ISSN 1870-4069

Reconocimiento de gestos dinámicos para la manipulación de imágenes

Damian A. Michel-Vera[1], Francisco J. Hernandez-Lopez[2], Anabel Martin-Gonzalez[1]

[1] Universidad Autónoma de Yucatán, Mérida, Yucatán, México

[2] CONACYT Centro de Investigación en Matemáticas, Mérida, Yucatán, México

damian-michel@hotmail.com, fcoj23@cimat.mx, amarting@correo.uady.mx

Resumen. El presente artículo muestra los resultados obtenidos al aplicar el reconocimiento de gestos dinámicos de una mano con el fin de manipular imágenes en tiempo real. Mediante el uso de un sensor de movimiento (*Leap Motion*) se obtuvo una base de datos de las posiciones tridimensionales de las puntas de los dedos y del centro de la palma de la mano en cada instante de tiempo, de 8 gestos dinámicos correspondientes a 8 acciones aplicadas a una imagen. A partir de estas posiciones, se generaron tres diferentes conjuntos de características y se les aplicó el algoritmo de alineamiento temporal dinámico (DTW, por sus siglas en inglés) para obtener un discriminante que permita clasificar los gestos de la mano y con base en esto analizar los resultados obtenidos.

Palabras clave: reconocimiento de gestos, sensor de movimiento, leap motion, alineamiento temporal dinámico, programación dinámica**.**

Recognition of Dynamic Gestures for Image Manipulation

Abstract. The present article shows the results obtained from applying pattern recognition of dynamic gestures of a hand to manipulate images in real time. Using a motion sensor (*Leap Motion*) a database was obtained of the three-dimensional positions of the fingertips and the palm center at each instant of time, there were 8 dynamic gestures corresponding to 8 actions applied to an image. From these positions, three different sets of characteristics were generated and the dynamic time warping (DTW) algorithm was applied, to obtain a discriminant that allows classifying the hand gestures and with this, analyze the obtained results.

Keywords: gesture recognition, motion sensor, leap motion, dynamic time warping, dynamic programming.

1. Introducción

El reconocimiento y clasificación de elementos ha sido un tema que se ha estudiado por un largo tiempo. Existe una gran cantidad de métodos para llevar a cabo la tarea de diferenciar dos o más clases, con el fin de crear sistemas capaces de tomar decisiones de manera automática para resolver distintas situaciones en tiempo y forma.

En cuanto a sistemas controlados por gestos, existen trabajos que toman como entrada, solo el video obtenido a partir de una cámara monocular [1,2], lo cual incluye el problema de detectar la parte del cuerpo que va a realizar el gesto, a través de la secuencia de imágenes.

Por otro lado, hay diversos productos que facilitan el control en estos sistemas con base en el reconocimiento de gestos, como es el caso del Samsung Galaxy S4 que implementa un sistema denominado *Air Gesture*, con el cual, mediante el movimiento de las manos se puede desplazar uno a través de una página, mostrar la hora e incluso se puede contestar una llamada activando el "manos libres" [3].

Otro producto, es el sistema que implementa Microsoft con el Xbox, captura movimientos corporales con el sensor Kinect, en tiempo real, y los interpreta como gestos para controlar el menú del sistema y poder disfrutar del uso de algunos títulos de juegos.

Este trabajo pretende comparar tres conjuntos de características en la clasificación de gestos de una mano a través del tiempo, usando el algoritmo conocido como alineamiento temporal dinámico (DTW) que presenta una medida no lineal y que permite comparar secuencias de gestos con formas similares a pesar del desfase temporal.

El estado del arte ha demostrado la eficiencia del uso del algoritmo DTW en el reconocimiento inteligente de gestos manuales [4] y para el desarrollo de un prototipo grabador de gestos [5], después de haberlo comparado con diversos algoritmos.

Por otro lado, otros trabajos han mostrado algunos ejemplos usando conjuntos de características de las posiciones 3D de los dedos de la mano, investigados mediante el uso del sensor *Leap Motion* [6,7].

Este artículo está dividido en 5 secciones, de tal manera que en la sección 2 se describen las muestras y características utilizadas para realizar los experimentos. Luego, en la sección 3 se explica el método utilizado para clasificar las muestras y el proceso general que se llevó a cabo, posteriormente en la sección 4 se muestran los resultados obtenidos de los experimentos y en la sección 5 se dan las conclusiones y las proyecciones futuras.

