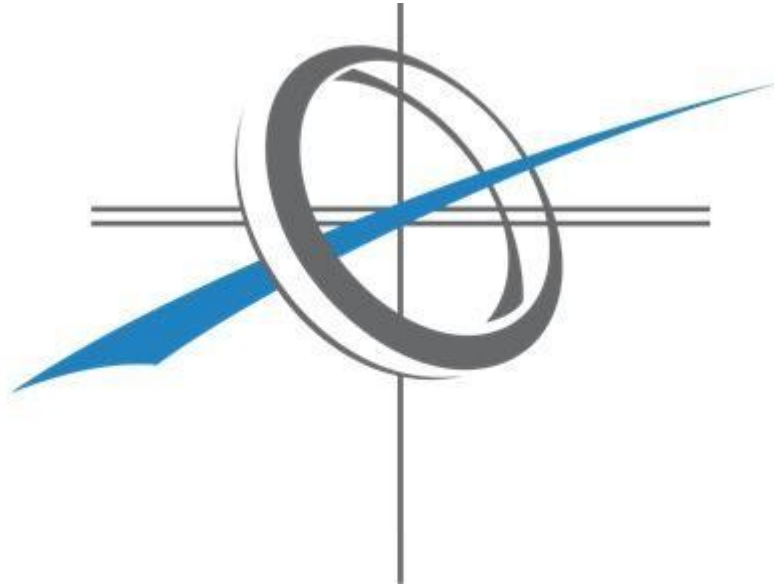


UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
SISTEMAS INFORMÁTICOS



GRADO EN INGENIERÍA DE COMPUTADORES
PROYECTO FINAL DE GRADO

Reconocimiento del movimiento de la mano
con cámara Kinect

Autor: Ana Marcos Fuertes
Tutor: Javier de Lope Asiaín

2018/2019

Resumen

A lo largo de los últimos años una de las áreas tecnológicas que más se ha desarrollado es la Inteligencia Artificial, esto ha provocado que muchas áreas a su vez también hayan podido crecer bastante. Una de ellas, es el área del estudio del comportamiento, aunque al principio si que era muy difícil poder sacar estudios muy precios, con el paso de los años y gracias al desarrollo tanto de las técnicas, como de dispositivos han podido realizarse estudios más precisos, además, pueden servir de mucha ayuda en algunos casos.

El objetivo de este trabajo es ver el resultado de la investigación y desarrollo de un sistema que, pretende reconocer el movimiento de la mano a partir de una serie de imágenes que se han obtenido de una cámara de profundidad, concretamente de la cámara Kinect. Así que, podremos ver una explicación teórica de las técnicas que se han utilizado para desarrollar el sistema, tanto de filtrado de datos, como de desarrollo de un modelo que sirva para reconocer el movimiento que se produce con la mano. Y finalmente, se podrán ver los resultados obtenidos después del desarrollo del sistema.

Abstract

Over the last few years one of the technological areas that has developed the most is Artificial Intelligence, this has caused many areas in turn have also been able to grow quite a bit. One of them is the area of behavioral study, although at first it was very difficult to get very expensive studies, over the years and thanks to the development of both techniques and devices have been able to perform more accurate studies, also can be very helpful in some cases.

The objective of this work is to see the result of the research and development of a system which, intends to recognize the movement of the hand from a series of images which have been obtained from a depth camera, specifically from the Kinect camera. Thus, we will be able to see a theoretical explanation of the techniques that have been used to develop the system, both of data filtering and of developing a model that serves to recognize the movement that is produced with the hand. And finally, you will be able to see the results obtained after the development of the system.

Índice

Resumen	3
Abstract.....	5
Índice de figuras	8
Índice de tablas	9
1. Introducción	11
1.1. Objetivos.....	12
1.2. Plan de trabajo	13
2. Estado del arte	15
3. Métodos de filtrado de datos y clasificación del movimiento.....	23
3.1. Filtro de Kalman	23
3.2. Modelos ocultos de Markov (HMM).....	27
4. Detección y reconocimiento del movimiento.....	33
4.1. Kinect.....	33
4.2. Filtrado de datos mediante filtro de Kalman.....	35
4.3. Obtención del modelo a reconocer con HMM.....	36
5. Resultados	41
5.1. Resultados obtenidos	41
5.2. Análisis de los resultados.....	43
6. Aspectos sociales, ambientales, éticos y legales	51
7. Conclusiones	53
8. Futuros trabajos	55
9. Bibliografía.....	57

Índice de figuras

Imagen 1:	11
Imagen 2:	12
Imagen 3:	17
Imagen 4:	18
Imagen 5:	19
Imagen 6:	20
Imagen 7:	20
Imagen 8:	21
Imagen 9:	26
Imagen 10:	29
Imagen 11:	29
Imagen 12:	30
Imagen 13:	31
Imagen 14:	33
Imagen 15:	34
Imagen 16:	34
Imagen 17:	35
Imagen 18:	41
Imagen 19:	42

Índice de tablas

Tabla 1:.....	28
Tabla 2:.....	37
Tabla 3:.....	38
Tabla 4:.....	38
Tabla 5:.....	44
Tabla 6:.....	44
Tabla 7:.....	45
Tabla 8:.....	46
Tabla 9:.....	46
Tabla 10:.....	47
Tabla 11:.....	48
Tabla 12:.....	48
Tabla 13:.....	49

1. Introducción

A lo largo de los últimos años, una de las áreas que más atención ha suscitado es la del estudio del comportamiento. El estudio del comportamiento tanto de animales, como de humanos es muy beneficioso, dado que ayuda a entender mejor las enfermedades y los comportamientos que pueden tener las personas o animales en determinadas situaciones. En este tipo de estudios lo que se hace es un análisis del comportamiento del sujeto, concretamente se desarrollan sistemas para que capturen el movimiento. Estos estudios también se conocen como etología computacional, como definen David J. Anderson y Pietro Perona (David J. Anderson and Prieto Perona 2014):

“El nuevo campo de la "etología computacional" es posible gracias a los avances en tecnología, matemáticas e ingeniería que permiten a los científicos automatizar la medición y el análisis del comportamiento animal.”

Al principio, la recogida de la información se basaba en observar al sujeto de estudio durante un tiempo e ir registrando a mano los datos que fueran más interesantes, un trabajo bastante arduo. Aunque, con el paso de los años la recogida de los datos se ha ido refinando gracias a los avances tecnológicos, los cuales han permitido que se lleven a cabo el desarrollo de diversas técnicas.

Una de las técnicas más utilizadas para la captura del movimiento, es la del uso de wearables, como, por ejemplo, un traje (Rokoko 2017) el cual lleva incorporados ciertos sensores, situados en lugares estratégicos del mismo, para poder capturar el movimiento con mayor precisión. Hoy en día, es uno de los métodos más utilizados para el desarrollo de videojuegos o algunos efectos especiales de películas. Aunque en este ámbito, se utilice para realizar análisis sobre ciertos movimientos que realiza el ser humano.

