistema de captura de movimiento. Trabaja con una tecnología denominada SMART-DX (Figura 2.1), la cual consiste en un sistema de optoelectrónicos para apoyar las necesidades de análisis de movimiento en el ámbito clínico, deportivo e industrial. Los modelos tienen cámaras digitales, con iluminadores de infrarrojos para garantizar los resultados en las condiciones más críticas.



Figura 2.1: Sistemas optoelectrónicos SMART-DX.

Su misión es “Estamos conscientes de que existe una estrecha relación entre la capacidad motriz del individuo y la calidad de su vida. Creemos que sólo a través de una evaluación objetiva del movimiento podemos definir una estrategia para mejorar el rendimiento motor y trabajamos diariamente para que esto se realice. Desde 1986, hemos desarrollado tecnologías innovadoras para analizar y medir el movimiento, ofreciendo a nuestros clientes soluciones adaptadas a sus necesidades reales.”

En el ámbito médico evalúan la disfunción motora y definen los mejores tratamientos, obteniendo un análisis cuantitativo, detallado y no invasivo de los parámetros biomecánicos y neuromusculares del paciente. También incluyen la integración de sistemas de realidad aumentada, sincromiografía inalámbrica, plataformas eléctricas y sistemas optoelectrónicos en el área deportiva, todo esto para proporcionar un análisis multifactorial exhaustivo de cualquier gesto atlético.

Por último, los institutos científicos y universitarios de todo el mundo utilizan los sistemas BTS para realizar investigaciones en los campos de Biomecánica, Biometría, Diagnóstico y Ciencias Motrices [7].

2.2.2. Dynamixyz

Dynamixyz se especializa en análisis y síntesis de rostros tridimensionales. Ha diseñado sistemas de animación, proporcionado renderizado fotorrealista de la piel en tiempo real y desarrollo sistemas avanzados de captura de movimiento. Se especializan en capturar el rostro humano y es uno de los sistemas de seguimiento facial sin marcas más estables del mundo [8].

Hace uso de toda la imagen facial para seguir los movimientos y se diferencia de los sistemas tradicionales de captura de movimiento basados en marcadores, ya que otras aplicaciones capturan un número muy limitado de posiciones de marcadores. Acá cada píxel de la cara se usa como fuente de información, proporcionando una solución de animación en tiempo real y de alta calidad por medio de la cámara que se muestra en la Figura 2.2.

Si uno se registra en su página web puede acceder a 1 mes de prueba, ver videos tutoriales y documentación en línea.



Figura 2.2: Cámara GoPro utilizada por Dynamixys.

2.2.3. Faceware Technologies INC

Faceware Technologies INC es una empresa dedicada a la captura de movimiento facial completamente sin marcadores. Sus productos se han utilizado en premios Oscar, series web independientes y video juegos. Su software es en tiempo real y está disponible como producto independiente, complementando el paquete de animación y motor de juego común de Autodesk [13].

Tienen un sistema de cámara de video, como una DSLR, GoPro, Webcam, cámaras montadas en la cabeza sin marcadores (Figura 2.3) ideal para el software de captura de movimiento facial, además de una fácil sincronización con cualquier sistema de cuerpo mocap.

Los sistemas completos de Faceware incluyen todo el software y hardware que necesita para crear el sistema de captura de movimiento facial desde \$ 2,495 hasta \$ 29,995 USD.



Figura 2.3: Algunas cámaras ocupadas por Faceware

Resumen del Capítulo

Hablamos de algunas técnicas y algoritmos que se ocupan en la actualidad, algunos de ellos son más fáciles de implementar pero no satisfacen todos los requerimientos de este proyecto.

Se mencionaron también algunas empresas que se dedican al desarrollo de software de captura de movimiento facial y la tecnología que ocupan. Algunas de estas empresas cuentan con un período de prueba y en otras simplemente se tiene que pagar, cada una cuenta con su propia aplicación de escritorio.

Recordemos que este proyecto está orientado al uso libre y su desarrollo está orientado a ser usado en una página web.

3. Introducción al Morphing

Como se mencionó anteriormente, en este proyecto se propone trabajar con el modelamiento de objetos en tercera dimensión, por lo que a menudo nos vamos a encontrar con el término de Morphing. Es por ello que en este capítulo se explica en que consiste este algoritmo, la importancia de él y que técnicas existen. Este algoritmo trabaja con puntos de warp y mallas triangulares, es por ello que también se abordan estos temas.

3.1. Qué es Morphing y un Poco de Historia

La palabra “Morphing” viene de la palabra “metamorfosis”. El concepto de Morphing es la abreviatura del anglicismo metamorphosing, el cual proviene de la palabra griega morfe que a su vez significa cambiar de forma, aspecto o forma. Morphing se realiza acoplando warping con interpolación de color y este consiste en la disposición de una única imagen. El Morphing ha sido definido como la transformación de formas vía técnicas digitales [15] donde podemos realizar la distorsión o transformación de una imagen, es decir, es el proceso cuando la imagen original se distorsiona progresivamente y se desvanece mientras se produce la imagen de destino, la cual es comprimida y/o estirada pareciendo que esta tenga características de elasticidad, dando un efecto de movimiento [27]. Las imágenes que siguen en la secuencia son similares a la imagen de origen, y la última imagen es similar a la imagen objetivo, las imágenes del medio de la secuencia son el promedio de la imagen original y la imagen de destino [10].

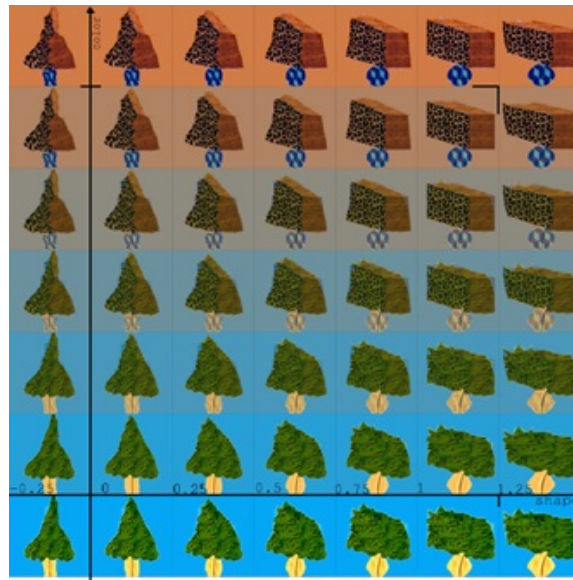


Figura 3.1: Deformación simultánea y disolución de una imagen.

Morphing fue inventado y utilizado por primera vez por Industrial Light and Magic. El autor original del primer algoritmo de morphing es Douglas B. Smythe, esto se puede observar en su artículo [23]. Este tipo de morphing es técnicamente nada más que una deformación simultánea y disolución (ver Figuras 3.1 y 3.2) de una imagen. También existe otro tipo de morphing, que es mucho más complejo ya que utiliza modelos tridimensionales de las dos imágenes que se transforman [10].



Figura 3.2: Proceso de distorsión de una imagen.

El primer uso comercial de morphing estaba en una secuencia en la película *Willow* y desde entonces morphing ha sido ampliamente utilizado. Entre las secuencias

más memorables de Morphing son las que se encuentran en el video de Michael Jackson “Black or White” y en la película Terminator 2 [1].

3.2. La Importancia de Morphing

Como se mencionó anteriormente el Morphing es una tecnología para producir efectos especiales en imágenes. Morphing es común en la industria del entretenimiento y es ampliamente utilizado en las películas, juegos de animación, etc. Además de la utilización de la industria del entretenimiento, morphing se puede utilizar en entrenamientos informáticos, ilustraciones de libros electrónicos, presentaciones, efectos de educación etc. Además la industria de la animación está en una constante búsqueda de tecnologías avanzadas para producir efectos especiales en sus películas [1]. Es por ello que aquí morphing toma una gran importancia.

3.3. Algunas Técnicas para Realizar Morphing

Las técnicas de morphing en una imagen se pueden clasificar en dos categorías, de métodos basados en malla (mesh-based) y métodos basados en funciones (feature-based) para especificar características. En los métodos basados en malla, las características en una imagen son especificadas por una malla no uniforme. En los métodos basados en funciones se especifican las características con un conjunto de puntos o segmentos de línea [3]. A continuación se presentan más técnicas de morphing.

Warping

Warping es una transformación geométrica de dos dimensiones y genera una distorsión cuando es aplicada a una imagen, lo que significa que se aplica una deformación a la misma imagen. Existen dos maneras de deformar una imagen: Forward mapping (Mapeo delantero) y Reverse mapping (Mapeo inverso) [6]. En **Forward mapping** cada pixel en la imagen original es mapeado a un pixel apropiado en la imagen de destino, ver Figura 3.3, pero algunos pixeles de la imagen de destino puede

que no sean mapeados. Existe dos desventajas en este método: los vacíos y los solapamientos. Los vacíos se producen cuando la función de transformación no recibe un pixel de entrada pero esta sí tiene pixeles de salida. Los solapamientos son cuando un pixel de salida tiene asociado más de un pixel de entrada.

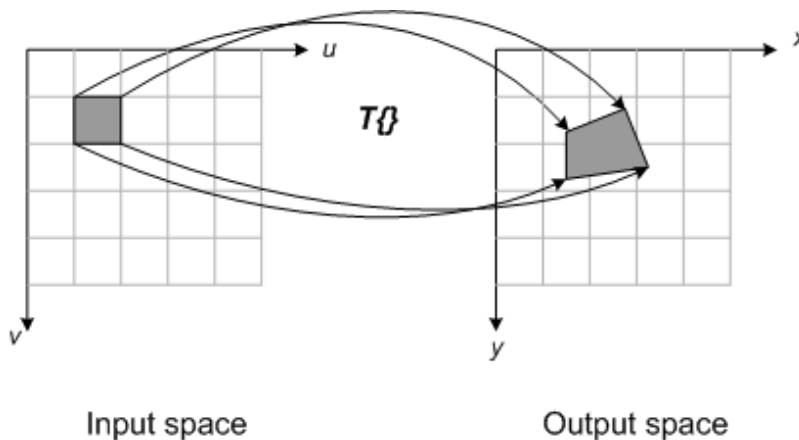


Figura 3.3: Ilustración de Forward mapping.

El método **Reverse Mapping** pasa por cada píxel de la imagen original y todos los píxeles de la imagen de destino se asignan a algún píxel de la imagen original y esta asignación se utiliza en el método de línea de morphing Beier/Neely. El algoritmo de **Beier/Neely** realiza una interpolación en las líneas y no en los puntos, define el warp como el promedio ponderado de cada línea de warp. Al crear el warp para cada línea, el píxel a ser deformado (X en la Figura 3.4) se representa primero en una base con ejes paralelos y perpendiculares a la línea PQ . La coordenada temporal es (u, v) , donde u representa el porcentaje a lo largo de la línea y v representa la distancia de la línea. Para obtener la nueva coordenada, obtenemos el punto X' usando (u, v) y la base para la nueva línea $P'Q'$. Esto genera un warp por línea, donde cada uno de ellos es una simple rotación y una escala no uniforme (la escala solo se realiza a lo largo del eje de la línea). Estas deformaciones deben ser promediadas para obtener el warp final [26].

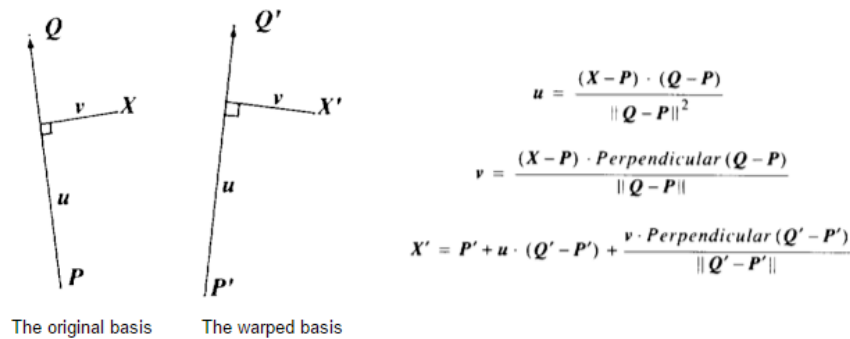


Figura 3.4: Ilustración del algoritmo de Beier-Neely.

Cross-dissolve

Una forma de lograr el efecto morphing es transformar una imagen en otra por medio de **cross-dissolve** que es la transición o el cómo se disuelve la imagen origen a la imagen final como se muestra en la Figura 3.5. En este método, el color de cada pixel se interpola según el transcurso del tiempo desde el primer valor de la imagen con el valor correspondiente de la segunda imagen. Sin embargo, esto no es muy eficaz en una metamorfosis entre dos caras que no tienen aproximadamente la misma forma.



Figura 3.5: Ejemplo de Cross-dissolve.

Interpolación

Otra manera de lograr morphing es por medio de la **interpolación**, esta se realiza mediante la combinación de warps con la interpolación. Las características de dos imágenes y sus correspondencias son especificadas por un animador con un conjunto

de puntos o segmentos de línea. Entonces, el warp se calcula para distorsionar las imágenes y así las funciones tienen posiciones y formas intermedias.

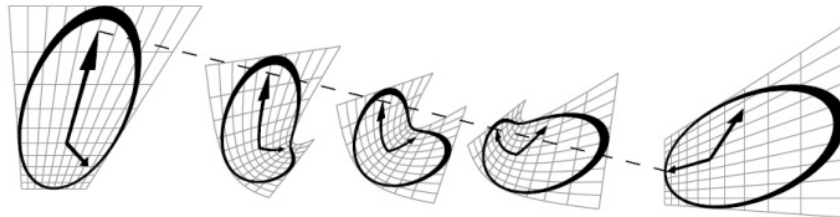


Figura 3.6: Una forma de distorsión del morph con interpolación.

Interpolando linealmente las dos perspectivas del reloj, provoca un efecto geométrico de flexión en las imágenes “entre medio”, Figura 3.6. La línea discontinua muestra la trayectoria lineal de una función en el transcurso de la transformación. Este ejemplo es uno de los tipos de distorsión que puede surgir en las técnicas de morphing de imágenes.

3.3.1. ¿Cómo se Logra la Deformación de la Imagen?

En morphing la tarea más difícil es la deformación de una imagen en otra [22]. El estiramiento y contracción de las imágenes hace que el efecto de morphing sea realista. El morphing de la imagen se puede lograr mediante el uso de algunos métodos, tales como puntos de morph o líneas de morph. A continuación se da una breve reseña de algunos de ellos:

Puntos de Morph: son los marcadores que establecen el inicio y el final en la imagen. El programa de morphing utiliza estos marcadores para calcular cómo se debe doblar o deformar la imagen inicial para que coincida con la forma de la imagen final.

Líneas de Morph: el segundo método utiliza líneas (bordes) en lugar de puntos individuales.

Ambos métodos producen efectos de morphing muy realistas. Una de las tareas que más tiempo consume en morphing es la selección de los puntos o líneas en la imagen inicial y final para que la metamorfosis sea suave y natural. Hay varios consejos útiles a tener en cuenta cuando se desea hacer morphing en objetos. El primero es elegir cuidadosamente las fotos a las cuales se les va a aplicar la transformación. Por ejemplo, si desea transformar dos animales, lo mejor es utilizar imágenes que tienen el mismo tamaño y contorno. Siguiendo el ejemplo de la Figura 3.2, que muestra un primer plano de la cabeza, lo ideal sería que la otra imagen también tenga un primer plano de la cabeza para obtener resultados exitosos. Otro consejo es seleccionar cuidadosamente el fondo; si se utiliza un único color de fondo, el efecto morphing se centra en el objeto. Idealmente, lo mejor es usar el mismo fondo para cada imagen [22].

Ahora que ya definimos qué son los puntos y líneas de morph hablaremos de técnicas un poco más avanzadas que ocupan los métodos previamente mencionados, la triangulación y las mallas triangulares.