2. Adquisición de datos

2.1. Tipos de muestras

El sensor *Leap Motion* consta de un tamaño de 7.5 cm x 2.5 cm x 1.1 cm y contiene dos cámaras ubicadas en sus extremos, cada cámara cuenta con un sensor monocromático sensible al infrarrojo. Este dispositivo, también contiene 3 leds que se

encargan de mantener una iluminación uniforme en la zona de cobertura, y además protege a los sensores de una posible saturación de luz [8]. Trabaja en un espacio físico como se puede apreciar en la Fig. 1, el cual posee un ángulo de inclinación en las partes de abajo, lo que acaba generando una cúpula incompleta, dato que hay que tomar en cuenta para su uso y programación.

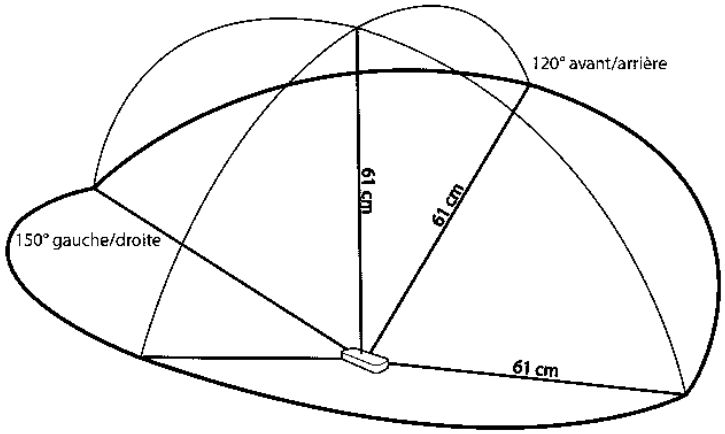


Fig. 1. Campo de trabajo del sensor *Leap Motion* [8].

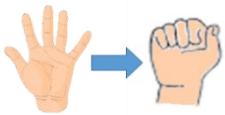
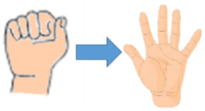
A partir del sensor *Leap Motion*, se tomaron 8 tipos distintos de muestras de gestos con la mano derecha. Estos gestos se muestran en la Tabla 1 y fueron diseñados para manipular imágenes de forma intuitiva. Cada muestra contiene un conjunto de registros (*frames*) de las posiciones (x, y, z) de cada punta de los dedos y el centro de la palma de la mano. Las muestras fueron capturadas en un periodo de 3 segundos, con un número de *frames* variables, ya que no se tiene un control de cuantos cuadros puede capturar el sensor en un tiempo determinado. Para cada gesto se capturaron 20 muestras, generando una base de datos de 160 muestras.

2.2. Estructura de las características de las muestras

Una vez obtenidas las características, estas se almacenan en un vector, de modo que siguen la forma c_i^j en donde:

- c: Puede ser x, y, z para las coordenadas en esos ejes o d si es una distancia entre uno de los dedos y el centro de la palma de la mano.
- i: Indica el número de dedo comenzando desde el pulgar al meñique. El último número es el centro de la palma de la mano.
- j: Indica el número de *frame*.

Tabla 1. Tipos de gestos para la manipulación de imágenes. Las figuras fueron tomadas de [9].

Gesto	Imagen	Gesto	Imagen
Gesto 1: "Agrandar Imagen"		Gesto 2: "Encoger imagen"	
Gesto 3: "Señalar"		Gesto 4: "Mover imagen"	
Gesto 5: "Acercar imagen"		Gesto 6: "Alejar imagen"	
Gesto 7: "Girar imagen a la derecha"		Gesto 8: "Girar imagen a la izquierda"	

Para un conjunto de características conformado por los puntos tridimensionales (x, y, z) de las puntas de los dedos y el centro de la palma, se tiene la siguiente matriz presente en la ecuación (1):

$$\begin{matrix} x_1^1 & y_1^1 & z_1^1 & x_2^1 & y_2^1 & z_2^1 & \dots & x_6^1 & y_6^1 & z_6^1 \\ x_1^2 & y_1^2 & z_2^2 & x_1^2 & y_1^2 & z_1^2 & \dots & x_6^2 & y_6^2 & z_6^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_1^n & y_1^n & z_1^n & x_2^n & y_2^n & z_2^n & \dots & x_6^n & y_6^n & z_6^n \end{matrix}, \quad (1)$$

en donde n es igual a la cantidad de *frames* de esa muestra, y no es necesariamente igual para todas las muestras del mismo gesto. En este trabajo, se manejaron los siguientes tres tipos de características:

– Puntos (x, y, z) de la punta de los dedos y el centro de la palma de la mano, ordenados como se indica en la ecuación (2):

$$P = \{x_1^1 \quad y_1^1 \quad z_1^1 \quad x_2^1 \quad y_2^1 \quad z_2^1 \quad \dots \quad x_6^1 \quad y_6^1 \quad z_6^1\}. \quad (2)$$

– Distancias de la punta de los dedos al centro de la palma de la mano, ordenados como se indica en la ecuación (3):

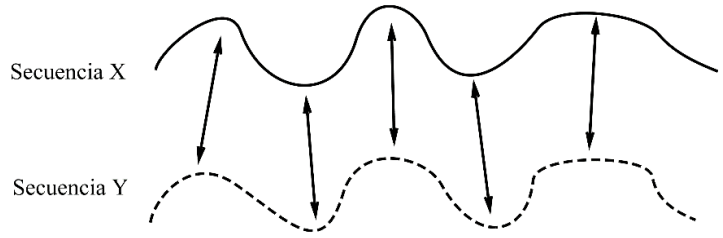


Fig. 2. Alineamiento de dos señales dependientes del tiempo.

$$D = \{d_1^1 \quad d_2^1 \quad \dots \quad d_5^1\}. \tag{3}$$

– Combinación de ambos elementos P y D mediante una concatenación, ordenados como se indica en la ecuación (4):

$$C = \{x_1^1 \quad y_1^1 \quad z_1^1 \quad x_2^1 \quad y_2^1 \quad z_2^1 \quad \dots \quad x_6^1 \quad y_6^1 \quad z_6^1 \quad d_1^1 \quad d_2^1 \quad \dots \quad d_5^1\}. \tag{4}$$

Un gesto puede cambiar de persona a persona, ya sea por el tamaño de la mano, por no colocar el sensor alineado, por la velocidad con que se realice el gesto e incluso por los *frames* que pueda o no captar el sensor dependiendo de las condiciones de luz del ambiente. Por estas razones, utilizamos el algoritmo DTW, el cual permite hallar similitudes entre este tipo de muestras a pesar de su desfase en el tiempo.

3. Metodología

3.1. Método de clasificación

El algoritmo de alineamiento temporal dinámico (DTW por sus siglas en inglés), es una técnica que permite medir la similitud entre dos señales que pueden variar en tiempo o velocidad, dejando que las señales se comporten de forma elástica para encontrar su respectiva similitud como se puede apreciar en la Fig. 2.

Originalmente, esta técnica fue utilizada para reconocer palabras en el área del estudio del reconocimiento de voz [5]. Esta técnica utiliza la programación dinámica, haciendo que se descomponga el problema de hallar la similitud de las cadenas a través de varias etapas resueltas en diversos estados, en donde la resolución de cada uno de estos se va dando mediante un cálculo recursivo de los estados anteriores.

Dadas dos señales $X = \{x_1, x_2, \dots, x_n\}$ y $Y = \{y_1, y_2, \dots, y_m\}$, el algoritmo DTW crea una matriz $D_{(n+1)\times(m+1)}$, con posiciones (i, j) para $i = 0, \dots, n$ y $j = 0, \dots, m$ que sigue la siguiente función recursiva presentada en la ecuación (5):

$$D(i,j) = d\big(x_i, y_j\big) + \min\big(D(i-1,j), D(i,j-1), D(i-1,j-1)\big), \tag{5}$$

para $i = 1, \dots, n$ y $j = 1, \dots, m$. La primera fila y la primera columna de la matriz D se encuentran inicializadas en ∞. La función $d\big(x_i, y_j\big)$ es una función de distancias entre

los puntos recibidos, esta puede ser la distancia Euclidiana, el absoluto de la diferencia de esos puntos o alguna otra medida lineal [4].