Con este tipo de sistemas se puede lograr que no se pierda ningún dato que se recoja. Pero, este sistema limita las posibilidades de estudio, ya que, sólo se pueden realizar durante un tiempo determinado, además de que no permiten mucha libertad, algo que sí ocurre con las cámaras.

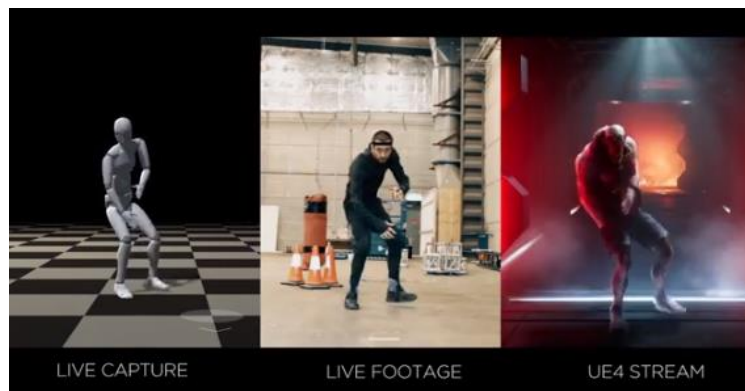


Imagen 1: Ejemplo de captura del movimiento mediante sensores (Rokoko 2017)

Otra técnica bastante utilizada para poder reconocer el comportamiento es el uso de cámaras, este método, permite mayor flexibilidad que el anterior, dado que, con esta técnica lo que se realizan son grabaciones de los sujetos de estudio. Esto permite ver las diferentes capturas y hacer un filtrado de las mismas, además de desarrollar un modelo con el que se compararán los vídeos, para después reconocer el comportamiento.

En este método podemos hacer una división dependiendo del tipo de dispositivo que se utilice, ya que, se puede elegir entre dos tipos de cámaras: cámaras RGB (dan imagen RGB) y cámaras de profundidad.

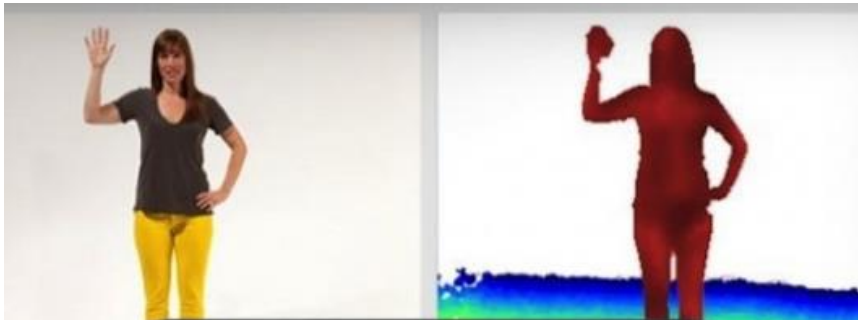


Imagen 2: Ejemplo de filtrado con la cámara Kinect para la captura de movimiento

Para el uso de las cámaras de profundidad el estudio que se realiza es muy parecido al del otro tipo de cámara, la cámara que proporciona imágenes RGB, con la diferencia de que con las cámaras de profundidad se puede hacer un filtrado de la distancia a la que se realiza el movimiento, pudiendo hacer un estudio más preciso.

Esta última técnica es la que se utilizará para realizar el desarrollo del proyecto, se hará con la cámara de profundidad Kinect, con ella se desarrollará un sistema con el que se pueda realizar el reconocimiento de ciertos movimientos que se realizan con la mano.

1.1. Objetivos

El objetivo del proyecto es el desarrollo de un sistema que pueda reconocer ciertos movimientos que realice una persona con la mano delante de un ordenador. Para ello se utiliza una cámara de profundidad (Kinect). Los movimientos se registran exclusivamente dentro de una zona, que es lo que se denomina zona objetivo.

Lo que se pretende es desarrollar un método con el que reconocer la posición inicial de la mano, comparándola con la posición final y comprobar qué tipo de movimiento se está produciendo. Para ello se utilizarán técnicas de filtrado de la imagen y modelos que permitan que el sistema “aprenda” y pueda reconocer dichos movimientos.

Para desarrollar este sistema se llevarán a cabo unos objetivos determinados, que son los siguientes:

- Realizar un estudio previo para saber el estado del arte en este campo y ver cuál es el procedimiento más adecuado para desarrollar este sistema.
- Estudiar cómo se podrá aplicar el filtro de Kalman para el filtrado de datos que se obtienen de la imagen.
- Investigar el funcionamiento de los modelos ocultos de Markov (HMM), para poder aplicarlo al sistema y así poder realizar el reconocimiento de los movimientos realizados.
- Definir y realizar una serie de experimentos para analizar el rendimiento del sistema al reconocer los movimientos.

1.2. Plan de trabajo

Para poder llevar a cabo el desarrollo de este proyecto lo primero que se hará es un amplio estudio del estado del arte en este campo, cómo se encuentra y qué tipo de métodos se pueden utilizar para el sistema que se pretende desarrollar, esto se llevará a cabo durante los primeros dos meses.

A continuación, se llevará a cabo una investigación sobre la instalación de la cámara y qué sistema operativo permite una mayor flexibilidad, para poder utilizar todos los métodos que se necesiten.

Una vez se tenga todo lo anterior, se comenzará a desarrollar el sistema, sopesando cuáles pueden ser las mejores opciones de desarrollo, como, por ejemplo, qué zona objetivo es la más adecuada o cómo se va a llevar a cabo el reconocimiento de la mano.

Y, por último, investigar cuál es el método que mejor puede adaptarse al sistema que se pretende desarrollar a la hora de reconocer el movimiento de la mano, para empezar, se hará el reconocimiento de cuatro movimientos que se pueden realizar con la mano. Mover la mano hacia arriba, hacia abajo, hacia la izquierda y hacia la derecha.

2. Estado del arte

En los últimos años, gracias a los avances tecnológicos, una de las áreas que más se ha podido desarrollar y crecer es la Inteligencia Artificial. En la actualidad, es una de las áreas donde más se está invirtiendo ya que se puede aplicar a muchos campos.

En el área del estudio del comportamiento también se ha podido notar este crecimiento, dado que, ha permitido que las técnicas que se utilizan en la actualidad sean más avanzadas que las de hace unos años. Esto ha sido posible gracias al avance tanto de los dispositivos que se utilizan como las técnicas de recogida y filtrado de datos. Con este tipo de estudios lo que se pretende es ayudar a la comprensión del cerebro, además de comprender ciertos comportamientos que se pueden tener debido a una enfermedad.

El desarrollo de las técnicas de reconocimiento puede ser muy diferente, dependiendo de los dispositivos de recogida de datos que se utilicen y de los métodos que se vayan a utilizar para tratar los datos. A continuación, se verán una serie de estudios en los que se describen las diferentes técnicas y dispositivos que se han utilizado en cada estudio.