Triangulación

Dada una nube de puntos, una triangulación es una forma de unir tripletas de puntos tal que se forme una red de triángulos que no se superponen los unos a los otros. Existe un tipo de triangulación especial llamada **Triangulación de Delaunay (TDL)**, en la cual siempre se trata de formar triángulos lo más balanceados posible, donde no existen dos triángulos contiguos que sean cocirculares (Figura 3.7). Para poder ejecutar este algoritmo se necesita tener 3 triángulos como base y estos triángulos tienen que tener todos los posibles puntos de la TDL final, además de un sistema de coordenadas respecto de un triángulo llamada coordenadas Baricéntricas (Figura 3.8) [30].

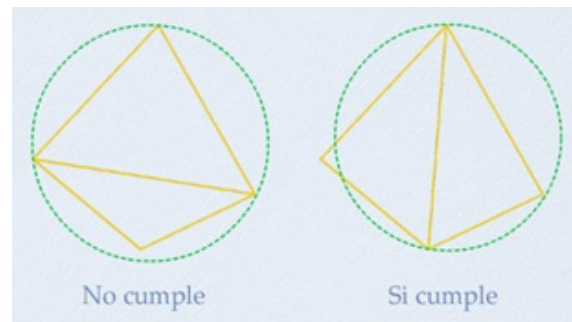


Figura 3.7: Ejemplo de triángulos cocirculares.

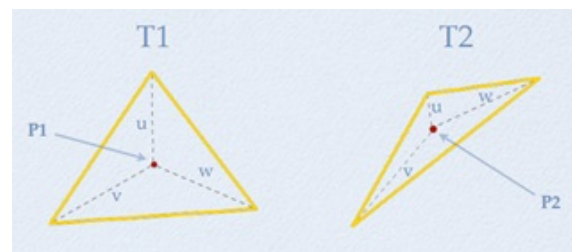


Figura 3.8: Ejemplo de coordenadas baricéntricas.

Mallas triangulares

Una vez que se tienen los puntos de morph, y se crean las líneas, se genera lo que se llama una malla triangular. Una malla triangular es un tipo de malla poligonal que comprende un conjunto de triángulos, en nuestro caso en tres dimensiones (Figura 3.9) que se conectan por sus lados o vértices comunes y aproximan una superficie. Aunque el campo de aplicación de la generación automática de mallas triangulares ha sido tradicionalmente la obtención de modelos digitales de elevaciones de terreno, hoy en día son usadas en renderización, animación, realidad virtual y en análisis de elementos finitos [19].

Uno de los elementos asociados a los vértices de una malla triangular es el plano promedio. Dado que tres puntos en el espacio definen un plano, entonces cada triángulo de la malla tiene un plano asociado. Así, para el vértice “v” hay un plano por cada triángulo incidente. El plano promedio es entonces el “promedio” de estos planos

asociados. Este plano se usa en todos los procesos de re-triangulación. El problema de hallar el plano promedio asociado a un vértice “v” se entiende como el problema de hallar el plano que minimiza la distancia del conjunto de vértices de la vecindad al conjunto de planos en el espacio [2]. Para encontrar tal plano se usan comúnmente dos técnicas: ajuste por mínimos cuadrados y análisis de componentes principales (ACP). Esta última es una técnica usada en estadística para reducir la dimensionalidad [29] .

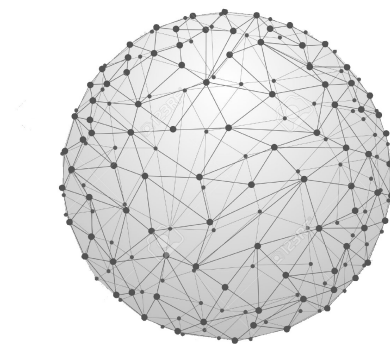


Figura 3.9: Malla triangular de una esfera en tercera dimensión.

3.4. Modelo Utilizado

El modelo que se ocupó en este proyecto fue un demo de una aplicación web hecha en WebGL, que lleva por nombre “Ginger”, que se muestra en la Figura 3.10 y fue creada por Stickman Ventures, una compañía global enfocada en ayudar a organizaciones a desarrollar software personalizado. El modelo fue lanzado el año 2011 cuando WebGL tenía una larga lista negra de controladores, cuando terminaron el demo completo recibieron un gran apoyo de la gente que disfrutaba de su producto logrando estar en la página principal de Chrome Experiments, por lo que decidieron crear una mejora.

El Ginger actual cuenta con lo mejor de web, utiliza una shell de aplicación ¹ y server worker ² por medio de platinum-sw [20] que permite ser ocupado sin conexión a Internet, utiliza también la capacidades de Google App Engine con HTTP/2 Push [4], además tiene un “manifest.json” para la compatibilidad de la pantalla de inicio, se puede tomar y guardar una captura de pantalla, un CSS y una shell más rápida para la primera carga. Con todo eso el modelo ofrece una experiencia mejorada, el caché es tan rápido que permite un mejor rendimiento del usuario en una amplia gama de dispositivos.

El modelo cuenta con un selector que determina que parte se desea modificar, los elementos que se pueden modificar son:

Ojos: Sirve para cerrar o abrir los ojos.

Expresión: Hace que el modelo cambie de expresión de enojado a feliz.

Abrir la Mandíbula: Como su nombre lo indica abre la boca.

Torsión de la Mandíbula: Mueve la mandíbula de izquierda a derecha.

Simetría: Mueve la mitad izquierda del rostro.

Mover Labio [1]: Mueve solo los labios hacia arriba o hacia abajo.

Mover Labio [2]: Hace que los labios se muevan hacia el centro.

Estructura del Rostro: Cambia la forma del rostro, haciendo que el modelo sea más delgado o más tosco.

¹Shell de aplicación (o shell de app) es una forma de crear una Progressive Web App que se carga al instante y de manera confiable en la pantalla de tu usuario, en forma similar a lo que ves en las apps nativas.

²Un service worker es una secuencia de comandos que tu navegador ejecuta en segundo plano, separado de una página web, abriéndoles la puerta a funciones que no necesitan una página web ni interacción de usuario.

Anchura de Mandíbula: Como su nombre lo indica cambia el ancho que tiene la mandíbula.

Lengua: Mueve la lengua de arriba hacia abajo pero sin sacarla para afuera.



Figura 3.10: Modelo Ginger.

Para lograr la animación del objeto tridimensional se va hacer uso de warping, donde se especifican una serie de puntos que guían los movimientos de la imagen. Estos puntos los llamaremos warp-point o puntos de deformación (Figura 3.11) y tienen una ubicación específica según a que parte del rostro están representando. Al modificar las coordenadas se ve la distorsión del objeto tridimensional.

Mientras más warp-point tenga la imagen tridimensional, más exacta debería ser la animación del objeto tridimensional respecto a los movimientos originales.

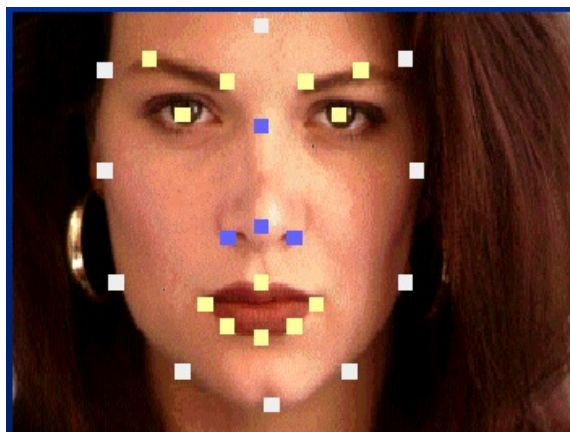


Figura 3.11: Warp-point en un rostro.

Resumen del Capítulo

El Morphing es la distorsión o transformación de una imagen, se comprime o estira para que parezca que tiene propiedades de elasticidad. Se usa principalmente en las industrias de entretenimiento, en las películas y los vídeos juegos, pero también es ocupado en el área de salud, ilustraciones de libros electrónicos, y para efectos educativos, entre otros.

Existen diversas técnicas para realizar morphing, entre ellas se mencionan las siguientes: Waring que ocupa el método Reverse Mapping y el algoritmo Bier/Neely, Cross-Dissolve para transformar una imagen a otra por medio de una transición, Interpolación que se realiza mediante combinaciones de hilos de deformaciones.

La deformación se logra gracias a los puntos y las líneas de morph, pero existen métodos más sofisticados que ocupan estos elementos: puntos y líneas de morph, ellos son la Triangulación y las Mallas Triangulares.

Se presentó el modelo que se va a utilizar, Ginger, el cual fue creado en WebGL por la compañía Stickman Ventures donde se pueden mover los elementos de la cara tales como los ojos, la mandíbula, los labios, la lengua, entre otros, y lograr el efecto de imitar las emociones leídas por la detección.

4. Reconocimiento Facial

Como se espera que el objeto tridimensional imite los gestos de un rostro humano, hablaremos también del reconocimiento facial y las expresiones que se desean capturar.

El reconocimiento facial busca identificar automáticamente a una persona. Esto es posible mediante un análisis de las características faciales del sujeto. Estas características son extraídas de la imagen o de una fuente de video, y comparadas con una base de datos. Para identificar una cara en una imagen o secuencia de video es necesario detectar en qué posiciones de la imagen es más probable que haya una cara [17]. El objetivo de un sistema de reconocimiento facial es, generalmente, el siguiente: dada una imagen de una cara “desconocida”, o imagen de prueba, encontrar una imagen de la misma cara en un conjunto de imágenes “conocidas”, o imágenes de entrenamiento. En la Figura 4.1 se muestra los posibles pasos a seguir para el reconocimiento de un rostro. La gran dificultad añadida es conseguir que este proceso se pueda realizar en tiempo real. El sistema debe identificar las caras presentes en imágenes o videos automáticamente; por ello un detector de caras debería ser capaz de encontrar todas las posibles caras de una imagen y además debe devolver para cada cara detectada en la imagen la posición y tamaño en donde sea que encuentren los ojos, la nariz, la boca y demás elementos. En el caso particular de esta aplicación el detector encuentra una sola cara, hablaremos de ello más adelante.

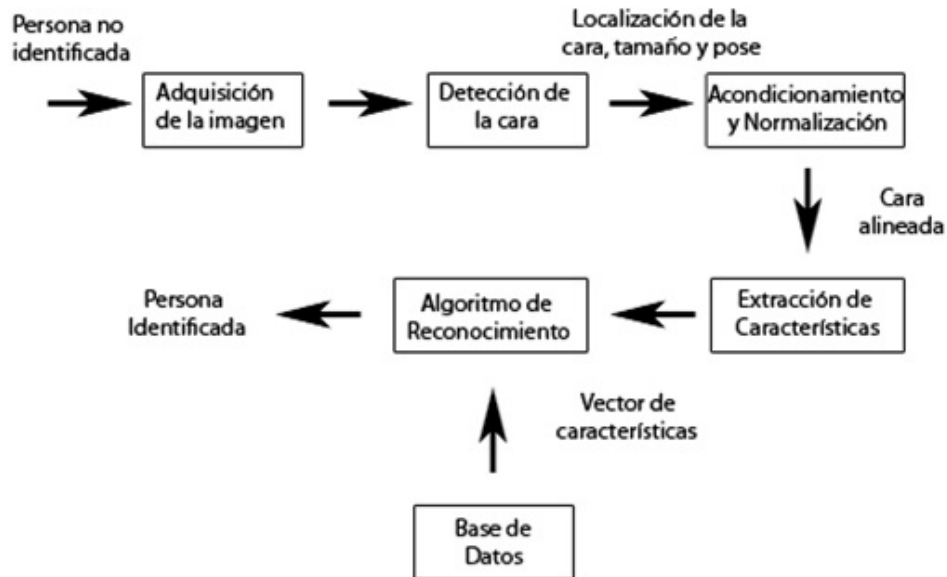


Figura 4.1: Ejemplo de lo que podrían ser los pasos a seguir para el reconocimiento de una cara.

4.1. Animación de un Rostro

La cara es la parte del personaje donde tenemos la más compleja y variada cantidad de matices expresivos. Resulta fundamental que la expresión sea clara y que todos los elementos (boca, ojos, cejas, etcétera) de una expresión nos diga qué está pasando por la cabeza del personaje en cada momento.

En la animación se suele distinguir entre dos métodos de animación: animación directa y animación de pose a pose. La animación directa consiste en comenzar en el inicio de una animación y proseguir hacia delante, creando cada nuevo fotograma a medida que se desarrolla el movimiento. La animación de pose a pose consiste en colocar poses específicas en lugares cuidadosamente considerados sobre la línea de tiempo y después rellenar los espacios intermedios con poses intermedias. Generalmente la mejor forma de trabajar es con alguna combinación de estos dos enfoques [17].

4.2. Expresiones Faciales

Como ya se dijo, la cara es la parte más expresiva del cuerpo humano, el canal más importante de comunicación no verbal, y además es una buena interfaz para comunicar porque las personas tienen capacidad de reconocer expresiones e inferir de ellas estados emocionales [18]. Paul Ekman, el mayor experto en comunicación no verbal, ha pasado más de 40 años investigando los gestos faciales y las emociones humanas, y ha logrado detectar las expresiones faciales básicas, las cuales son: alegría, tristeza, ira, sorpresa y miedo, estas son universales, independientemente a la sociedad o cultura a la que se pertenezca. El resto de los gestos sí son aprendidos [9].

A continuación detallaremos un poco en que consiste cada emoción básica:

Alegría: Contracción del músculo cigomático (que va del pómulo al labio superior) y del orbicular que rodea al ojo.

Tristeza: Muy duradera. Caen los párpados superiores y se angulan hacia arriba las cejas. Además, el entrecejo se arruga y los labios se estiran horizontalmente.

Ira: La emoción más peligrosa para los demás, puede generar violencia. Mirada fija, ojos feroces, cejas juntas y hacia abajo.

Sorpresa: Es la emoción más breve y puede fundirse con otras. Los párpados superiores suben, pero los inferiores no están tensos. La mandíbula suele caer.

Miedo: Párpados superiores elevados al máximo e inferiores tensos, las cejas levantadas se acercan y los labios se alargan hacia atrás.

Esta aplicación es capaz de detectar tres de las emociones descritas, Alegría (Figura 4.3), Ira (Figura 4.4) y Sorpresa (Figura 4.5), además de la cara neutra (Figura 4.2). Eso se debe a las limitaciones que tiene el algoritmo de detección facial que se

está utilizando, ya que este no permite la torsión de las cejas ni la captura de las arrugas que están en los alrededores de la boca y nariz.



Figura 4.2: Cara Neutra



Figura 4.3: Cara Alegre



Figura 4.4: Cara enojada



Figura 4.5: Cara sorprendida

4.3. Técnicas de Detección Facial

4.3.1. Técnica CLMtracking

Se basa en los algoritmos descritos por Jason Saragih y Simon Lucey, denominado “Face Alignment through Subspace Constrained Mean-Shifts” [14] o “La alineación de rostros a través de medios restringidos en el sub-espacio”. El objetivo es adaptar un modelo facial a una cara en una imagen o video a partir de una inicialización aproximada.

El algoritmo se ajusta al modelo facial mediante el uso de 70 puntos donde cada uno de ellos se ajusta a un clasificador. Dada una posición aproximada inicial, los clasificadores buscan una región pequeña (de ahí el nombre “local”) alrededor de cada punto para un mejor ajuste, y el modelo se mueve gradualmente en la dirección con el mejor ajuste, convergiendo gradualmente al ajuste óptimo.

Un rostro no es complicado de modelar, ya que además de las expresiones, posturas y facciones, las caras no varían mucho de persona en persona. En esta técnica los puntos están marcados ya que de esa forma es más fácil de trabajar y ocupa una anotación existente en la base de datos de MUCT ¹[24].

¹La base de datos MUCT consta de 3755 caras con 76 puntos de referencia manuales y se creó para proporcionar más diversidad de iluminación, edad y etnicidad a las bases de datos para los rostros 2D que actualmente están disponibles.