Dentro de este método se presentan ciertas restricciones para asegurar su correcto funcionamiento [10]:

– *Monotonicidad:* Los puntos deben de estar ordenados con respecto al tiempo, cumpliendo las condiciones indicadas en la ecuación (6):

$$x_{k-1} \leq x_k \quad y \quad y_{k-1} \leq y_k. \tag{6}$$

– *Continuidad:* El siguiente punto en la malla debe de ser vecino del anterior cumpliendo las condiciones indicadas en la ecuación (7):

$$x_k - x_{k-1} \leq 1 \quad y \quad y_k - y_{k-1} \leq 1. \tag{7}$$

– *Ventana de deformación:* Los puntos posibles deben de estar dentro de la siguiente ventana definida en la ecuación (8):

$$|x_k - y_k| \leq w, \qquad con\ w, el\ ancho\ de\ la\ ventana. \tag{8}$$

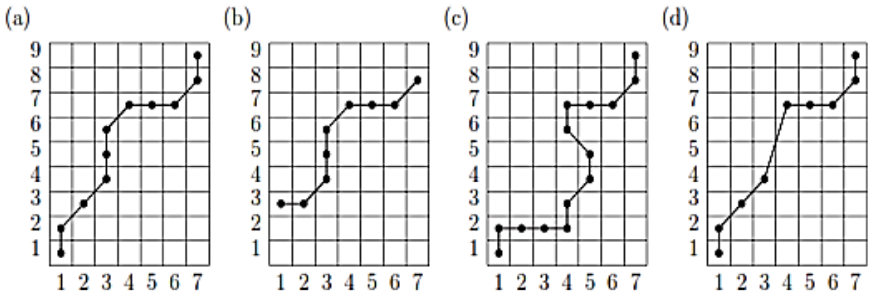


Fig. 3. Ejemplos de las condiciones del DTW [11] (a) Camino correcto (b) Condiciones de frontera que no se cumplen (c) Monotonicidad no se cumple (d) Continuidad no se cumple.

– *Restricción de pendiente:* La curvatura o pandeo del camino no debe de ser excesivamente larga en una sola dirección.

– *Condiciones de frontera:* Como se presenta en la ecuación (9), los puntos de inicio y termino del método deben ser:

$$x_1 = 1, \qquad y_1 = 1\, y\ x_k = n, \qquad y_k = m \tag{9}$$

En la Fig. 3, se puede apreciar un ejemplo gráfico del algoritmo, en donde algunas de las condiciones mencionadas previamente se cumplen y otras en donde no.

Tabla 2. Tabla de medidas estadísticas tomadas de [12].

Medida	Fórmula
Exactitud promedio (AA)	$\frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}$
Tasa de error (ER)	$\frac{\sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_i + fn_i + fp_i + tn_i}}{l}$
Precisión M (PM)	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$
Exhaustividad M (RM)	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}$
Medida-F M (FM)	$\frac{(\beta^2 + 1) * PM * \mathrm{RM}}{\beta^2 * PM + \mathrm{RM}}$

Tabla 3. Resultados estadísticos del entrenamiento, usando el 70% de la base de datos.

Tipos de Características	AA	ER	PM	RM	FM
Distancia (D)	91.29	8.70	61.15	65.17	63.10
Puntos (P)	92.63	7.36	75.97	70.53	73.15
Combinado (C)	**93.52**	**6.47**	**79.51**	**74.10**	**76.71**

Tabla 4. Resultados estadísticos de las pruebas, usando el 30% de la base datos.

Tipos de Características	AA	ER	PM	RM	FM
Distancia (D)	**91.67**	**8.33**	60.44	**66.67**	63.40
Puntos (P)	90.10	9.90	67.34	60.42	63.69
Combinado (C)	91.15	8.85	**73.04**	64.58	**68.55**

3.2. Implementación

Para la realización de este proyecto se utilizó el lenguaje C++ de visual studio 2017 con las librerías de la versión de desarrolladores de *Leap Motion* 3.2.1+45911, que contienen una API para el lenguaje C++ facilitando el controlar, calibrar y comprobar el correcto funcionamiento del dispositivo.

En primera instancia, se procedió a capturar el conjunto de muestras de los 8 gestos a distintas personas para crear la base de datos, se procedió entonces a tomar las muestras guardadas y tanto centralizarlas como a normalizarlas para generalizar su uso.

Posteriormente se empezaron a crear las combinaciones (descritas en la Sección 2.2) según el tipo de características que serían evaluadas. Una vez obtenidos todos estos elementos, se aplicó el algoritmo DTW a lo largo de cada una de las 8 clases de los

Tabla 5. Matriz de confusión usando el tipo de característica combinado (C), para la etapa de pruebas.