Uno de los estudios más importantes, en lo que llamamos etología computacional, es *“Toward a Science of Computational Ethology”* (David J. Anderson and Prieto Perona 2014), en este estudio se define el término de la etología computacional, y realiza un estudio del comportamiento de la mosca del vinagre para poder comprender el funcionamiento del cerebro. Como ya se ha comentado antes, en este tipo de investigaciones, era muy tedioso el proceso de la recogida de la información. En este estudio se puede ver, como se desarrolla un sistema que aplica una serie de métodos que le permiten realizar esa recogida de información, además de hacer el reconocimiento de ciertos comportamientos.

Para el desarrollo de este tipo de sistemas hay que tener en cuenta que podemos distinguir claramente 3 fases. Una primera fase, en la que se realiza la detección del sujeto de estudio y se predice la trayectoria de su movimiento. Esto se hará mediante video; aunque, el gran problema que se indica es que para realizarlo de esta forma todavía no se ha elaborado una estandarización, lo que lleva a que algunos estudios no puedan servir como modelo para otros, o que, si lo hace, limite mucho las técnicas que se pueden utilizar para el filtrado de datos.

En la segunda fase, es en la que se realiza la clasificación del movimiento, es decir, recoger los vídeos que se han hecho y ver cuáles son las características que se buscan para reconocer el movimiento. Después se utilizan las técnicas para que, teniendo esos vídeos y los de modelo, hacer una comparación y que el sistema pueda hacer el reconocimiento. Esto nos lleva a una diferenciación entre sistemas supervisados, en los que se supervisa esa clasificación, y no supervisados, en los que se utiliza clasificadores para realizar esta tarea, por ejemplo, redes de neuronas.

Y en última fase, lo que se realiza es un estudio sobre la frecuencia con la que se realiza la acción y la probabilidad de que otro sujeto pueda seguir ese mismo comportamiento, esto puede servir para ampliar los estudios a más especies.

Esta misma estructura se ha utilizado para poder reconocer el comportamiento de la mosca del vinagre, con la que se ha intentado reconocer el comportamiento del cerebro en ciertas situaciones. Con ello han conseguido ver cuáles son las zonas que se activan en esos comportamientos, y así, después, poder ampliar este estudio a otras especies e incluso en situaciones diversas.

En el estudio “*Automated home-cage behavioural phenotyping of mice*” (Hueihan Jhuang , Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D. Steele y Thomas Serre 2010), podemos ver que se realizó sobre los ratones y el comportamiento que podrían tener en sus jaulas a lo largo de un tiempo determinado. Esto tiene como objetivo poder reconocer dichos comportamientos, sobre todo para poder saber si sufren algún tipo de enfermedad. Para realizar este estudio se ha tomado como modelo un trabajo anterior en el que se estudiaba el comportamiento humano.

La elaboración del software se ha hecho mediante machine learning, el sistema ha ido “aprendiendo” a cómo reconocer determinados comportamientos; los datos iniciales que se han utilizado son los que se han tomado a mano. La toma de datos de esta forma es bastante subjetiva, dado que la precisión de una persona es mucho menor, si la comparamos con la precisión de los datos que toma un sistema desarrollado para ello. Lo bueno es que al tratarse de un sistema que utiliza machine learning, aunque, los primeros datos que se han almacenado se hayan tomado a mano, el sistema al haber ido “aprendiendo”, ha ido registrando cada vez datos más precisos haciendo que el reconocimiento sea más preciso a su vez, con el paso del tiempo.

La toma de los datos la han realizado mediante cámaras, lo que les ha permitido realizar una serie de vídeos, de los que después se ha sacado toda la información que se necesitaba. Para poder realizar un filtrado de la información que han obtenido de los vídeos, han tenido en cuenta tanto los píxeles, como la forma de capturar el movimiento del ratón, además del fondo de la imagen. Con esto se ha separado el fondo de la imagen de lo que realmente les interesa; que son los datos que se recogen del movimiento del ratón. Para la clasificación de los comportamientos se utilizó una matriz de confusión, la herramienta más útil para estos casos. En la siguiente imagen podemos ver más claramente cómo se realiza este procedimiento.

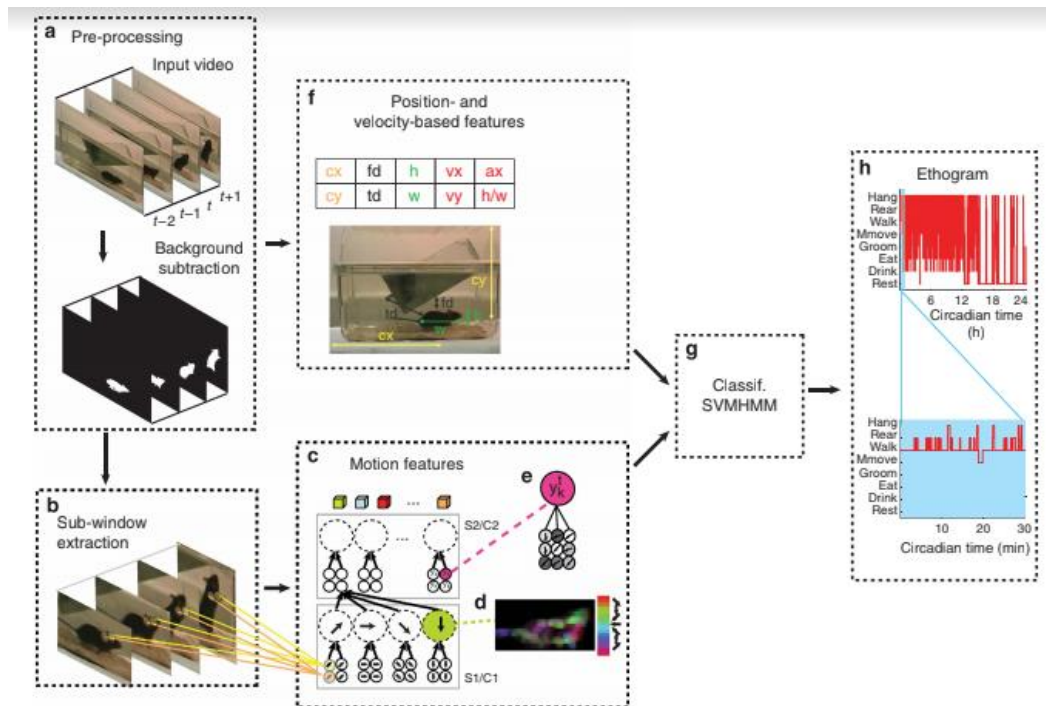


Imagen 3: Muestra del funcionamiento del sistema.

En el recuadro a, podemos ver la fase de procesamiento de la imagen, en el b, se puede ver cómo se hace la extracción de los datos, en la que vemos cómo se realiza el filtrado de datos para su clasificación en los distintos movimientos. En el f, podemos ver los datos que se han tenido en cuenta para la captura de la imagen. Finalmente, una vez se tiene todo esto, se realiza la clasificación de los datos recogidos.