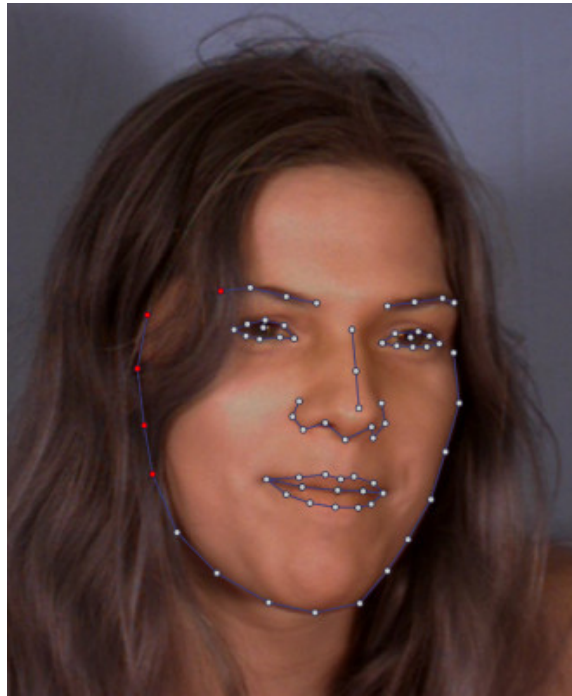


Figura 4.6: Un rostro anotado de la base de datos MUCT.

Para la construcción del modelo se ocupa “Principal Component Analysis” (PCA) que significa análisis de componentes principales. Una vez que se calculan los puntos medios de las anotaciones, se usa el PCA para extraer las variaciones de las caras como combinaciones lineales de vectores o componentes, donde se extraen por orden de importancia. Dado que los primeros componentes logran cubrir la mayor parte de la variación en las posturas faciales, se puede descartar el resto sin ninguna pérdida en la precisión del modelo, logrando extraer primero las variaciones básicas de la postura, seguido de apertura y cierre de la boca, sonrisa, etc.

PCA almacena los valores propios de cada componente, estos nos dicen la desviación estándar de los pesos de cada componente de acuerdo a las poses de los rostros con los datos que fueron anotados, lo cual es útil para regularizar los pesos cuando se requiera optimizar. Es importante destacar la necesidad de que las caras para entrenar el modelo sean varias poses y expresiones diferentes para que el modelo pueda recrear la mayor cantidad de poses posibles.

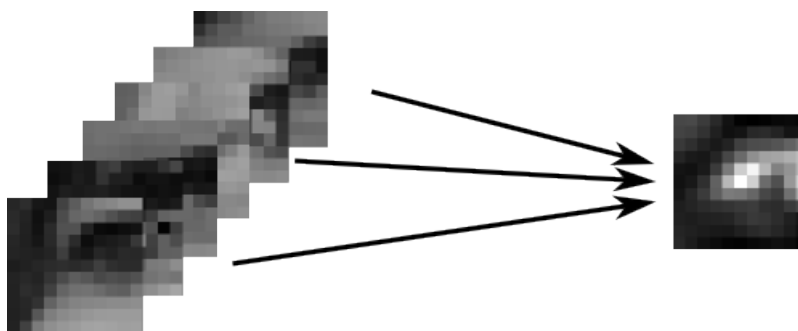


Figura 4.7: Entrenando un clasificador con parches de las caras anotadas

Para entrenar los clasificadores (los 70 puntos mencionados anteriormente), se recorta un parche X por X centrado en la posición marcada por el punto seleccionado y en cada una de las imágenes faciales que se anotaron para dicho punto, estas imágenes son las que entrenarán al clasificador, como se muestra en la Figura 4.7.

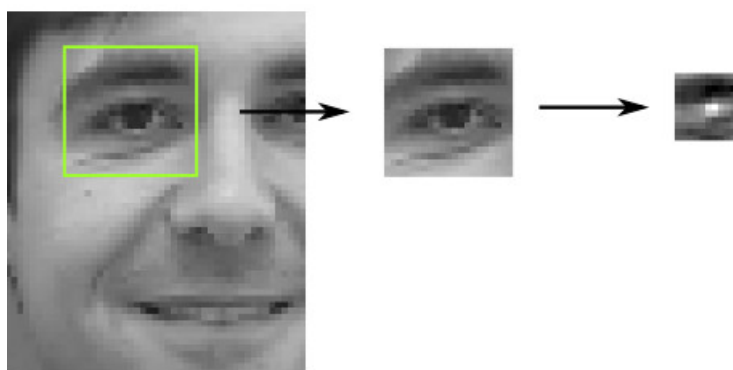


Figura 4.8: El parche recortado y la “respuesta” para el ojo izquierdo

El clasificador utilizado es de regresión logística con un kernel SVM² (Support Vector Machines o Máquinas de soporte vectorial) y también un filtro MOSSE³. Al usar estos clasificadores para ajustar el modelo se recorta una ventana y se busca

²SVMs son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T.

³Para más información revisar la tesis “Tutorial on Minimum Output Sum of Squared Error Filter”, escrita por Rumpal Kaur Sidhu, Department of Computer Science, Colorado State University.

alrededor de cada una de las posiciones aproximadas iniciales, se aplica el clasificador respectivo a una cuadrícula de Y en píxeles dentro de la ventana de búsqueda. Así se obtienen los resultados “respuesta” de $Y \times Y$ que mapean la probabilidad de que cada uno de estos píxeles sean el punto de característica “alineado”, como se muestra en la Figura 4.8.

Para que el modelo se ajuste de la mejor manera posible se calcula la forma en la que el modelo debería moverse hacia la región con mayor probabilidad. Esto se hace gracias al “cambio medio” (técnica de análisis del espacio característico no paramétrico para localizar los máximos de una función de densidad). Luego, se regulariza este movimiento al restringir las “nuevas posiciones” al espacio de coordenadas abarcado por el modelo facial. De esta forma, se asegura que los puntos del modelo no se muevan de forma inconsistente con el modelo en general. Este proceso se realiza de forma iterativa, lo que significa que el modelo facial converge gradualmente hacia el ajuste óptimo que se trata de un caso de maximización de expectativas, donde encontrar el mejor movimiento según las respuestas es el paso esperado y la regularización para modelar es el paso de maximización. Este funcionamiento se puede ver paso a paso con las Figuras 4.9, 4.10, 4.11, 4.12 y 4.13.

Es importante que se inicie el modelo no muy lejos de la ventana que busca el rostro, ya que estas ventanas son un poco pequeñas y el detector facial debe encontrar el cuadro delimitador aproximado de la cara para luego encontrar la posición de los ojos y nariz a través de un filtro de correlación. Finalmente se usa el análisis Procrustes⁴ para ajustar el modelo facial a las posiciones de los elementos encontrados.

⁴Análisis de Procrustes a la serie de métodos estadísticos que aplican la teoría de grupos al análisis de conjuntos de datos, para poder compararlos y realizar inferencias de dichas comparaciones.



Figura 4.9: Detecta la cara.

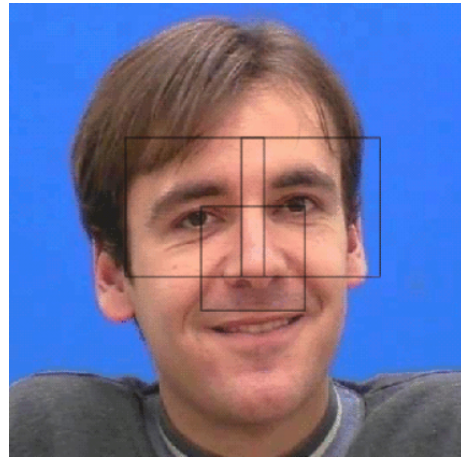


Figura 4.10: Cuadro delimitador de los ojos y nariz.



Figura 4.11: Posición de los ojos y nariz.

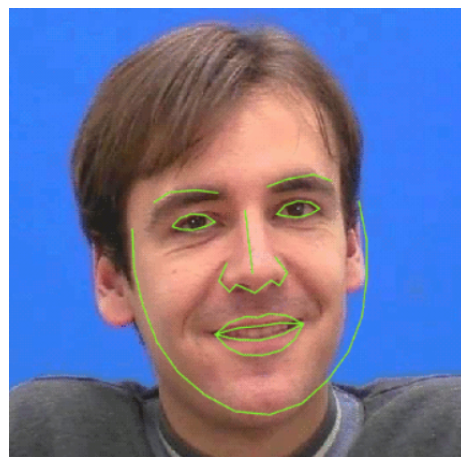


Figura 4.12: Coloca el modelo facial.

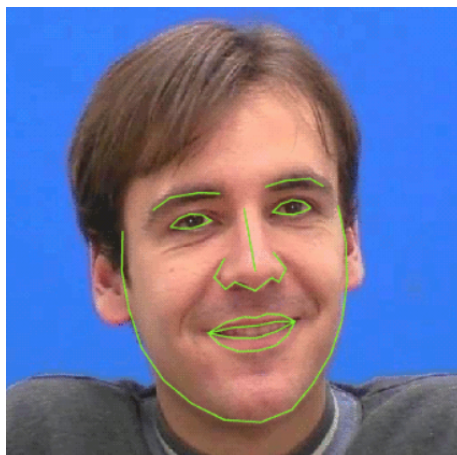


Figura 4.13: Ajuste con procrustes.

Al hacer el ajuste de cara con procrustes se tiene que verificar que el modelo no se haya alejado demasiado de la posición “verdadera” de la cara. Para ello se verifica periódicamente que la región aproximada que cubre el modelo se parece a la cara y se usan los mismos clasificadores que en los parches, con la regresión logística y con los que se entrenó la cara. La detección de rostros se reinicia cuando el modelo de la cara no se encuentra.

La implementación del algoritmo en JavaScript es un poco lenta. El cuello de botella se forma en los clasificadores que se llaman varias veces para cada punto del modelo en cada iteración. La implementación es una operación $O(m^2 \times n^2)$, donde el tamaño de la ventana de búsqueda es “ n ” y el tamaño de los parches clasificadores “ m ”. Usando la convolución por FFT (Fast Fourier Transform)⁵ puede quedar en $O(n \log(n))$, pero esto es aún más lento de lo que se desea. Acá toman importancia los núcleos SVM (Support Vector Machine), ya que tienen un rápido cómputo a través de la GPU, que se puede hacer a través de WebGL, y está disponible en la mayoría de los navegadores actuales.

Para poder hacer los cálculos eficientemente se ocupa una biblioteca llamada

⁵Transformada rápida de Fourier es de gran importancia en una amplia variedad de aplicaciones, desde tratamiento digital de señales y filtrado digital en general a la resolución de ecuaciones en derivadas parciales o los algoritmos de multiplicación rápida de grandes enteros

“numeric.js”, es la más rápida y completa para el cálculo de matrices y mitigar el otro cuello de botella que se forma al hacer estos cálculos con matrices.

4.3.2. Técnica de Viola-Jones con CLM

Dado que estamos interesados en diseñar un programa de computadora que pueda hacer detección y alineación de rostros como lo hace el ser humano, podemos diseñar nuestro programa en dos pasos. El primer paso es una exploración rápida con el detector de cara Viola-Jones. Este detector busca en una imagen y coloca rectángulos donde se encuentran las caras. El segundo paso es mirar más de cerca el rectángulo de la imagen generado por Viola-Jones, y determinar el límite de la nariz, la boca, etc. Esto es más difícil, y aquí es donde CLM (Constrained Local Model o Modelo local restringido) entra en acción.

Para que CLM pueda tener un uso más eficiente, se hace uso de Viola-Jones. Cada clasificador de la cascada se entrena con el algoritmo de boosting AdaBoost, donde los primeros clasificadores son muy sencillos y permiten rechazar una gran cantidad de no-caras mientras que aceptan un porcentaje muy alto de caras. En otras palabras hace un barrido en la imagen que se está analizando, determinando las zonas en las cuales no es probable encontrar algún rostro. Además, requiere de la biblioteca jsfeat.js (para la detección inicial de rostros) y numeric.js (para matrices matemáticas), pero estos se incluyen en la biblioteca minified.

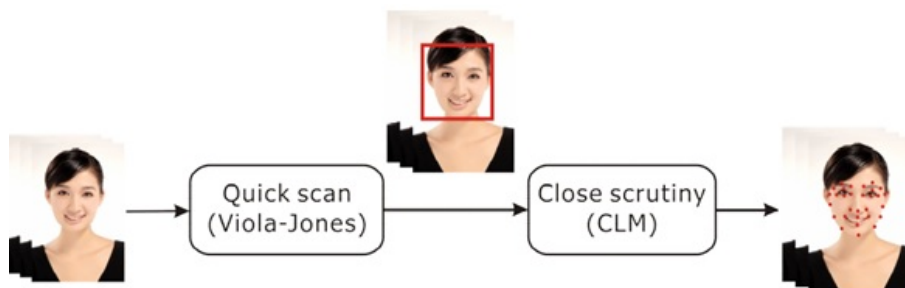


Figura 4.14: Arquitectura del sistema.

CLM tiene el trabajo de detectar los elementos específicos de la cara como lo son

los ojos, la boca y el contorno de la rostro. A estos nos referiremos como puntos de característica. Como se esta ocupando Viola-Jones ya tenemos un cuadrado de lo que podría ser un rostro. En la Figura 4.14 se muestra una imagen que se le aplica Viola-Jones para luego ocupar CLM.

El algoritmo de ajuste se basa en un artículo de Jason Saragih y Simon Lucey [14]. Los modelos son entrenados en datos anotados de la base de datos MUCT más algunas imágenes auto-anotadas.

Uso básico

Para poder usar el algoritmo Clmtrackr es necesario conocer las funciones básicas.

Primero esta la inicialización:

```
var ctracker = new clm.tracker();  
ctracker.init(pModel);
```

Luego el inicio de seguimiento:

```
ctracker.start(videoElement);
```

Obteniendo los puntos del modelo actualmente instalado:

```
var positions = ctracker.getCurrentPosition();
```

Dibujo del modelo actualmente instalado en un lienzo dado:

```
var drawCanvas = document.getElementById('somecanvas');  
ctracker.draw(drawCanvas);
```

Algunas funciones y breve descripción

Estas son las funciones que el objeto clmtracker expone:

init (model): inicializa clmtrackr con un modelo facial.
o model: objeto a modelar

start (element, box): inicia el rastreador. Devuelve falso si el rastreador no se ha inicializado con un modelo.

o element: un canvas o un video

o box: (opcional) el cuadro delimitador de donde está la cara, como un array [x, y, width, height] donde x e y se refieren a las coordenadas de la esquina superior izquierda del cuadro delimitador. Si no se da ninguna caja delimitadora, `clmtrackr` trata de detectar la posición de la cara misma.

stop(): detiene el rastreador en ejecución.

track(element, box): Devuelve las posiciones actuales del modelo ajustado como una matriz de posiciones $[[x_0, y_0], \dots, [x_n, y_n]]$ si la iteración de seguimiento tiene éxito. Devuelve falso si no se encuentra ninguna cara en el elemento dado o si se ha perdido el seguimiento.

o element: un canvas o un video.

o box: (opcional) el cuadro delimitador de donde está la cara, como un array [x, y, width, height] donde x e y se refieren a las coordenadas de la esquina superior izquierda del cuadro delimitador. Si no se da una caja delimitadora, utiliza la última posición conocida, o trata de detectar la posición de la cara.

reset(): restablece el seguimiento. Esto reinicializa la detección y el ajuste inicial.

draw(canvas, pv, path): dibuja el modelo facial que está actualmente equipado.

o canvas: el elemento del canvas para dibujar el modelo.

o pv: (opcional) los parámetros del modelo como una matriz. (El valor predeterminado es usar los valores de los parámetros actuales).

o path: (opcional) tipo de ruta a dibujar, ya sea “normal” o “vértices” (El valor predeterminado es: normal).

getScore(): Obtiene el puntaje actual de la adaptación del modelo. La puntuación se basa en un clasificador SVM que detecta qué tan precisa está la imagen bajo el

modelo que se asemeja a la cara. Los valores devueltos van desde 0 (sin ajuste) a 1 (ajuste perfecto). El umbral predeterminado para asumir que hemos perdido la pista de la cara es cualquier valor por debajo de 0.3.

getCurrentPosition(): Obtiene las posiciones actuales del modelo facial ajustado. Devuelve las posiciones como una matriz $[[x_0, y_0], \dots, [x_n, y_n]]$.

getCurrentParameters(): Obtiene los parámetros actuales para el modelo facial ajustado. Devuelve los parámetros del modelo como una matriz $[p_0, p_1, \dots, p_n]$.