		Valor Predicho							
		G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**
Valor Real	**G1**	**0.83**	0.00	0.17	0.00	0.00	0.00	0.00	0.00
	G2	**0.33**	**0.33**	0.17	0.00	0.17	0.00	0.00	0.00
	G3	0.00	0.17	**0.83**	0.00	0.00	0.00	0.00	0.00
	G4	**0.50**	0.00	0.17	0.33	0.00	0.00	0.00	0.00
	G5	0.17	0.00	0.00	0.00	**0.83**	0.00	0.00	0.00
	G6	0.00	0.00	0.00	0.00	0.00	**1.00**	0.00	0.00
	G7	0.17	0.17	0.00	0.00	0.00	0.00	**0.67**	0.00
	G8	0.17	**0.33**	0.00	0.17	0.00	0.00	0.00	**0.33**
Error de omisión		0.62	0.67	0.37	0.33	0.17	0.00	0.00	0.00
Exhaustividad		65%							

gestos, de manera que se obtuvo como resultado la muestra más significativa y la muestra menos significativa de cada uno de los gestos en toda la base de datos, obteniendo de esta manera los gestos ideales y los límites de la región de confianza correspondiente.

4. Resultados

Para analizar los resultados al aplicar el algoritmo DTW sobre los 3 tipos de características P, D y C, se utilizaron las medidas estadísticas para multi-clase mostradas en la Tabla 2. Para cada clase C_i las medidas están definidas con base en los conteos de los verdaderos positivos (tp_i), verdaderos negativos (tn_i), falsos negativos (fn_i) y falsos positivos (fp_i). Luego, se calcula el promedio sobre todas las clases para la exactitud promedio (AA) y la taza de error (ER). Como en nuestros experimentos todas las clases tienen el mismo número de muestras a predecir, entonces tomamos en cuenta las medidas macro Precisión M (PM), Exhaustividad M (RM) y Medida-F M (FM), las cuales son calculadas considerando que cada clase tiene igual peso. Para el caso de la medida FM consideramos $\beta = 1$, para dar la misma ponderación a la precisión y a la exhaustividad.

En la Tabla 3 se presentan los resultados obtenidos para la parte del entrenamiento, en donde se tomó el 70% de las muestras de manera aleatoria para hallar entre estas el gesto ideal. Por otra parte, en la Tabla 4 se pueden apreciar los resultados obtenidos durante la evaluación del 30% de las muestras en la fase de pruebas.

Observamos que para el caso distancia y el combinado, se obtuvieron resultados casi iguales en AA y ER, sin embargo, se puede apreciar que el FM es mayor en el combinado, ya que la medida FM relaciona la precisión y la exhaustividad, podemos decir que el mejor de estos tipos de características es el combinado de puntos con distancias.

Tabla 6. Matriz de confusión usando el tipo de característica combinado (C), para la etapa de pruebas, quitando los gestos G1 y G2.

		Valor Predicho					
		G3	**G4**	**G5**	**G6**	**G7**	**G8**
Valor Real	**G3**	**1.00**	0.00	0.00	0.00	0.00	0.00
	G4	0.17	**0.33**	0.00	0.00	0.17	**0.33**
	G5	0.00	0.17	**0.83**	0.00	0.00	0.00
	G6	0.00	0.00	0.00	**1.00**	0.00	0.00
	G7	0.00	0.00	0.17	0.00	**0.67**	0.17
	G8	**0.33**	0.17	0.00	0.00	0.17	**0.33**
Error de omisión		0.33	0.50	0.17	0.00	0.33	0.60
Exhaustividad		69%					

Tabla 7. Matriz de confusión usando el tipo de característica combinado (C), para la etapa de pruebas, quitando los gestos G1 y G2 y aumentando la base de datos al doble.

		Valor Predicho					
		G3	**G4**	**G5**	**G6**	**G7**	**G8**
Valor Real	**G3**	**0.86**	0.00	0.00	0.00	0.00	0.14
	G4	0.14	**0.64**	0.00	0.00	0.07	0.14
	G5	0.00	0.14	**0.86**	0.00	0.00	0.00
	G6	0.00	0.00	0.00	**1.00**	0.00	0.00
	G7	0.07	0.00	0.21	0.00	**0.64**	0.07
	G8	0.14	0.07	0.07	0.00	0.07	**0.64**
Error de omisión		0.29	0.25	0.25	0.00	0.18	0.36
Exhaustividad		77%					

En la Tabla 5 se presenta la matriz de confusión usando el tipo de característica combinado para poder observar su desempeño.

Podemos observar que los valores más altos de error están en los gestos 2, 4 y 8 que corresponden a las funciones de "Encoger imagen", "Mover la imagen" y "Girar imagen a la izquierda". Esto podría deberse a que estos gestos son parecidos a través del tiempo, y que, en cada uno de ellos, las muestras usadas en el entrenamiento para hallar al gesto ideal fueron demasiado discrepantes entre sí.