Otro estudio del comportamiento animal es “*A multi-animal tracker for studying complex behaviors*” (Eyal Itskovits, Amir Levine, Ehud Cohen and Alon Zaslaver 2017) este estudio sirve para reconocer el comportamiento de varias especies de animales; para ello se realiza una separación en dos categorías, los animales vertebrados y los animales invertebrados, este tipo de sistemas se denomina Multi-Animal Tracker (MAT).

El sistema desarrollado emplea también cámaras, para grabar y capturar el movimiento, además utiliza una interfaz gráfica que permite hacer un filtrado más selectivo de las capturas que se realizan con las cámaras, ya que, permite ver las capturas que se han hecho durante la grabación y compararlas con las que sirven como modelo.

Uno de los puntos débiles que tiene este sistema, es que se suele realizar sobre más de un sujeto, lo que delimita el tiempo de grabación, ya que, el sistema no sabe distinguir con tanto sujeto de estudio. Para solucionar esto se utilizó una técnica de machine learning, Kalman type predictor, con la que se puede hacer el estudio de varios sujetos a la vez, dado que, con este filtro, dados unos datos, se puede predecir la trayectoria del sujeto, lo que hace más fácil la identificación y diferenciación de cada sujeto.

En el área del estudio del comportamiento humano podemos encontrar estudios como *“Pose Estimation with a Kinect for Ergonomic Studies: Evaluation of the Accuracy Using a Virtual Mannequin”* (Pierre Plantard, Edouard Auvinet, Anne-Sophie Le Pierres y Franck Multon 2015), en este estudio se hace una investigación de cuáles son las posturas que puede realizar un ser humano, concretamente las posturas extrañas que puede llegar a realizar.

En esta investigación, el sistema que se ha desarrollado se basa en realizar un maniquí virtual a partir del sujeto de estudio, utilizando la cámara Kinect, como se muestra en la siguiente imagen.

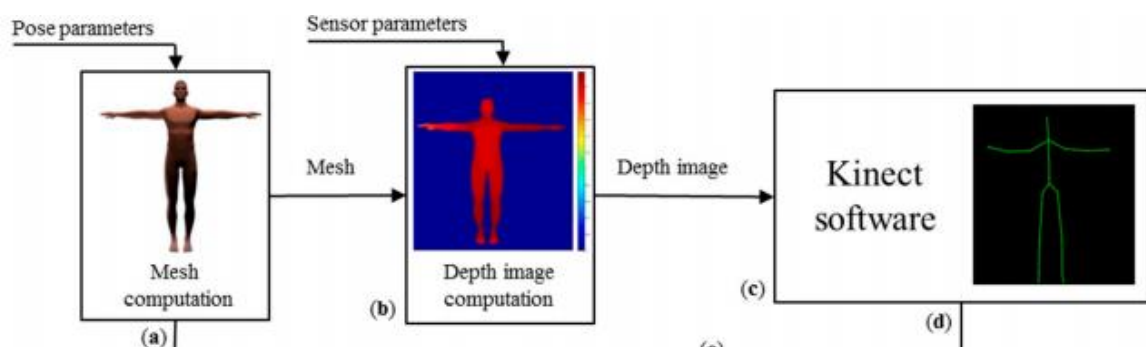


Imagen 4: Muestra del sistema maniquí virtual

Como se puede ver en la imagen anterior, el sistema captura el cuerpo del sujeto. Al hacer uso la cámara Kinect se trabaja con la cámara de profundidad, lo cual nos da como resultado la segunda imagen (Sensor parameters). Como se puede ver, al utilizarse este tipo de imagen se obtiene un filtrado bastante bueno de los datos capturados, esto les permitió poder hacer un maniquí virtual del sujeto de estudio y así poder hacer una investigación más precisa.

En este estudio para la recogida de datos de las posturas elegidas se pensó en desarrollar 3 métodos, dos de ellos se basaban en la toma de datos manual, la cual puede ser bastante subjetiva; el otro método era mediante el uso de la cámara. La gran desventaja de este método es que no se podía hacer uso de la cámara en un entorno real de trabajo, por lo que si se hacía era provocando alguna situación en un área controlada, lo que a su vez provocaba que los datos no fueran tan exactos como lo podrían haber sido. Para solucionar esto lo que hicieron fue realizar una comparación entre unos datos y otros para poder tener una perspectiva lo más exacta posible, haciendo que el estudio fuera lo más realista posible en cuanto a los datos que se recogieron.

También se ha utilizado la cámara Kinect para hacer una estimación de la pose de la cabeza, como en *“An evaluation of 3D head pose estimation using the Microsoft Kinect v2”* (John Darby, María B. Sánchez, Penelope B. Butler and Ian D. Loram 2016), en este caso se utiliza la versión 2 de la cámara Kinect, que es un sensor que no necesita

calibración para la estimación de la pose y un poco más pequeño, algo que, para este tipo de estudios favorece bastante. Esto se debe a que cuanto menos moleste al sujeto de estudio mejor. Además, de que los datos que se tomen van a ser mucho más precisos.

El sistema que se ha utilizado para la toma de datos se basa en el uso de la cámara Kinect junto con unos marcadores en la cara. En este caso sólo se han estudiado los movimientos de la cabeza, no se he entrado a profundizar en los gestos que puede realizar la persona, aunque se hayan utilizado sensores en la cara.

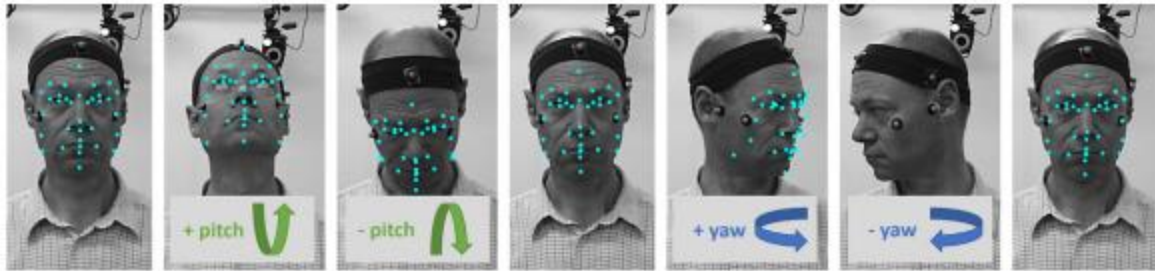


Imagen 5: Reconocimiento de la pose

Como podemos ver en la imagen anterior, se trata de distintas imágenes tomadas con las distintas poses que reconoce el sistema. En esta imagen se puede ver claramente los marcadores que se han estimado para la cara. Estos sensores se utilizan para ver si se trata de movimientos verticales, ya que, estos pueden estar a una altura u otra y dependiendo de eso, se verá si se trata de un movimiento hacia arriba o hacia abajo. En el caso de los movimientos horizontales se trata de ver qué marcadores se pueden ver en la imagen, dependiendo de estos se tratará de un movimiento hacia la derecha o hacia la izquierda.