Resumen del capítulo

El canal más importante de comunicación no verbal es la cara, ya que es la parte más expresiva del cuerpo y por ende, en ella se encuentran los matices más complejos y variados.

Según el experto Paul Ekman existen emociones básicas que todo ser humano puede interpretar, entre ellas está la alegría, tristeza, ira, sorpresa y miedo, pero en este proyecto sólo se trabajará con las tres primeras.

El reconocimiento facial realiza un análisis de las características del rostro de un sujeto, donde el objetivo es identificar las caras que se encuentran en una imagen o video.

Para poder detectar los rostros es necesario un entrenamiento previo.

También se describió el algoritmo de Viola-Jones y CLMtracking, técnicas utilizadas para la detección facial.

5. Diseño y Solución del Problema

Este proyecto consta de un trabajo en conjunto entre dos algoritmos, el primero de ellos es CLMtrack que nos proporciona todas la funcionalidades para poder hacer la detección de rostros. El otro es Ginger, el modelo que se trabajó. En el cual están todas la funcionalidades para lograr la deformación y lograr el efecto de imitar las emociones leídas por la detección.

La mayor problemática de este proyecto fue lograr que estos dos algoritmos trabajen juntos ya que cada algoritmo funciona por sí solo y cumple con el objetivo de su desarrollo. Para lograr la interacción entre ambos se tuvo que rearmar parte del código de cada uno y así lograr amalgamarlos logrando crear una sola aplicación.

5.1. Proceso de unión de los algoritmos

Existen varios algoritmos para realizar la detección de caras y el que se ocupa en este proyecto es **CLMtrackr**. Clmtrackr es una biblioteca de javascript para ajustar modelos faciales a caras en videos o imágenes. En la actualidad es una implementación de modelos locales restringidos, como se describe en el artículo de Jason M. Saragih [14].

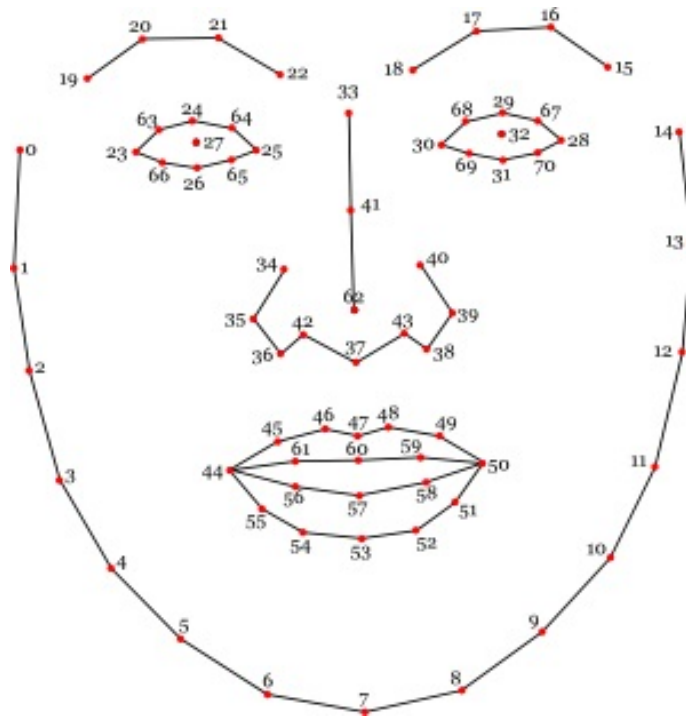


Figura 5.1: Máscara con los puntos de warp.

CImtrackr rastrea una cara y emite las posiciones de coordenadas del modelo de cara como una matriz, siguiendo la numeración del modelo como se muestra en la Figura 5.1.

El modelo Ginger cuenta con una serie de selectores donde cada uno sirve para mover una parte de la cara en específico. Como se menciona en la Sección 3.4, estos se modificaron para que pudieran funcionar independientemente, ya que en el modelo inicial sólo permitía el movimiento de un elemento de la cara a la vez.

Cada selector tiene su identificador, es de tipo rango y está oculto, los valores van del 0 al 1 como se muestran a continuación.

Abrir y cerrar los ojos

```
<input id="range_eyes" class="range" type="hidden" min="0" max="1"
step="0.01" value="0" >
```

Abrir o cerrar la mandíbula

```
<input id="range_jawrange" class="range" type="hidden" min="0" max="1"
step="0.01" value="0" >
```

Mostrar expresiones de felicidad

```
<input id="range_expression" class="range" type="hidden" min="0" max="1"
step="0.01" value="0" >
```

Inicialización de Clmtrackr

El código siguiente inicia el CLMtrackr con el modelo que incluimos, e inicia el rastreador que se ejecuta en un elemento de video, además de un canvas que luego sirve para colocar la máscara que se muestra en la Figura 5.1.

```
<video id="videoel" width="400" height="300" preload="auto" loop></video>
<canvas id="overlay" width="400" height="300"></canvas>
<script type="text/javascript">
  var vid = document.getElementById('videoel');
  var overlay = document.getElementById('overlay');
  var ctrack = new clm.tracker({useWebGL : true});
  ctracker.init(pModel);
  ctracker.start(vid);
</script>
```

Comenzando la detección

Para comenzar con la detección de rostros se debe hacer click en el botón “detectar rostro”(ver Figura 5.2), donde inicia las funciones de CLMtrack y la máscara de la Figura 5.1 se ajusta al rostro proyectado en la cámara.

Una vez que se inicia la detección de rostro se habilita el botón “Comenzar la Animación”. La función se llama ready4Morph() y habilita una bandera que nos

indica si las posiciones fueron capturadas para luego ser ocupadas más adelante. La variable “p” va tomando los valores de los elementos que se van a mover. Estos números son las posiciones de los warp-points de la Figura 5.1. Esto es importante ya que más adelante se van ocupar para calcular el porcentaje a dónde se van a mover los rasgos faciales del modelo (ver el Algoritmo 1 como referencia).

Algorithm 1 Comenzar la Animación READY4MORPH(*ctrack*)

Entrada:

ctrack, Objeto que tiene las funciones del algoritmo CMLtracking

Salida:

void

Método:

- 1: Bandera *MORPHREADY* se hace *true*
 - 2: Asigno las posiciones de los warp-point al arreglo *positions*
 - 3: *p* toma el valor de 8
 - 4: *GlobalinitExpresion* recibe el valor *p* de *positions*
 - 5: *p* toma el valor de 27
 - 6: *GlobalinitEyeL* recibe el valor *p* de *positions*
 - 7: *p* toma el valor de 32
 - 8: *GlobalinitEyeR* recibe el valor *p* de *positions*
 - 9: *p* toma el valor de 60
 - 10: *GlobalinitLipUp* recibe el valor *p* de *positions*
 - 11: *p* toma el valor de 57
 - 12: *GlobalinitLipDown* recibe el valor *p* de *positions*
-

Funciones importantes

Una vez iniciada la función *drawloop()* se ejecutan las funciones que calculan los porcentajes (ver Algoritmo 2) en los que se van a mover los rasgos de la cara del modelo (sólo si se encontró la posición del rostro por medio de la detección facial). Los rasgos serían los ojos, la boca y la expresión en sí. Las funciones que hacen esto posible son *calculatePercentageOpenMouth()*, *calculatePercentageExpresion()*

y `calculatePercentageEye()` respectivamente.

Algorithm 2 Inicia el bucle para dibujar la cara `DRAWLOOP()`

Entrada:

overlay, Máscara con los warp-point

ctrack, Objeto que tiene las funciones del algoritmo

Salida:

void

Método:

- 1: **if** *ctrack* obtiene posiciones **then**
 - 2: *ctrack* dibuja *overlay*
 - 3: Asigno las posiciones de *ctrack* a *positions*
 - 4: **if** *positions* existe **then**
 - 5: calculo el porcentaje de abrir la boca.
 - 6: calculo el porcentaje de la expresión.
 - 7: calculo el porcentaje de los ojos.
-

Como se dijo anteriormente primero debemos saber las posiciones de los warp-points que determina la detección facial y para obtener las posiciones de las características faciales, como se hace en el Algoritmo 2 en el paso 3, usamos `getCurrentPosition()`. Esta función esta pre-definida en el javascript de `CLMTraking`, el archivo se llama `clm.js`

La función `calculatePercentageOpenMouth()` es la encargada de hacer los cálculos para que la boca del modelo se abra o se cierre (ver Algoritmo 3). Primero se verifica que la bandera `MORPHREADY` sea verdadera; esta nos indica que las posiciones iniciales fueron capturadas. Luego de inicializar las variables se obtienen las posiciones de los puntos 57 y 60 que corresponden a los labios superior e inferior (estos valores se pueden ver en la máscara con los puntos de warp de la Figura 5.1). El cálculo en sí es simple ya que se toma el valor que corresponde al labio inferior (más grande, p:57) y se resta con el valor del labio superior (valor más pequeño, p:60). El máximo que se puede abrir la boca es de 20 unidades, este valor se calculó según

la posición donde ésta la ventada del video y la distancia que va a estar la persona frente a la cámara. Por último, el valor se transforma a porcentaje y se asigna a una variable que luego se pasará al modelo.

Algorithm 3 Calcula el porcentaje de abrir la boca CALCULATEPERCENTAGE-MOUTH(*morphReady*)

Entrada:

morphReady, indica *true* si el morph esta listo

Salida:

void

Método:

```

1: if morphReady then
2:   positions  $\leftarrow$  getCurrentPosition()      ▷ Posición de todos los warp-point
3:   p  $\leftarrow$  60
4:   boca1  $\leftarrow$  positions(p)                ▷ Obtengo el valor en el índice p
5:   p  $\leftarrow$  57
6:   boca2  $\leftarrow$  positions(p)                ▷ Obtengo el valor en el índice p
7:   first  $\leftarrow$  boca2 - boca1
8:   max  $\leftarrow$  20
9:   if first < max then      ▷ Condición para saber como calcular el porcentaje
10:    percentageMouth  $\leftarrow$  first/max
11:  else
12:    first  $\leftarrow$  max
13:    percentageMouth  $\leftarrow$  first/max
14:  PERCENTMOUTHGLOBAL  $\leftarrow$  percentageMouth

```

La función *calculatePercentageEye()* es la encargada de determinar cuando se abrirán o cerrarán los ojos del modelo (ver Algoritmo 4). Primero se verifica que la bandera MORPHREADY sea verdadera; esta nos indica que las posiciones iniciales fueron capturadas. Luego se inicializan las variables y se obtienen las posiciones actuales, el trasfondo detrás de este código es calcular la distancia entre el punto que

se encuentra en la parte superior de la nariz (p:33) y la posición de los ojos (p:32), con estos dos puntos se calcula la distancia entre ellos, el valor se obtiene cuando se mueve la posición de los ojos. Ese dato se trasforma a un porcentaje que luego recibe el modelo.

Algorithm 4 Calcula el porcentaje de abrir los ojos CALCULATEPERCENTAGEEYE(*morphReady*)

Entrada:

morphReady, indica *true* si el morph esta listo

Salida:

void

Método:

```

1: if morphReady then
2:   positions ← getCurrentPosition()    ▷ Posiciones de todos los warp-point
3:   p1 ← 32
4:   p2 ← 33
5:   x1, y1 ← getPositionPoint(p1)    ▷ Valores X e Y del índice p1
6:   x2, y2 ← getPositionPoint(p2)    ▷ Valores X e Y del índice p2
7:   x ←  $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$     ▷ Distancia entre dos puntos
8:   percentageEye ← toPercentage(x)    ▷ Conversión a porcentaje
9:   PERCENTEYESGLOBAL ← percentageEye

```

La función que nos cambia la expresión del modelo ya sea para que esté feliz o enojado es *calculatePercentageExpresion()*, ver Algoritmo 5. Al igual que en las funciones anteriores, existe la bandera que nos indica si la posición inicial de los warp-points fueron capturados. Esta función es un poco más compleja que el resto, ya que el movimiento de las cejas puede ser hacia arriba o hacia abajo. El punto que se toma como referencia es el 22 que corresponde a la ceja derecha y como ambas cejas se mueven en conjunto basta con tomar un solo punto. Luego se obtiene el valor del punto inicial y se guardan los valores máximos de qué tan arriba o abajo se puede mover. Luego de hacer un cálculo entre la cercanía del sujeto que va a estar en frente a la cámara y el tamaño que tiene la máscara con los warp-point, se determinó

que el valor óptimo sería de 20 unidades. Es por ello que a la posición inicial se le suma 10 para obtener la posición máxima hacia abajo y se le restan 10 para obtener la posición máxima que se mueven las cejas hacia arriba. Luego se tiene que validar que la posición actual de las cejas no exceda los valores predefinidos anteriormente.

Para explicarlo mejor se ocupa un ejemplo. Si nuestra posición inicial está en 120 (estos números son entregados por el modelo de detección facial), el máximo superior es de 130 y el máximo inferior es de 110, por ende el rango por el que se puede hacer el movimiento de las cejas es de 20 (el rango que se calculó anteriormente). Ahora se tiene que transformar el valor de la posición actual a un porcentaje. Para lograr esto se resta el máximo inferior a la posición actual, es decir $120 - 110 = 10$ con el fin de que este valor se mueva entre 0 a 20 (rango definido anteriormente) y a este valor le hacemos una regla de tres para calcular el porcentaje que luego se le pasa al modelo Ginger.

Algorithm 5 Calcula el porcentaje la expresión del modelo CALCULATEPERCENTAGEEXPRESION(morphReady)

Entrada:

morphReady, indica *true* si el morph esta listo

Salida:

void

Método:

```

1: if morphReady then
2:   positions ← getCurrentPosition()    ▷ Posiciones de todos los warp-point
3:   p ← 22
4:   initExpresion ← GlobalinitExpresion    ▷ Posición inicial
5:   endBotExpresion ← initExpresion + holgura    ▷ Limite máximo inferior
6:   endTopExpresion ← initExpresion – holgura    ▷ Limite máximo superior
7:   moving ← positions(p)    ▷ Posición actual
8:   if endTopExpresion > moving then
9:     moving ← endTopExpresion    ▷ Comprueba que no exceda los límites
10:  else if endBotExpresion < moving then
11:    moving ← endBotExpresion
12:    x ← moving – endTopExpresion
13:    h2 ← holgura * 2
14:    percentageExpresion ← 1 – x/h2    ▷ Cálculo del porcentaje
15:    if percentageExpresion > 1 then
16:      percentageExpresion ← 1
17:    else if percentageExpresion < 0 then
18:      percentageExpresion ← 0
19:    PERCENTEXPRESIONGLOBAL ← percentageExpresion

```

Por último para poder pasar los valores obtenidos por las funciones anteriores se modificó la función *animate*(*o*). Para ello antes de hacer la actualización del modelo Ginger por medio de *updateMorph*(*o*) tenemos que decirle al *select*(*o*) qué elemento del modelo queremos modificar. Luego se llaman las funciones correspondientes para

realizar el refresco de la imagen en el navegador. Este procedimiento se muestra en el Algoritmo 6.

Algorithm 6 Generando el morph con ANIMETE()

Entrada: ▷ Esta función no tiene parámetros de entrada

Salida:

void

Método:

```

1: select("jawrange") ▷ Selecciona la mandíbula
2: if PERCENTMOUTHGLOBAL  $\neq$  0 then
3:   updateMorph(PERCENTMOUTHGLOBAL) ▷ Actualiza el modelo
4:   morph()
5: select("expression") ▷ Selecciona la expresión
6: if PERCENTEXPRESIONGLOBAL  $\neq$  0 then
7:   updateMorph(PERCENTEXPRESIONGLOBAL) ▷ Actualiza el modelo
8:   morph()
9: select("eyes") ▷ Selecciona los ojos
10: if PERCENTEYESGLOBAL  $\neq$  0 then
11:   updateMorph(PERCENTEYESGLOBAL) ▷ Actualiza el modelo
12:   morph()
13: requestAnimationFrame(animate) ▷ Refresca la escena
14: render(scene, camera)

```

El proyecto finalizado se puede apreciar en la Figura 5.2, para ver más imágenes revisar Anexo A.1.