En la Tabla 6 mostramos los resultados de la matriz de confusión sin considerar los gestos G1 y G2, los cuales presentan mayor error de omisión. Observamos que ahora la exhaustividad es de 69%; sin embargo, en los gestos G4 y G8 aún se obtienen exactitudes muy bajas.

Finalmente, aumentamos la base de datos al doble y esta vez en la Tabla 7 se puede observar un incremento de la exhaustividad al 77%.

5. Conclusiones

La implementación de sistemas de reconocimiento de gestos es un tema que tiene mucho futuro, pues con la constante creación de nuevos sistemas que usan AR y de compañías que están desarrollando e investigando estos temas, les darán a dispositivos como el *Leap Motion* más cabida en el uso cotidiano.

En relación con los resultados, se observó que el tipo de combinación de puntos y distancias fue el que obtuvo un mejor resultado, pero cabe recalcar que incluso cuando solo se usaron distancias, el resultado obtenido no fue tan bajo como se esperaría. La característica de distancia parece ser bastante representativa de los gestos. Al quitar los gestos G1 y G2, y aumentar la base de datos al doble, observamos que hubo un incremento significativo en la exhaustividad.

Como trabajo a futuro se planea en primer lugar conseguir una base de datos más grande para poder analizar adecuadamente los gestos, de esta forma se podrían encontrar gestos que sean más representativos que los actuales. Se planea fusionar los gestos G1 y G2, para tratar de discriminarlos después en función de su respectiva dirección de movimiento, lo mismo para los gestos G7 y G8. Además, se tiene pensado hacer una gráfica que analice el AA con respecto a la cantidad de *frames* que se usan. Se planea comprobar a partir de que cantidad de *frames* se comporta correctamente el algoritmo DTW y establecer entonces una relación entre la longitud de *frames* de muestra y los de llegada, para tomarlo en cuenta en la optimización del sistema. Posteriormente se planea crear la interfaz, en donde puedan verse en acción los gestos propuestos para la manipulación de las imágenes en tiempo real.

Referencias

1. Avilés-Arriaga, H.H., Sucar, L.E., Mendoza, C.E., Vargas, B.: Visual recognition of gestures using dynamic naive bayesian classifiers. In: 12th IEEE International Workshop on Robot and Human Interactive Communication, pp. 133–138 (2003)
2. Brethes, L., Menezes, P., Lerasle, F., Hayet, J.: Face tracking and hand gesture recognition for human-robot interaction. In: IEEE International Conference on Robotics and Automation, 2, pp. 1901–1906 (2004)
3. COMPUTER HOY: http://computerhoy.com/paso-a-paso/moviles/controla-tu-samsung-galaxy-s4-mediante-gestos-tocarlo-5139 (2018)
4. Andrade, F.: Un enfoque inteligente para el reconocimiento de gestos manuales. Bachelor's Thesis, Facultad de Ciencias Exactas de la Universidad del Centro de la Provincia de Buenos Aires (2016)
5. Ruiz K.: Desarrollo de un prototipo usando como dispositivo de interacción Leap Motion. Bachelor's Thesis, Facultad de Informática de la Universidad Politécnica de Madrid (2014)
6. Marin, G., Dominio, F., Zanuttigh, P.: Hand gesture recognition with jointly calibrated Leap Motion and depth sensor. Multimedia Tools and Applications 75(22), pp. 14991–15015 (2015)
7. Lu, W., Tong, Z., Chu, J.: Dynamic hand gesture recognition with leap motion controller. IEEE Signal Processing Letters 23(9), pp. 1188–1192 (2016)

8. SHOWLEAP: http://blog.showleap.com/2015/04/leap-motion-caracteristicas-tecnicas/ (2018)
9. DEPOSITPHOTOS: https://sp.depositphotos.com/22218953/stock-illustration-gestures.html (2018)
10. Berndt, D., Clifford J.: Using Dynamic Time Warping to Find Patterns in Time Series. In: Workshop on Knowledge Discovery in Databases, pp. 359–370 (1994)
11. Müller, M.: Information Retrieval for Music and Motion. Springer-Verlag (2007)
12. Sokolova, M., Lapalme G.: A systematic analysis of performance measures for classification tasks. Information Processing and Management 45(4), pp. 427–437 (2009)