Otro de los usos que se le ha dado a los sistemas desarrollados con la cámara Kinect es el de reconocimiento de los movimientos de la mano, como podemos ver en “*New Techniques for Hand Pose Estimation Based on Kinect Depth Data*” (Simon Hummel, Victor Häfner, Polina Häfner y Jivka Ovtcharova 2014), en este estudio podemos comprobar como el uso de la cámara Kinect es muy beneficioso para el filtrado de la imagen, ya que, permite utilizar la cámara de profundidad para hacer una separación entre la parte de la imagen donde se encuentra el objetivo, en este caso la mano, y la parte que no nos interesa, que es el fondo de la imagen.

En este estudio podemos ver que para el desarrollo de este tipo de sistemas se puede trabajar con el uso de marcadores en las manos o sin el uso de los mismos. Para el desarrollo del sistema que se describe en este estudio hacen uso del primero, con él, lo que hacen es fijar una serie de marcadores en la figura de la mano, lo que hace más fácil la identificación de la misma, hasta poder crear un esqueleto de la mano con los mismo marcadores, como se puede ver en la siguiente imagen.

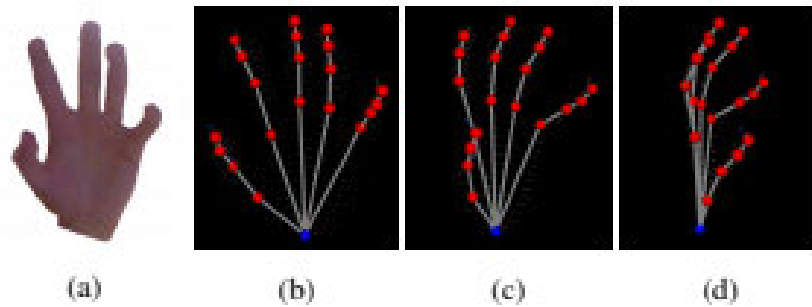


Imagen 6: Ejemplo de los marcadores utilizados para reconocer la mano

En la imagen anterior, tenemos el modelo de la mano y después los esqueletos que se pueden formar a partir de los marcadores y así poder reconocer la mano de una forma más sencilla. Por otro lado, para obtener datos más precisos haciendo uso de marcadores, requiere que se tengan que usar uno o más sensores (cámara Kinect).

Otro estudio en el que también se realiza el reconocimiento de la mano, con la diferencia de que en este caso no se utilizan marcadores para la mano, como en el anterior, es “3-D Hand Pose Estimation from Kinect’s Point Cloud Using Appearance Matching” (Pasquale Coscia, Francesco A.N. Palmieri, Francesco Castaldo and Alberto Cavallo 2016), este estudio presenta una nueva técnica, la cual se basa en la aproximación de la pose estimada de la mano, haciendo uso de nubes de puntos junto con el sensor Kinect.

Para poder llevar a cabo la estimación, lo primero que se realiza es el aislamiento de la mano en la imagen, para así, poder hacer una mejor aproximación de su pose. Esta aproximación pertenece a la clase de estimación de la pose parcial, esto hace que se tome la pose de la mano del frame actual, la cual, se utiliza como inicialización para el siguiente frame. Para la estimación de la postura se aplica una versión modificada del algoritmo Iterative Closest Point (ICP) (Punto más cercano iterativo). En la imagen siguiente se puede ver un esquema de cómo se lleva a cabo el reconocimiento de la pose de la mano.

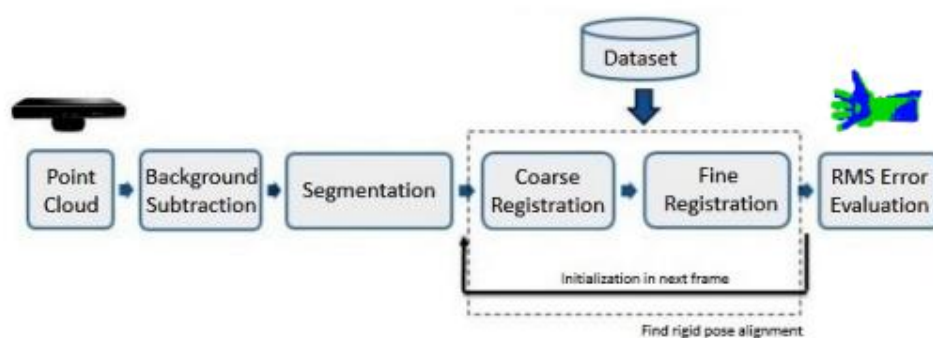


Imagen 7: Esquema del sistema de reconocimiento de la pose de la mano (Pasquale Coscia, Francesco A.N. Palmieri, Francesco Castaldo and Alberto Cavallo 2016)

El método propuesto en este estudio, lo que hace es trabajar con la nube de puntos que le proporciona la cámara Kinect sin más información, es decir, no se obtienen valores RGB de la nube de puntos que proporciona la cámara. Esto hace que este método pueda aplicarse, a su vez, a otros tipos de sensores de profundidad, o cámaras RGB-D.

Además, utilizan el principio de la luz estructurada; la idea general de este principio es proyectar un patrón conocido dentro de una escena e inferir la profundidad de los objetos utilizando la deformación de dicho patrón y un método de triangulación.

A continuación, veremos un ejemplo del filtrado de datos que realizan con las imágenes que obtienen.

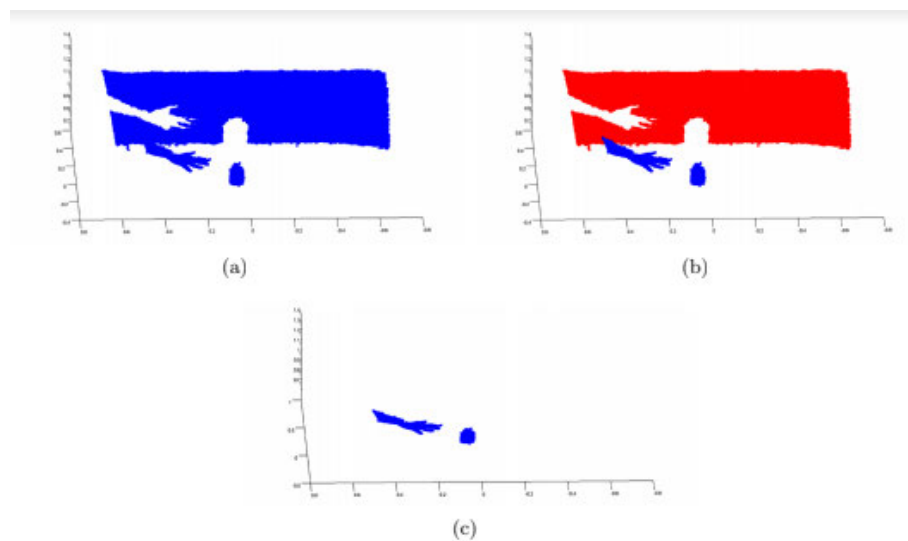


Imagen 8: Ejemplo del filtrado de la imagen (Pasquale Coscia, Francesco A.N. Palmieri, Francesco Castaldo and Alberto Cavallo 2016)

En la imagen anterior podemos ver que la parte a) se corresponde con la imagen tomada por la cámara Kinect, después, en la parte b), podemos ver cómo se diferencian los datos que interesan de los que no, en este caso, los datos más interesantes son los que se corresponden al color azul. Por último, en la parte c) es la imagen filtrada sin el fondo, el cual, no interesa.