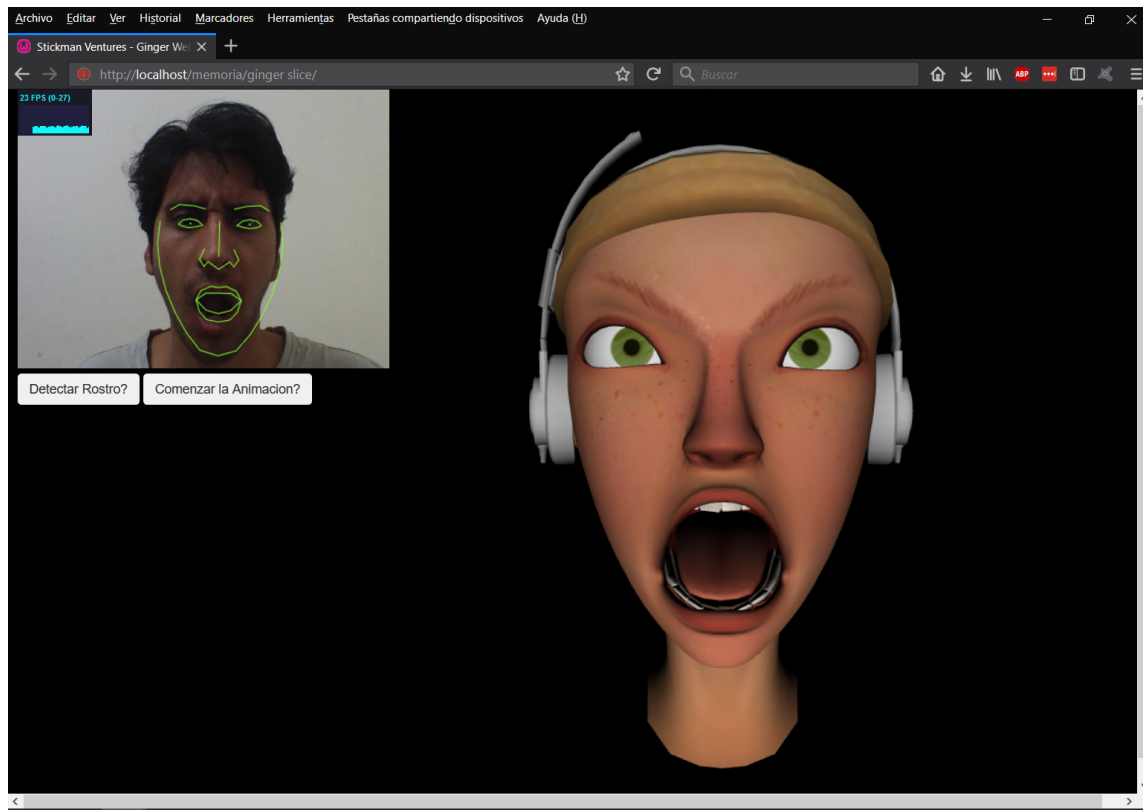


Figura 5.2: Resultado final.

Resumen del Capítulo

Como el problema radica en cómo realizar la interacción de dos algoritmos distintos se optó por crear un sólo archivo, integrado las funciones más importantes de ambos.

Para hacer funcionar esta aplicación se realizaron cambios a las funciones ya creadas. Las funciones que se mencionaron en éste capítulo fueron: `ready4Morph()` que prepara los valores iniciales, `drawLoop()` para dibujar la máscara con los warp-point, las funciones que calculan los porcentajes de la expresión, ojos y boca, y por último la función `animate()`, encargada de realizar la actualización del modelo en la pantalla del navegador.

6. Evaluación de Resultados Obtenidos

En este capítulo se detalla la evaluación de la aplicación. Esta se llevo a cabo por medio de una encuesta, y para la realización de ella se siguió el procedimiento que decreta el Comité de Bioética de la Dirección de Investigación de la Universidad de Talca.

6.1. Descripción del Método de Evaluación

Para la evaluación se realizó una encuesta que consta de 25 imágenes con diferentes expresiones de Alegría, Enojo y Sorpresa. Estas imágenes se obtienen de un modelo tridimensional en específico y de rostros de personas. Las imágenes de las personas reales serán de rostros aleatorios y de diferente sexo.

El número total de voluntarios requeridos para esta encuesta es de 15, en donde los participantes tienen que, para cada imagen, elegir entre las 3 diferentes emociones mencionadas anteriormente.

El rango de edad se fijó entre los 18 y 28 años de edad, ya que el uso de la aplicación está orientada a estudiantes universitarios. El objetivo de este estudio es determinar qué tan fácil/difícil se le hace a los encuestados identificar la emoción que está expresando un rostro de una persona real y el rostro en el modelo tridimensional.

La aplicación de esta encuesta fue muy rigurosa, a los encuestados se les dio la misma presentación e instrucciones para completar la encuesta, se realizó en un solo

computador con una intensidad de brillo y contraste fijas. Las fotos que se muestran en la encuesta son todas iguales en tamaño, con el fondo blanco y enfocadas en el rostro, las opciones son iguales para cada imagen y en el mismo orden.

La encuesta es única, esto hace referencia al orden de las fotografías mostradas. Pero cabe destacar que las fotografías están mezcladas, es decir, las fotos del modelo tridimensional y las fotos de los rostros reales fueron puestas al azar, al igual que las emociones que representan cada una.

El propósito de tener diversas fotografías es demostrar que la detección facial y el modelo se adaptan a cualquier gesto que realice un rostro.

El orden de los pasos para la realización fue estandarizado. Primero se saludó e indicó en qué consiste la encuesta para luego preguntar si tienen la disponibilidad de contestarla. Si aceptan se lee el documento de “Consentimiento Informado” donde se explica todo más detallado y se le hace saber al encuestado que la decisión de participar es completamente suya y que puede aceptar o rechazar la investigación, e incluso arrepentirse de su primera decisión sin ningún problema o represalia. Dependiendo de su respuesta (aceptación, rechazo o revocación) se le pasa un documento para que firme.

Las instrucciones dadas a los encuestados se puede encontrar en el Anexo B.1, junto a los documentos mencionados anteriormente.

Se toma el porcentaje de cada imagen agrupándola por emociones. Por ejemplo, si existen 6 imágenes con la emoción de alegría y responden las 6 correctas el porcentaje de éxito de esa pregunta sería del 100%. Si el porcentaje supera el 70% diremos de esa pregunta tuvo éxito en ser reconocida. Además supondremos que la probabilidad de la detección de emociones que se hace al objeto tridimensional para esa emoción es más alta, pero eso lo averiguaremos cuando obtengamos todos los resultados.

6.2. Resultados de la Encuesta



Figura 6.1: Gráfico de los porcentajes máximos de cada pregunta.

El gráfico de la Figura 6.1 muestra los porcentajes máximos que obtuvo cada pregunta. Luego se analiza en más detalle cuál fue la opción seleccionada en cada pregunta. El objetivo de este gráfico es mostrar que el porcentaje varía entre 53,3% y 100%, esto quiere decir que los encuestados decidieron inclinarse por una respuesta en concreto, ya que cada pregunta supera el 50% de una elección y ahora queda determinar si esa respuesta es la esperada.

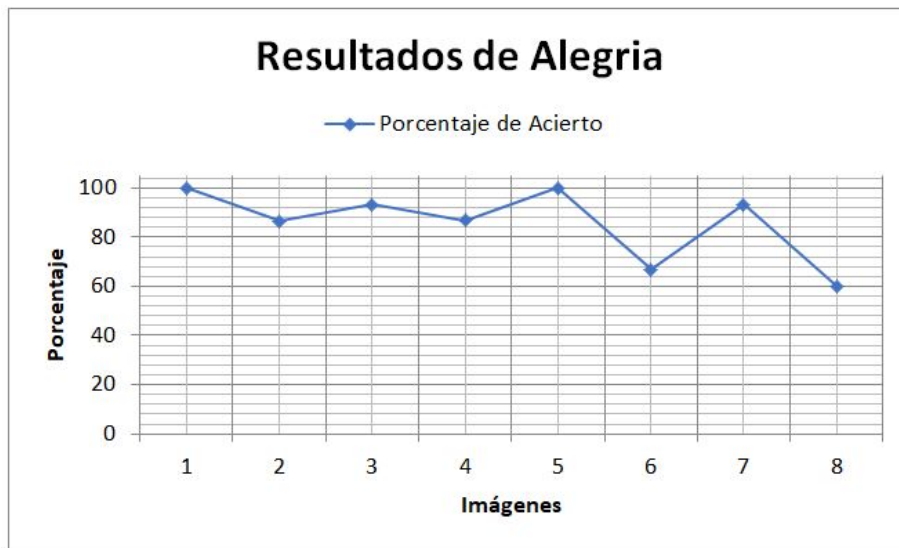


Figura 6.2: Porcentaje de Acierto para la emoción de Alegría.

El gráfico de la Figura 6.2 muestra que solo 2 imágenes tienen un porcentaje menor al 80%. Los rostros tridimensionales están en la 4ta, 6ta y 8va posición, donde las últimas 2 posiciones corresponde a un bajo porcentaje, con los datos que se muestran en el Cuadro 6.1 podemos saber cuál fue la segunda opción más votada y llegar a un mayor análisis.

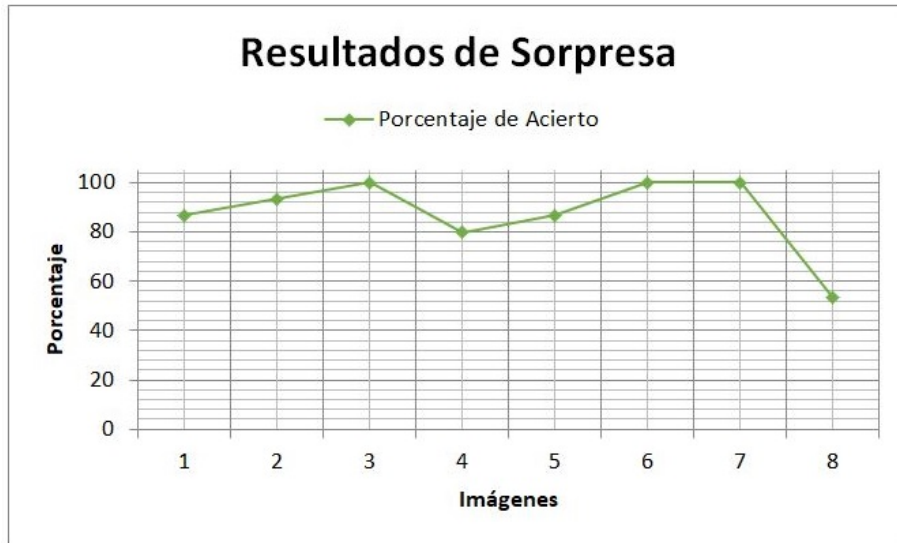


Figura 6.3: Porcentaje de Acierto para la emoción de Sorpresa.

En el gráfico de la Figura 6.3 las imágenes del modelo tridimensional están en la 1ra, 2da y 4ta posición, siendo la 4ta posición la más baja con 80% y el dato más bajo de todo el gráfico corresponde a un rostro real con un 53,3%, sin contar este último podemos decir que los valores fueron buenos.

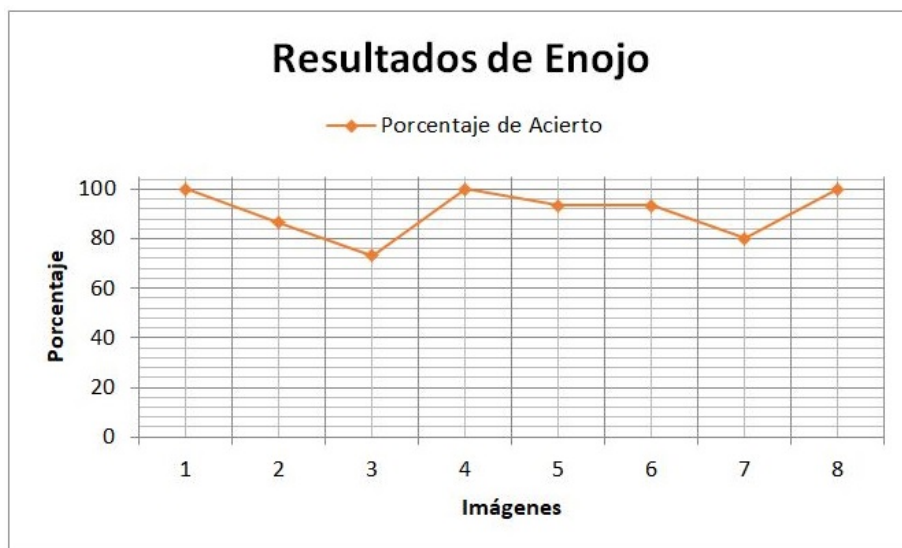


Figura 6.4: Porcentaje de Acierto para la emoción de Enojo.

En el gráfico de enojo (Figura 6.4) el porcentaje más bajo fue de un rostro real en la posición 3 con un 73,3% y las imágenes del modelo tridimensional están en las posiciones 2da, 6ta y 7ta con 86,6%, 93,3% y 80% respectivamente.

Cuadro 6.1: Porcentajes según respuesta.

Preguntas	Alegría	Enojo	Sorpresa	Más de Una	Ninguna de las Anteriores
1	100	0	0	0	0
2	0	0	86,7	13,3	0
3	0	100	0	0	0
4	86,6	0	6,7	6,7	0
5	0	0	93,3	6,7	0
6	0	0	100	0	0
7	0	86,6	0	6,7	6,7
8	0	73,3	0	20	6,7
9	93,3	0	0	6,7	0
10	86,7	0	0	13,3	0
11	100	0	0	0	0
12	0	100	0	0	0
13	6,7	0	80	6,6	6,7
14	0	0	86,7	13,3	0
15	66,7	0	0	26,6	6,7
16	13,3	0	66,7	20	0
17	0	93,3	0	6,7	0
18	0	93,3	6,7	0	0
19	0	0	100	0	0
20	0	80	0	13,3	6,7
21	0	100	0	0	0
22	0	0	100	0	0
23	93,3	0	0	0	6,7
24	60	0	0	13,3	26,7
25	26,7	0	53,3	20	0

El Cuadro 6.1 muestra las preguntas con todos los porcentajes obtenidos en cada

alternativa, el valor correcto con menor porcentaje fue de 53,3% , correspondiente a la última pregunta.

Cuadro 6.2: Resultados esperados con sus respectivos porcentajes.

Preguntas	Respuesta Esperada	Porcentaje de Acierto
1	Alegría	100
2	Alegría	86,7
3	Enojo	100
4	Alegría	86,6
5	Alegría	93,3
6	Alegría	100
7	Enojo	86,6
8	Enojo	73,3
9	Alegría	93,3
10	Alegría	86,7
11	Alegría	100
12	Enojo	100
13	Alegría	80
14	Alegría	86,7
15	Alegría	66,7
16	Alegría	66,7
17	Enojo	93,3
18	Enojo	93,3
19	Alegría	100
20	Enojo	80
21	Enojo	100
22	Alegría	100
23	Alegría	93,3
24	Alegría	60
25	Sorpresa	53,3

Resumen del Capítulo

El Cuadro 6.2 nos muestra las respuestas esperadas por cada pregunta y el porcentaje de encuestados que escogieron esa alternativa. Estos datos los podemos comparar con los resultados del Cuadro 6.1.

En este caso todos los datos del Cuadro 6.2 están sobre el 50 % y por ende podemos decir que más de la mitad de los encuestados encontraron la emoción que se deseaba.

Ejemplo hipotético: tomemos una pregunta donde la respuesta esperada fuera de Alegría y 37 % de acierto ¿Cómo sabemos si ese valor (que es el correcto) es el mayor? para ello se tiene que mirar el Cuadro 6.1, sus valores resultaron algo así: Alegría 37 %, Enojo 0 %, Sorpresa 6 %, Más de una 57 % y Ninguna de las anteriores 0 %. La respuesta correcta dijimos que fue Alegría, pero la más votada fue Más de una con un 57 %. Esto nos dice que el valor esperado no fue el más alto.

Según los resultados obtenidos el porcentaje promedio fue de un 87,2 %, todas las preguntas obtuvieron la respuesta esperada, el acierto de cada una supero el 50 %, donde la pregunta con menor tasa de acierto fue la 25 con un 53,3 % en sorpresa.

7. Conclusiones y Trabajo Futuro

7.1. Análisis de Resultados

Ahora que tenemos los datos en los cuadros los analizaremos detenidamente emoción por emoción, las imágenes que se ocuparon en la encuesta están en el Anexo B.1.

La pregunta 25 fue puesta con la intención de generar opiniones distintas, ya que en ella aparece un rostro que se puede interpretar de 2 formas, de alegría o sorpresa. La tendencia de los encuestados fue para sorpresa con un 53,3% (la emoción que se esperaba encontrar) y con un 26,7% para alegría, el 20% restante está en la opción “Más de una”. El objetivo se logró por que la respuesta predominante fue la esperada y sabíamos que podría existir una inclinación hacia otras opciones.

En un comienzo se planteo el supuesto: “si el porcentaje de éxito supera el 70% significa que la probabilidad de que la detección de emociones que se hará al objeto tridimensional será más alta”. Pero esto no resulto ser cierto, ya que como se puede apreciar en el gráfico de la emoción Alegría, los resultados obtenidos en los rostros de personas fueron elevados, superando el 80%, en cambio en el rostro tridimensional, sólo uno de ellos logró superar ese porcentaje. Con esto podemos sacar dos conclusiones, la primera es que el modelo tridimensional no es lo suficientemente expresivo para lograr una expresión más clara de la emoción alegría, la segunda es que no importa la cantidad de rostros de personas que se le muestren a los encuestados,

ya que eso no influirá en la toma de decisiones cuando les toque evaluar al modelo tridimensional. Es importante decir que los rostros fueron ordenados al azar con la intención de no entrenar o influir en la toma de decisiones del encuestado, vale decir, no se pusieron las fotos de alegría de los rostros y del modelo juntas.

Gracias a los gráficos se puede determinar que la emoción más fácil de reconocer (en ambos tipos de imágenes, las reales y el modelo) fue la sorpresa. Esto se puede ver claramente en la Figura 6.3, las respuestas superan el 80 % a excepción de la última que corresponde a la pregunta 25 mencionada anteriormente.

Para el gráfico de la Figura 6.4 se obtuvieron buenos resultados para el modelo tridimensional, ya que todas las respuestas superaron el 80 %. Con esto podemos concluir que el modelo es lo suficientemente expresivo en la emoción de enojo. Las imágenes de los rostros reales también tuvieron buenos resultados a excepción de la pregunta 3 que fue la más baja, sus porcentajes fueron 73,3 % para enojo, 20 % para más de una y el 6,7 % para ninguna de las anteriores. Con esto podemos sacar 2 posibles conclusiones, la primera y la más evidente, a los encuestados les resulta difícil catalogar esta fotografía y la segunda, la selección de la imagen puede que no haya sido la más acertada y por ello resultó ser un poco más compleja de evaluar. La imagen se puede ver en la pregunta 8 de la encuesta que esta en el Anexo B.2.

Como el porcentaje de los datos obtenidos fue de un 87,2 %, los resultados fueron buenos, la mayoría superó el 80 %, además todas las respuestas obtuvieron el mayor porcentaje en la opción correcta.

La emoción de sorpresa fue la que obtuvo los resultados más elevados, con esto podemos decir que entre las tres es la más fácil de reconocer, ya sea en los rostros de las personas como en el modelo.

Los modelos tridimensionales de la emoción alegría fueron los más difíciles de identificar, ya que 2 de 3 de ellos están cerca del 65 %. Caso contrario fue para la emoción de enojo, ya que el modelo obtuvo resultados igual o mayores al 80 %, lo que es muy bueno por que esto nos quiere decir que el modelo logra expresar de manera clara la emoción de enojo.

7.2. Algoritmos Ocupados

El algoritmo de CLMtrack que realiza la detección de rostros resultó de gran ayuda por que tenía varios métodos implementados, el código era fácil de leer y los comentarios dentro de él eran acertados, en el documento está todo bien especificado, y cuenta con lo necesario para que uno se haga una idea del funcionamiento.

En lo personal el algoritmo funcionó sin inconvenientes pero nos encontramos con una gran limitante, y esta fue que no se pudieron reproducir todos los gestos que buscábamos replicar. En un comienzo se quería hacer la detección de 5 emociones, las cuales eran Alegría, Sorpresa, Enojo, Tristeza y Miedo, pero estas últimas 2 no se pudieron realizar debido al algoritmo CLMtrack, ya que no reconoce los músculos de la cara que hacen posible visualizar dichas emociones. Para poder lograr esta detección se debe desarrollar e implementar código extra y esto escapa del alcance de este proyecto.

El modelo Ginger fue complejo de entender, la escritura de las funciones estaban en javascript, un lenguaje que este autor no conocía. Eso implicó comenzar a estudiar un lenguaje nuevo desde cero. Esto se traduce a dedicar más tiempo para entender y luego desarrollar nuestra aplicación. Otro punto en contra fue la poca información que estaba disponible, ya que en la página solo se menciona un funcionamiento general y la documentación del código era escasa o casi nula. Una vez se entendió el funcionamiento del código, se descubrió la versatilidad que podía llegar tener el modelo. En él ya existían varios arreglos y matrices con una infinidad de puntos con coordenadas X, Y y Z que permiten realizar diversos movimientos, pero no se pudo explotar el potencial del modelo debido a que la detección del rostro estaba limitada por el algoritmo CMLtrack, lo mencionado en el párrafo anterior.

Realizar la interacción entre los 2 algoritmos en un comienzo fue difícil debido al poco conocimiento de javascript que se poseía, además que dentro del modelo Ginger existen varias funciones y variables que nacen y mueren dentro del él y al momento de querer llamarlas en el código de CLMtrack simplemente no se podía. Se buscaron varios métodos para poder pasar los parámetros, ya sea por medio de funciones que devolvían el valor por medio de “return”, ocupar parte del código de

CLMtrack dentro del modelo Ginger, o viceversa, crear el modelo Ginger dentro del código de CLMtrack, algunas de ellas no tuvieron éxito y otras funcionaron en parte, pero luego de varios intentos se llegó a una solución. Es por ello que es importante realizar una buena investigación de ambos códigos antes de empezar a trabajar y leer bien cada documento para evitar problemas similares.

7.3. Trabajo a Futuro

Lo más importante como trabajo a futuro es buscar una forma de lograr que el algoritmo de detección facial capture todos los gestos necesarios para replicar las 5 emociones que se querían lograr en un comienzo, para ello se debe mejorar CMLtrack o buscar un algoritmo que cumpla con todo lo que se tiene y lo que falta.

Como se menciona en el Capítulo 2, existen empresas dedicadas a este tipo de aplicaciones y ocupan cámaras profesionales para realizar la detección facial (como se muestran en la Figura 2.3). Este proyecto ocupa la cámara web que viene de fábrica en el computador portátil, por ello un trabajo a futuro será implementar la opción de poder ocupar las cámaras de cabeza de la Figura 2.2 y buscar la forma de hacer la conexión con la aplicación web.

A lo largo de este proyecto se trabajó con el modelo Ginger, pero en el transcurso de la implementación surgió la idea de poder cargar más modelos, esto se trató de agregar pero para poder hacer otro modelo se tienen que identificar todos los índices de los puntos que recrean los gestos de Ginger. Estos están agrupados según los elementos de la cara (boca, nariz, ojos, etc.) y en cada uno de ellos hay una infinidad de puntos y reproducir todos estos puntos es una tarea que demandará mucho tiempo. Como el objetivo de este proyecto no es cargar más de un modelo, se optó por dejar este alcance como trabajo futuro.

Bibliografía

- [1] Ron Alcalay. Morphing out of identity politics: Black or white and terminator 2. <https://bad.eserver.org/issues/1995/19/alcalay>, 1995.
- [2] Andrea C G Mennucci, Alessandro Cellerino & Dario Valenzano. What is an average shape and what is an average texture. Technical report, Instituto Neurofisiología CNR Pisa, Julio 2004.
- [3] Elaine Bell. Morphing. Curso EDC385G Multimedia Authoring de la Univesidad de Texas.
- [4] Eric Bidelman. Proyecto: “HTTP 2.0 push on App Engine”, <https://github.com/GoogleChrome/http2push-gae>, Septiembre 2015.
- [5] Brendan Eich. <https://www.javascript.com/>, 1995.
- [6] Bruno Costa, Jonas Gomes, Lucia Darsa, Luiz Velho. *Warping and Morphing of Graphical Objects*. Institute of Pure and Applied Mathematics, Rio de Janeiro, Brazil, 1998.
- [7] BTS Bioengineering. <http://www.btsbioengineering.com>, Junio 1986.
- [8] Dynamixyz. Máquinas expresivas, <http://www.dynamixyz.com>, Abril 2010.
- [9] Paul Ekman. ¿Qué dice ese gesto? Descubre las emociones ocultas tras las expresiones faciales. 2004.

- [10] Michael J. Gourlay. *Xmorph y Gtkmorph*. 1994-2000.
- [11] Grupo Khronos. www.khronos.org/webgl, Marzo 2011.
- [12] V Hall. *Morphing in 2D and 3D*. *Dr. Dobb's Journal, Annual Graphics Issue*, pages 18–26 y 132. 1993.
- [13] Faceware Technologies INC. <http://facewaretech.com>, Enero 2012.
- [14] Jason M. Saragih, Simon Lucey & Jeffrey F. Cohn. Face alignment through subspace constrained mean-shifts. *Robotics Institute, Carnegie Mellon University, Pittsburgh, USA 15213*, 91, Enero 2011.
- [15] Dr. Dobb's Journal. V hall. morphing in 2d and 3d. *Annual Graphics Issue*, pages 18–26 y 132, 1993.
- [16] jQuery. <https://jquery.com>, Agosto 2006.
- [17] Joaquín Planells Lerma. Implementación del algoritmo de detección facial de violaciones. 2009.
- [18] Raquel P. Conde López. Mapeo facial de emociones sintéticas. 2005.
- [19] Edwar Samir Posada Murillo. *Reducción y ajuste de mallas triangulares*. PhD thesis, Universidad Nacional de Colombia, Sede Medellín, Facultad de Ciencias, Escuela de Matemáticas, 2013.
- [20] PolymerElements. Platinum service worker elements, <https://polymerelements.github.io/platinum-sw/components/platinum-sw>.
- [21] Ricardo Cabello. <http://threejs.org>, Abril 2010.
- [22] Roger Claypoole, Jim Lewis, Srikrishna Bhashyam & Kevin Kelly. Elec 539: Image morphing. <http://www.owl.net.rice.edu/elec539/Projects97/morphjrks/themainpage.html>, Abril 2001.

- [23] Douglas B. Smythe. A Two-Pass Mesh Warping Algorithm for Object Transformation and Image Interpolation. 1990.
- [24] Stephen Milborrow, John Morkel & Fred Nicolls. The muct face database. Technical report, University Of Cape Town, diciembre 2008.
- [25] Volker Blanz & Thomas Vetter. A morphable model for the synthesis of 3d faces. Technical report, Max-Planck-Institut fur biologische Kybernetik, Tubingen, Germany.
- [26] Evan Wallace. Cs 195-g: Face morphing.
- [27] Daw-Tung Lin y Han Huang. Facial expression morphing and animation with local warping methods. Department of Computer Science and Information Engineering.
- [28] David Matsumoto y Hyi Sung Hwang. Reading facial expressions of emotion. Technical report, Army Research Institute, 2011.
- [29] Javier Gorgas y Nicolás Cardie. Análisis de datos y Estadística Avanzada, Máster Interuniversitario de Astrofísica. Master's thesis, Departamento de Astrofísica y Ciencias de la Atmósfera, Facultad de Ciencias Físicas, Universidad Complutense de Madrid, 2011.
- [30] Paul Yiu. Introduction to the geometry of the triangle. Technical report, Department of Mathematics, Florida Atlantic University, 2011.

ANEXOS

A. Interfaz

A.1. Imágenes del Proyecto Finalizado

A continuación se muestran algunas imágenes de la interfaz de la aplicación funcionando.

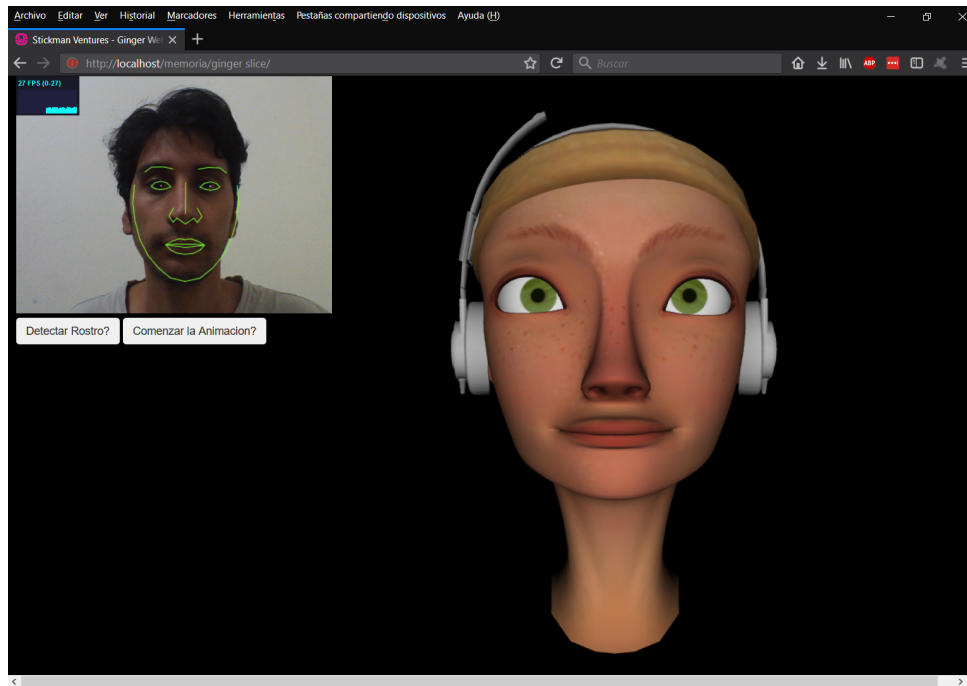


Figura A.1: Cara neutra.

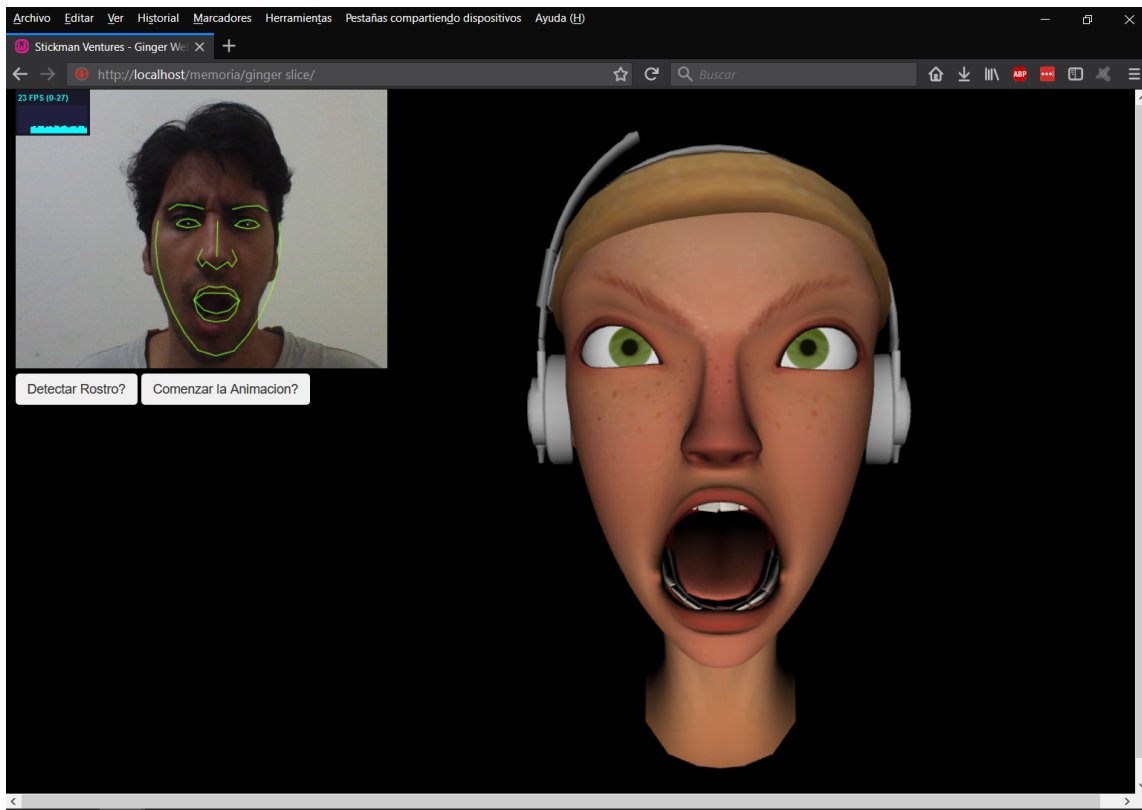


Figura A.2: Cara enojada.