La eliminación del fondo la hacen para poder quedarse con una nube de puntos mucho menor de la que tenían en un principio y así poder gestionar los datos de una manera más rápida y fácil en los siguientes pasos. Los cuáles serían la segmentación y después la generación de un modelo a reconocer.

Así que, por lo que se ha podido ver, para el estudio del comportamiento se pueden utilizar varios tipos de técnicas, aunque, una de las que más se está utilizando en la actualidad es con el uso de las cámaras Kinect. Con la imagen de profundidad, que nos proporciona la Kinect, el filtrado de la imagen se puede hacer mucho mejor, ya que, nos permite eliminar mucho ruido de la imagen, el cual puede provenir sobre todo del fondo de la misma. Con ello se puede hacer una recogida de datos más exacta y unos estudios del comportamiento más precisos.

3. Métodos de filtrado de datos y clasificación del movimiento

En este apartado lo que se realizará es una explicación teórica de los distintos métodos utilizados para la creación del sistema de reconocimiento.

Primero se explicará la parte teórica correspondiente al filtro de Kalman, después los Modelos Ocultos de Markov (MMH).

3.1. Filtro de Kalman

Para explicar el filtro de Kalman nos basaremos en la definición que dieron Greg Welch y Gary Bishop, es decir, “*El filtro Kalman es un conjunto de ecuaciones matemáticas que proporciona un medio computacional (recursivo) eficiente para estimar el estado de un proceso, de manera que minimiza la media del error cuadrático*”, (Greg Welch and Gary Bishop 2006), dicho de otra forma, el filtro de Kalman es un método recursivo que nos ayuda a estimar el estado de un proceso.

En 1960 R.E. Kalman publicó un documento en el que se describe una solución al problema del filtrado lineal de datos discretos. Desde entonces este filtro ha tenido muchas aplicaciones. Este método se basa principalmente en hacer estimaciones de los estados pasados, los estados actuales y en algunos casos hasta los estados futuros.

Para ello, Kalman aborda el problema de la estimación del estado, x_k , en el que $x \in \mathbb{R}^n$. En un proceso controlado en un tiempo determinado, k , mediante la siguiente ecuación lineal:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Donde la variable A es la matriz que relaciona el estado previo con el actual, esta variable puede cambiar a lo largo del tiempo, pero se asume que es constante; B a su vez también es una matriz, pero en este caso, relaciona una entrada de control opcional u con el estado x . La variable w_k es una variable aleatoria que representa el ruido del proceso.

En el momento k se hace una medición z_k del estado real x_k , que se representa como:

$$z_k = Hx_k + v_k$$

Donde $z \in \mathbb{R}^m$ y v_k es una variable aleatoria que representa el ruido en la medición. Por otro lado, la variable H es la matriz que relaciona el estado, x_k ,

con la medida de z_k ; esta variable al igual que pasa con la matriz A la consideraremos constante, aunque en la práctica puede cambiar.

Tanto w_k , como v_k se consideran independientes y con una probabilidad normal, esto lo representaremos con las siguientes ecuaciones, siendo $p(w)$, la probabilidad de w_k y $p(v)$ la de v_k :

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R).$$

Donde Q es la covarianza del ruido del proceso y R es la covarianza del ruido de observación, ambas variables asumimos que son constantes, aunque podrán cambiar con cada paso o medición.

Teniendo en cuenta lo anterior, se puede deducir que, tiene que haber una forma concreta para poder definir el estado actual y el estado siguiente. Para ello, se define una estimación del estado a priori (estado actual), que se define como $\hat{x}_k^- \in R^n$, de esta forma denotaremos la estimación del estado a priori en el paso k , y para el estado a posteriori (estado siguiente) lo denotaremos como $\hat{x}_k \in R^n$, dada la medición de z_k . Una vez definidos los estados a priori y a posteriori, podemos definir a su vez, las estimaciones de error de ambos estados. La primera ecuación es para el estado a priori y la segunda es para el estado a posteriori.

$$e_k^- = x_k - \hat{x}_k^-;$$

$$e_k = x_k - \hat{x}_k.$$

Con esto lo que se puede definir, a su vez, es la covarianza de estimación a priori P_k^- y a posteriori P_k , como podemos ver en las siguientes ecuaciones, la primera para el estado a priori y a segunda para el estado a posteriori.

$$P_k^- = E[e_k^- e_k^{-T}],$$

$$P_k = E[e_k e_k^T].$$

Con estas últimas ecuaciones lo que se podrá hacer es relacionar ambos estados. Lo que se hará es, derivar las ecuaciones de filtrado de Kalman (ecuaciones de x_k y z_k) con el objetivo de relacionar ambos estados, es decir, llegar a una ecuación que calcule la estimación a posteriori como una combinación lineal de la estimación a priori y una diferencia ponderada entre medición real y la predicción de medición.

El resultado es la ecuación que se muestra a continuación:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

La diferencia ponderada es la diferencia $z_k - H\hat{x}_k^-$, también se llama innovación en la medición, o el residuo. Esta medida lo que hace es reflejar la discrepancia entre la medida predicha $H\hat{x}_k^-$ y la medida actual z_k . Si entre ambas dan un resultado igual a 0, quiere decir que ambas son completamente iguales.

Por otro lado, la variable K es una matriz que representa el factor de ganancia o la mezcla que minimiza el error de covarianza a posteriori.

Esta matriz (K) la podemos hallar con la siguiente ecuación:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

Esta ecuación la podemos hallar sustituyendo la ecuación de estimación de error, del estado a posteriori, e_k , en la ecuación que relaciona tanto el estado a posteriori como el estado a priori, es decir, la ecuación donde se haya \hat{x}_k . Después, lo que se haría es la derivada con respecto a K y fijar el resultado a 0. Y por último se despejaría K , obteniendo la ecuación anterior.

A continuación, veremos como se utilizan las ecuaciones anteriormente definidas para poder aplicar el filtro de Kalman. Lo primero que tenemos que saber es que Kalman separa las ecuaciones anteriores en dos etapas, la etapa de predicción y la etapa de corrección. Con esta separación, el grupo de ecuaciones que pertenecen a una u otra etapa se denominan ecuaciones de actualización de tiempo, las que se corresponden con la etapa de predicción, y las ecuaciones de actualización de medición, las que se corresponden con la etapa de corrección.

Las ecuaciones de actualización del tiempo son:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

Y las ecuaciones de actualización de la medición son:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

Estas ecuaciones se aplican en un ciclo cerrado, el cual empieza utilizando las ecuaciones de predicción y después las de corrección, y así, hasta terminar el proceso, van pasando de una etapa a otra.

A continuación, vamos a ver un ejemplo de aplicación del filtro de Kalman, se trata de una simulación que se hace fijando una serie de valores, para así, poder comprobar cómo es el funcionamiento del filtro.

Para empezar, lo que haremos será elegir aleatoriamente una constante escalar, que en este caso será el estado, por ejemplo, $x = -0.37727$. Después lo que se hace es simular 50 mediciones distintas, las cuales tienen errores distribuidos normalmente alrededor de cero con una desviación estándar de 0,1.

En esta simulación fijamos la varianza de la medición en $R = 0.01$. Esto se hace así, ya que, se considera que es la variación que puede tener de error de medición. En la siguiente figura podemos ver el resultado de la simulación realizada, con los valores que se ha dicho. El valor verdadero de la constante aleatoria x está representado por la línea sólida, el ruido está representado por las cruces y la estimación que hace el filtro viene representada por la línea curva.

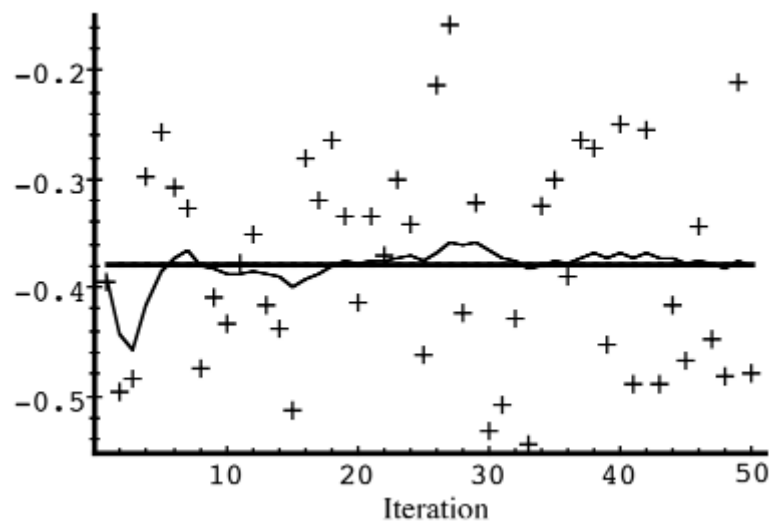


Imagen 9: Ejemplo de uso del filtro de Kalman (Greg Welch and Gary Bishop 2006)

Como se puede observar en la imagen anterior, el filtro en las primeras iteraciones no es muy exacto, pero con el paso de las iteraciones se va haciendo cada vez más exacto, lo que hace que se aproxime cada vez más a un valor real, sin tener en cuenta el ruido que pueda haber en cada iteración.

3.2. Modelos ocultos de Markov (HMM)

Primero, se realizará una explicación teórica del método para saber qué es lo que se aplica en este método y después se hará una pequeña explicación de cómo se aplicaría en un ejemplo práctico.

El modelo oculto de Markov (Hidden Markov Model en inglés), se trata de un modelo estadístico con el que se asume que, el sistema es un proceso de Markov del cual no se conocen de los parámetros. Lo que se pretende es hallar esos parámetros desconocidos a partir de una serie de parámetros que se conocen y son los que se recogen por el sistema. Estos parámetros que se recogen se utilizan para realizar análisis sucesivos, por ello se suelen emplear en aplicaciones de reconocimiento de patrones.

Teniendo esto en cuenta, podemos dar una definición más concreta. Con el modelo oculto de Markov lo que se obtiene son las probabilidades de secuencias de variables aleatorias, esto es lo que llamaremos estados ocultos, los cuales toman los valores de los parámetros que se conocen.

Con los modelos ocultos de Markov se pueden hacer varias suposiciones, una de ellas afirma que, si se quiere predecir el futuro en una secuencia de parámetros, todo lo que interesa es el estado actual, ya que, el estado actual es el que tiene un claro impacto sobre el estado futuro, no los estados anteriores.

Como ya se ha dicho, este modelo es muy útil cuando lo que se pretende es obtener la probabilidad de una secuencia de eventos observables, en la mayoría de las ocasiones este tipo de eventos se encuentran ocultos, es decir, no se pueden observar fácilmente. Con HMM podemos observar ambos tipos de eventos, los que podemos observar y los ocultos. Para ello se han definido los siguientes variables:

Nombre de la variable	Descripción
$Q = q_1 q_2 \dots q_N$	Es el conjunto de N estados observados
$A = a_{11} \dots a_{ij} \dots a_{NN}$	Es la matriz de probabilidad de transición A , cada a_{ij} representa la probabilidad de pasar del estado i al estado j , $\sum_{j=1}^N a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \dots o_T$	Es la secuencia de T observaciones, cada una extraída de un vocabulario, es decir, de la secuencia de los parámetros que se obtienen.
$B = b_i(o_t)$	Es la secuencia de probabilidades de observación, también llamadas probabilidades de emisión, cada una de las cuales expresa la probabilidad, b , de que se genere una observación o_t de un estado i .
$\pi = \pi_1, \pi_2, \dots, \pi_N$	Es la distribución inicial de probabilidad sobre los estados. π_i es la probabilidad de que la cadena Markov comience en el estado i . Algunos estados j pueden tener $\pi_j = 0$, lo que significa que no pueden ser estados iniciales.

Tabla 1: Descripción de las variables que se utilizan para hallar el modelo oculto de Markov (Bobick 1997)

Una vez sabemos esto, se puede hacer las siguientes suposiciones (Bobick 1997). La primera suposición, es en la que se halla la probabilidad, P , de que un estado en concreto, q_i , depende solamente del estado anterior, q_{i-1} . Esta suposición es la que se ha comentado anteriormente. Se representa de la siguiente manera:

$$P(q_i | q_1 \dots q_{i-1}) = P(q_i | q_{i-1})$$

En la segunda suposición, lo que se afirma es que la probabilidad, P , de que se produzca una observación o_i , depende solamente del estado que produce el estado q_i , no de ningún otro estado u observación. Se representa como:

$$P(o_i | q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i | q_i)$$

Para poder entender mejor la aplicación de este modelo, explicaremos dos aplicaciones que se pueden hacer en distintas situaciones; una de ellas estará más enfocada a la aplicación que se hace para el reconocimiento del movimiento de la mano.

Ejemplo 1: Estado del tiempo

En este ejemplo lo que se pretende analizar es el comportamiento que va a tener una persona dependiendo del tiempo que haga ese día, si se trata de un día nublado o con sol. (Hidden Markov model 2019)

Se ha fijado que la persona sólo realizará una de tres acciones, las cuales son: irse a caminar, hacer la compra o limpiar la casa.