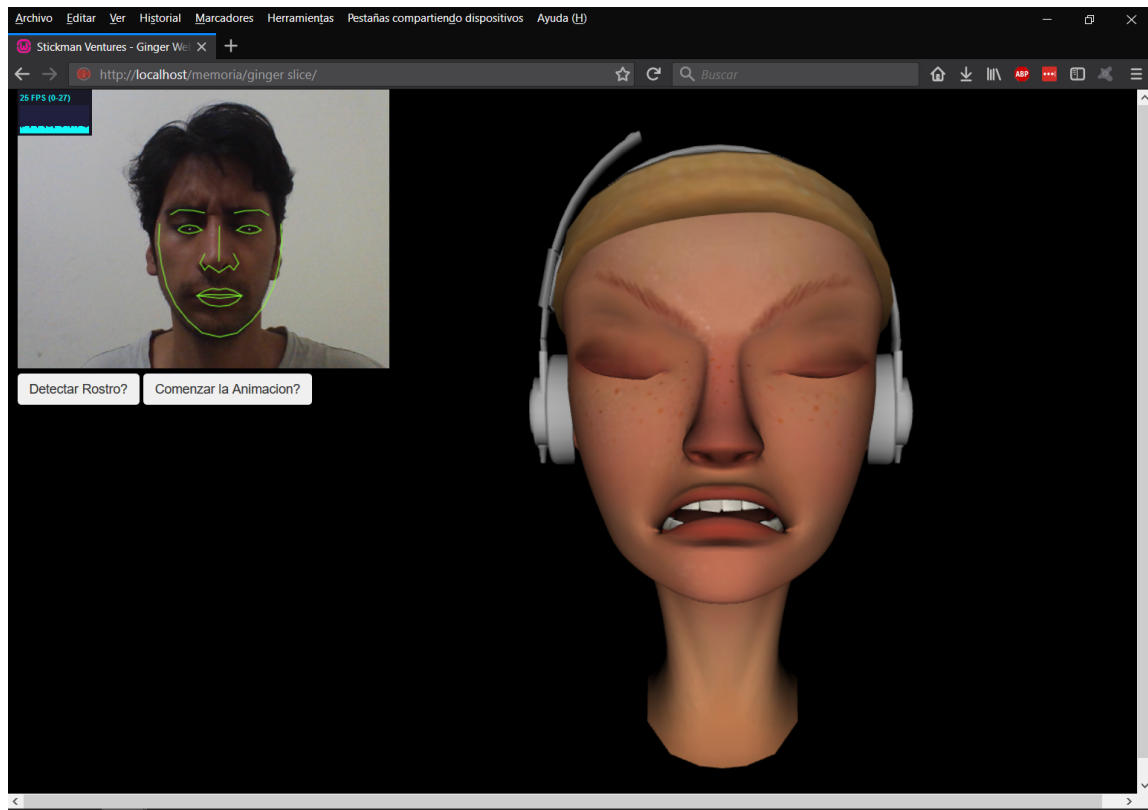


Figura A.3: Cara enojada con ojos cerrados.

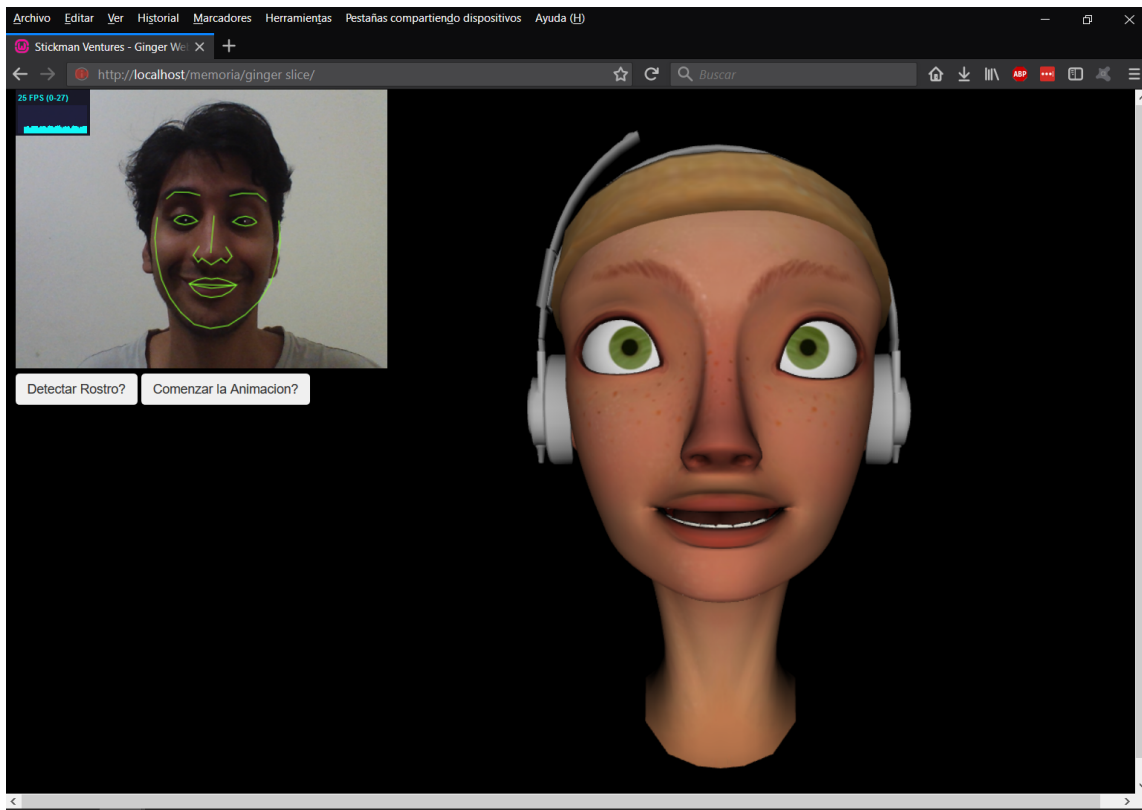


Figura A.4: Cara feliz.

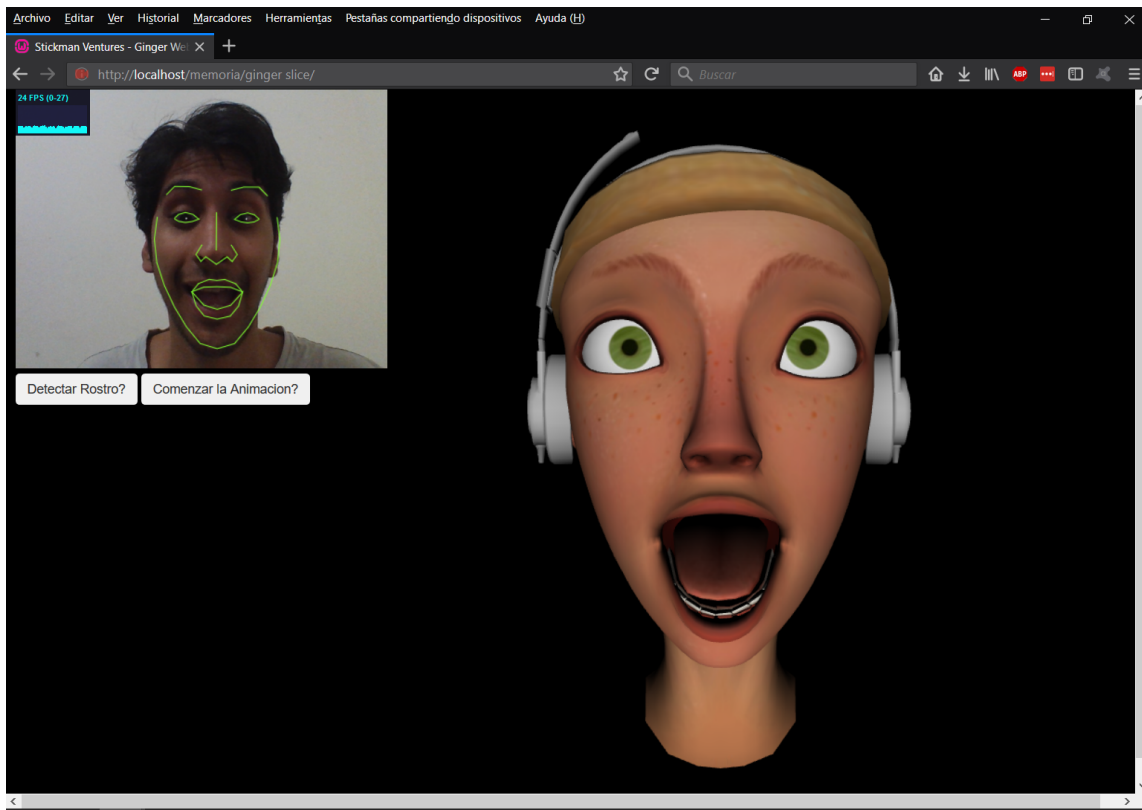


Figura A.5: Cara feliz con boca abierta.

B. Encuesta

B.1. Saludo e instrucciones iniciales

El guion que se ocupo para realizar la encuesta fue el siguiente: “Muy buenos dias/tarde/noche, mi nombre es Nicolas Garrido, estoy estudiando Ingeniería Civil en Computación en la Universidad de Talca, Facultad de Ingeniería , soy alumno memorista y estoy realizando una encuesta que ayudara a evaluar mi proyecto de titulación y me gustaría que usted la contestarla. La encuesta no demorara más de 15 minutos y consiste en evaluar una serie de imágenes donde usted tendrá que determinar que emoción representa mejor el rostro que aparece. ¿Estaría dispuesto a responder la encuesta?”

Si la persona accede a realizar la encuesta se procede a leer el documento de Consentimiento Informado.



CONSENTIMIENTO INFORMADO

Título del Proyecto: Animación de un rostro 3d guiado por Detección Facial

Patrocinante: Ingeniería Civil en Computación de la Universidad de Talca.

Estimado Sr. (Sra., Srta):

El propósito de este documento es entregarle toda la información necesaria para que Ud. pueda decidir libremente si desea participar en la **INVESTIGACIÓN** que se le ha explicado verbalmente, y que a continuación se describe en forma resumida.

Resumen del proyecto

Objetivo:

El fin de este estudio es determinar qué tan fácil/difícil se le hace a una persona entre 18 a 28 años identificar la emoción que está expresando un rostro de una persona real y el rostro de una persona tridimensional.

Esto se hace con el fin de evaluar una aplicación que detecta el rostro de una persona y captura sus expresiones para luego ser replicadas en un modelo tridimensional, todo esto en tiempo real.

Procedimientos:

La encuesta consta de una serie de imágenes con diferentes expresiones de Alegría, Enojo y Sorpresa. Estas imágenes serán de un modelo tridimensional en específico y de rostros de personas, las imágenes de las personas reales serán de rostros aleatorios y de diferente sexo.

Las personas encuestadas serán de diferente sexo y tendrán un rango de edad entre los 18 a 28 años, el rango de edad se fijó entre esos valores ya que el uso de la aplicación está orientada a estudiantes universitarios.

El número total de voluntarios requeridos para esta encuesta es de 15, en donde los participantes tendrán que, para cada imagen, elegir entre las 3 diferentes emociones mencionadas anteriormente.

Beneficios:

El fin de realizar esta aplicación es ayudar a los jóvenes estudiantes que quieren incursionar en el mundo de la animación de objetos tridimensionales en el área del entretenimiento y no tienen los medios, conocimientos o recursos para hacerlo.

Riesgos:

No existen riesgos o molestias asociadas en la participación de este estudio, más allá del tiempo que tarde en contestar la encuesta.

La decisión de participar es absolutamente suya. Puede aceptar o rechazar la investigación, e incluso arrepentirse de su primera decisión, sin ningún problema.

Usos potenciales de la investigación

El potencial uso que tendrá esta investigación tiene que ver con el área del entretenimiento, ya que la aplicación está enfocada en el uso de modelos tridimensionales para la creación de animaciones.

Costos

Serán asumidos por el estudio y si ocurriera alguna complicación derivada directamente de la participación en él, su costo será asumido por el investigador.

Compensaciones

No existen gastos involucrados de movilización, alimentación, u otros, para el participante en la investigación de este estudio.

Confidencialidad

Los datos serán almacenados en una computadora que tendrá una contraseña en la cual solo el co-investigador podrá tener acceso a ella, los datos estarán disponibles para el usuario y para el investigador. Los datos serán almacenados durante un año desde que se realizó la encuesta. Es importante recalcar que los datos obtenidos sólo se utilizarán para efectos de esta investigación y solo los resultados generales se harán públicos sin los nombres de los participantes para asegurar su anonimato.

En caso de una emergencia o duda está disponible el co-investigador Nicolás Garrido, Teléfono +56991524309, email: nigarrido09@alumnos.ugalca.cl

Así como también las vías de comunicación con el investigador responsable, Rodolfo Allendes, Teléfono +56752201754, email: rallendes@ugalca.cl

La manera de cómo se comunicarán los resultados de la investigación a los participantes será vía email y se garantizará el acceso a toda información nueva relevante por el mismo medio.

Estimado participante recuerde que la decisión de participar es absolutamente suya. Puede aceptar o rechazar la investigación, e incluso arrepentirse de su primera decisión, sin ningún problema o represalia para usted



ACEPTACIÓN:

He leído el documento, entiendo las declaraciones contenidas en él y la necesidad de hacer constar mi consentimiento, para lo cual lo firmo libre y voluntariamente, recibiendo en el acto copia de este documento ya firmado.

Yo,(nombre completo), Cédula de identidad o pasaporte N°, de nacionalidad....., mayor de edad, con domicilio en

Consiento en participar en la investigación denominada: “Animación de un rostro 3d guiado por Detección Facial”, y **autorizo** al señor Rodolfo Allendes, investigador responsable del proyecto y/o a quienes éste designe como sus colaboradores directos y cuya identidad consta al pie del presente documento, para realizar el (los) procedimiento (s) requerido (s) por el proyecto de investigación descrito.

Fecha:/...../.....

Hora:

Firma de la persona que consiente:

Investigador responsable: Rodolfo Allendes

Nombre

Firma

Co-investigador 1: Nicolás Garrido

Nombre

Firma



RECHAZO

He leído el documento, entiendo las declaraciones contenidas en él. Sin embargo, rechazo otorgar mi consentimiento, para lo cual firmo libre y voluntariamente el siguiente documento, recibiendo en el acto copia de éste ya firmado.

Yo,(nombre completo), Cédula de identidad o pasaporte N°....., de nacionalidad....., mayor de edad, con domicilio en

No Consiento en participar en la investigación denominada: "Animación de un rostro 3d guiado por Detección Facial", y **no autorizo** al señor Rodolfo Allendes, investigador responsable del proyecto y/o a quienes éste designe como sus colaboradores directos y cuya identidad consta al pie del presente documento, para realizar el (los) procedimiento (s) requerido (s) por el proyecto de investigación descrito.

Fecha:/...../.....

Hora:

Firma de la persona que consiente:

Investigador responsable: Rodolfo Allendes

Nombre

Firma

Co-investigador 1: Nicolás Garrido

Nombre

Firma



REVOCACIÓN

Mediante la presente revoco lo anteriormente firmado, para lo cual firmo este nuevo documento libre y voluntariamente, recibiendo en el acto copia de este documento ya firmado.

Yo,(nombre completo), Cédula de identidad o pasaporte N° de nacionalidad....., mayor de edad, con domicilio en,
Revoco lo anteriormente firmado.

Fecha:/...../.....

Hora:

Firma de la persona que consiente:

Investigador responsable: Rodolfo Allendes

Nombre

Firma

Co-investigador 1: Nicolás Garrido

Nombre

Firma

B.2. Encuesta realizada

La encuesta realizada consta de 25 preguntas y muestra las diferentes imágenes de Alegría, Enojo y Sorpresa.

A continuación se encuentran las preguntas y sus respuestas con sus respectivos porcentajes.

Encuesta para detectar emociones según una imagen.

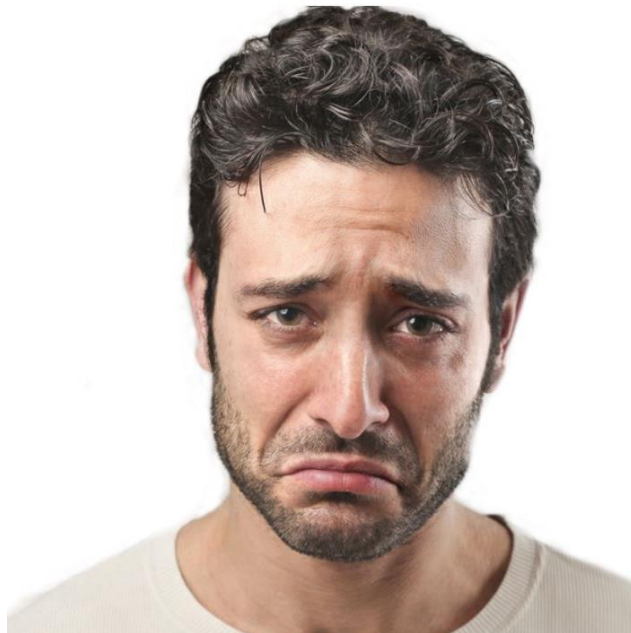
La encuesta consta de una serie de imágenes con diferentes expresiones de Alegría, Enojo y Sorpresa. Estas imágenes serán de un modelo tridimensional en específico y de rostros de personas (reales), las imágenes de las personas serán de rostros aleatorios y de diferente sexo.

Usted tendrá que evaluar una imagen y elegir UNA opción entre las diferentes emociones mencionadas anteriormente.

El fin de este estudio es determinar qué tan fácil/difícil se le hace a una persona entre 18 a 28 años identificar la emoción que está expresando un rostro de una persona real y el rostro de una persona tridimensional.

Esto se hace con el fin de evaluar una aplicación que detecta el rostro de una persona y captura sus expresiones para luego ser replicadas en un modelo tridimensional, todo esto en tiempo real.

Ejemplo: ¿Qué emoción representa mejor esta imagen?



- Alegría
- Tristeza
- Asco
- Más de una
- Ninguna de las anteriores

En este caso si usted dice que el hombre esta "Triste", selecciona esa opción.

Datos Personales

Nombre y Apellido (Opcional)

Fecha de Nacimiento

Día: _____ Mes: _____ Año: _____

Correo Electrónico

Genero

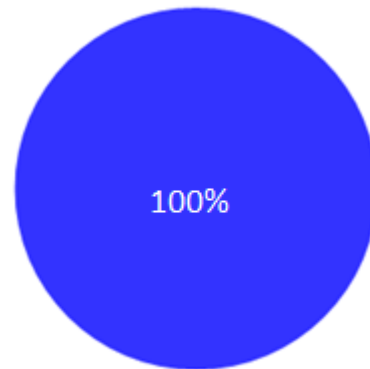
- Masculino
- Femenino

1- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



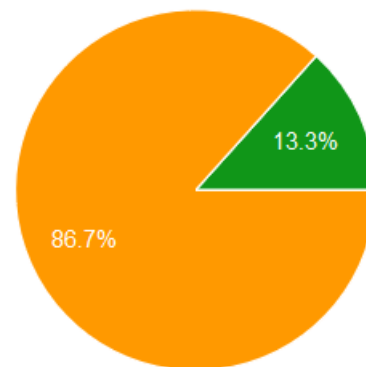
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

2- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



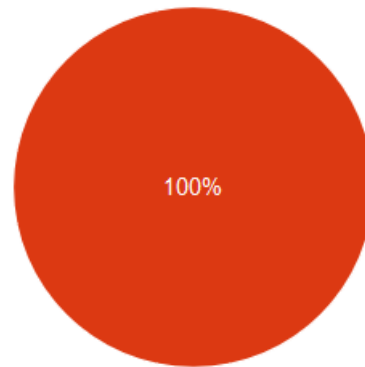
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

3- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



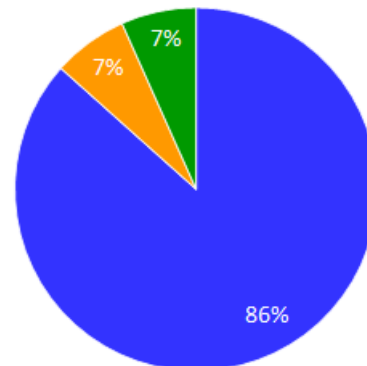
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

4- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



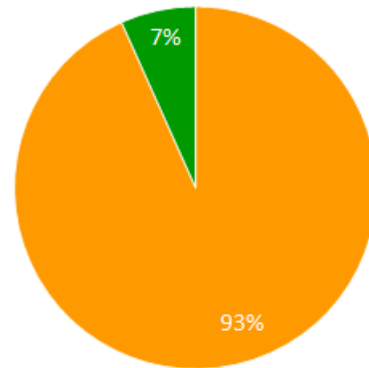
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

5- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



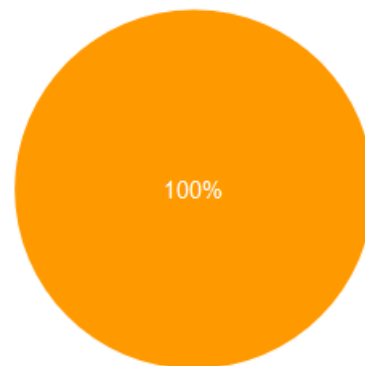
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

6- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



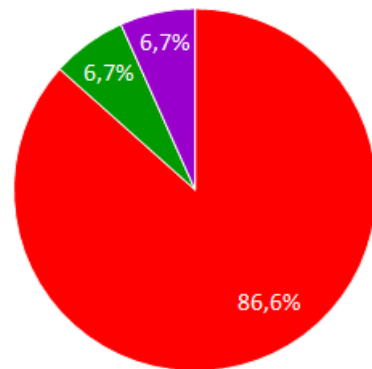
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

7- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



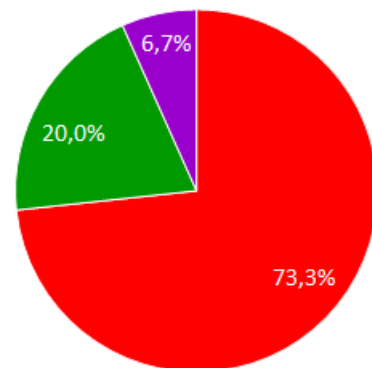
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

8- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



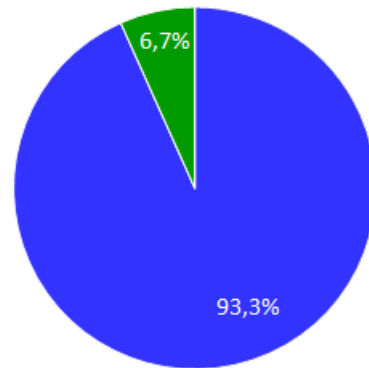
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

9- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



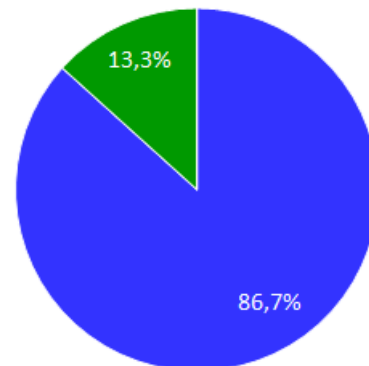
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

10- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



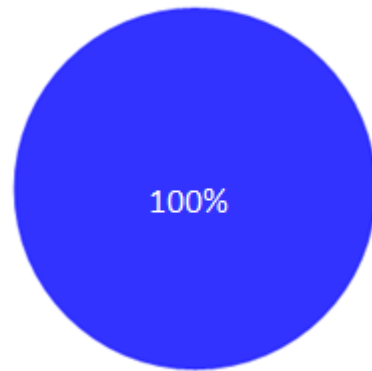
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

11- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



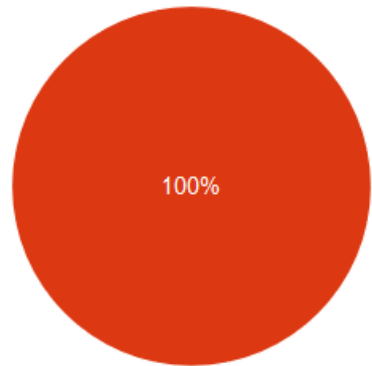
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

12- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



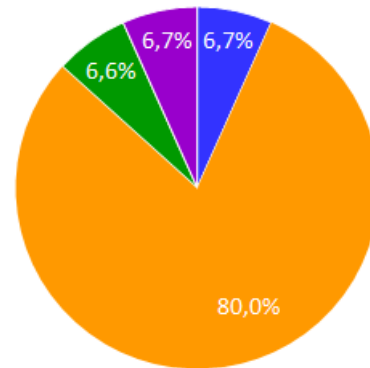
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

13- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



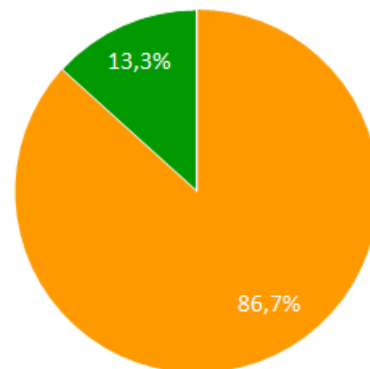
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

14- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



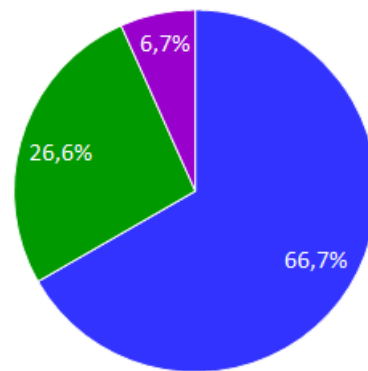
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

15- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



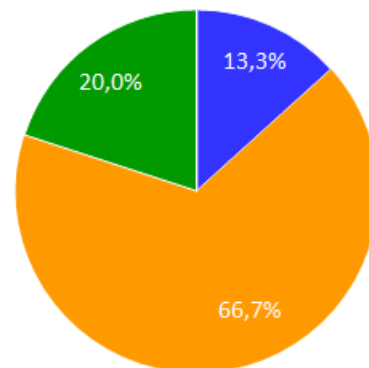
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

16- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



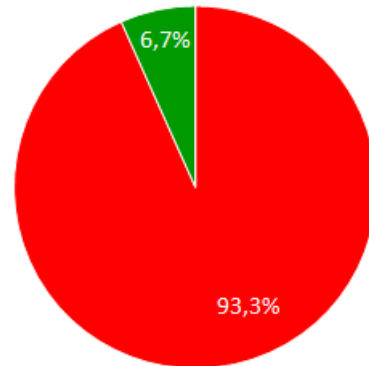
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

17- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



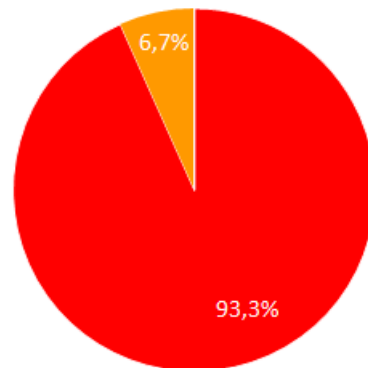
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

18- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



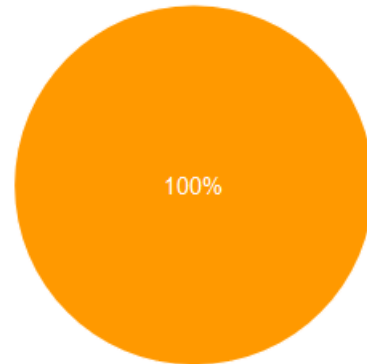
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

19- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



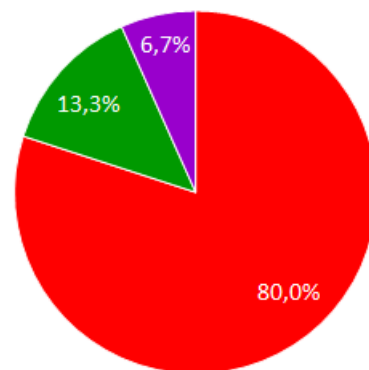
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

20- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



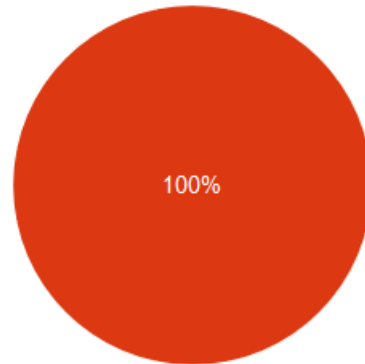
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

21- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



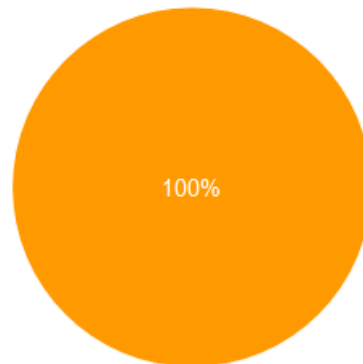
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

22- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



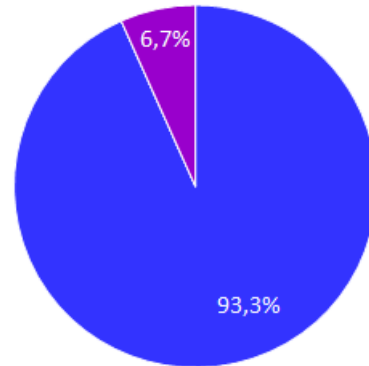
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

23- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



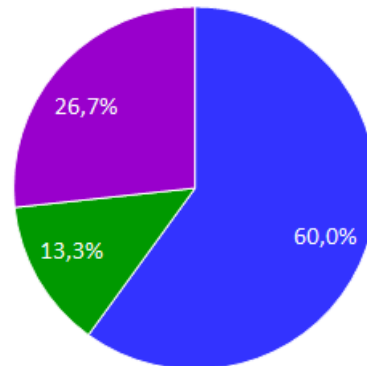
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

24- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



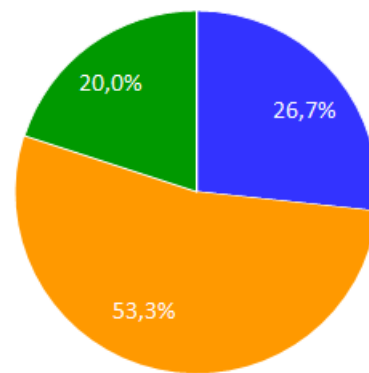
- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

25- ¿Qué emoción representa mejor esta imagen?



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores

Respuesta:



- Alegría
- Enojo
- Sorpresa
- Más de una
- Ninguna de las anteriores