Con esta información podemos definir una serie de variables que nos van a servir para realizar el modelo y la predicción de lo que podrá hacer la persona en un día determinado. Los estados que se van a definir son los del tiempo que puede hacer en el día, que son dos. Después se definirá una variable que represente las observaciones, es decir, las actividades que sabemos que puede realizar esa persona. Y, por último, se definirán las probabilidades iniciales de cada estado.

Con el estado y la probabilidad inicial de cada uno lo que podemos obtener es la matriz de transición entre cada estado. Esto lo se puede ver definido en la siguiente imagen.

```
estados = ('Lluvioso', 'Soleado')

observaciones = ('caminar', 'comprar', 'limpiar')

probabilidad_inicial = {'Lluvioso': 0.6, 'Soleado': 0.4}

probabilidad_transicion = {
    'Lluvioso' : {'Lluvioso': 0.7, 'Soleado': 0.3},
    'Soleado' : {'Lluvioso': 0.4, 'Soleado': 0.6},
}
```

Imagen 10: Definición de variables para el ejemplo 1

Por otro lado, relacionado con el modelo que hemos explicado antes, está la actividad que va a realizar esa persona dependiendo del día que haga. Esto se define con la probabilidad de emisión, como podemos ver en la siguiente imagen.

```
probabilidad_emision = {
    'Lluvioso' : {'caminar': 0.1, 'comprar': 0.4, 'limpiar': 0.5},
    'Soleado' : {'caminar': 0.6, 'comprar': 0.3, 'limpiar': 0.1},
}
```

Imagen 11: Resultados del ejemplo 1

Cómo podemos ver en la última imagen, si se trata de un día lluvioso hay un 50% de probabilidades de que esa persona esté limpiando, y un 40% de que esté limpiando. Por otro lado, si se trata de un día soleado hay un 60% de probabilidad de que lo que haga sea salir a caminar.

El grafo que representaría todo este ejemplo sería el siguiente:

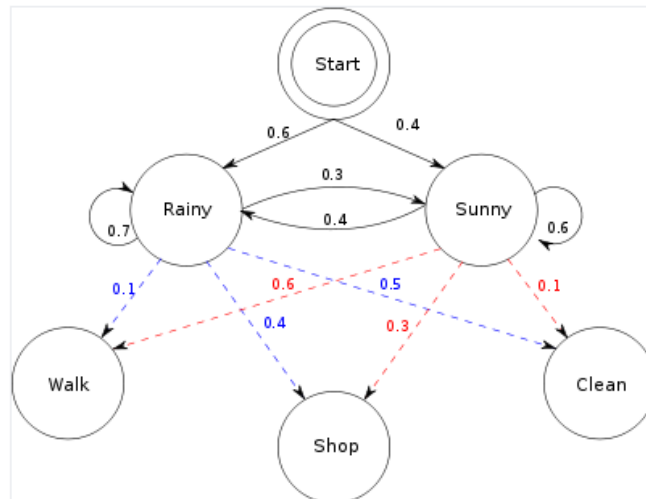


Ilustración 12: Resultado del ejemplo 1 (Hidden Markov model 2019)

Teniendo en cuenta todo lo que se ha explicado, ahora veremos cómo se puede aplicar al problema del reconocimiento del movimiento, concretamente, al reconocimiento del movimiento de la mano.

Basándonos en “*Movement, activity and action: the role of knowledge in the perception of motion*” (Bobick 1997), podemos ver que, para la detección del movimiento, se puede distinguir entre 3 conceptos de movimiento, dependiendo del que se pretenda reconocer se utilizará una técnica u otra. Estos tres conceptos son:

- Movimientos: se trata de estados que no requieren ningún conocimiento previo del contexto en el que se producen para ser reconocidos.
- Actividades: secuencias de movimientos o estados, donde el único conocimiento real requerido son las estadísticas de la secuencia.
- Acciones: son actividades a mayor escala, que normalmente, incluyen la interacción con el entorno y las relaciones causales

En el caso de este proyecto y para el ejemplo que se va a explicar, lo que utilizaremos es el segundo concepto, “actividad”. Para poder verlo más claramente nos ayudaremos del siguiente ejemplo.

Ejemplo 2: Reconocimiento del movimiento de saludo de la mano.

En este caso se trata de un movimiento de “actividad”, ya que, tiene una secuencia de puntos/estados por los que pasa, como se puede ver en la imagen 13. (Bobick 1997)

Con estos estados lo que hace es un modelo, con el cual se pueda reconocer los movimientos parecidos al que se han tomado. En la siguiente imagen podremos ver cómo sería el modelo que se ha hallado. Este modelo se representa mediante un grafo. En la imagen siguiente, a la derecha vemos representado el movimiento de la mano, cada uno de ellos se corresponde con un estado del grafo (izquierda de la imagen), el cual se identifica con un momento determinado de la imagen.

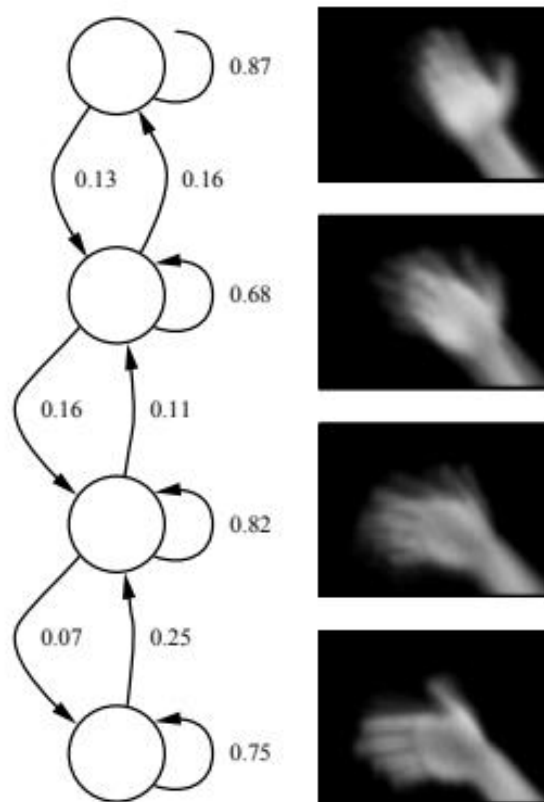


Imagen 13: Resultados del modelo del ejemplo 2 (Bobick 1997)

Uno de los puntos débiles que tiene utilizar este método es que no se pueden elegir concretamente cuales son las características más significativas para poder realizar el modelo, dado que este se hace a partir de los datos recogidos.

4. Detección y reconocimiento del movimiento

4.1. Kinect

Para llevar a cabo el desarrollo de este sistema se ha hecho uso de la cámara Kinect, la cuál nos proporciona dos tipos de imágenes, como se puede ver a continuación:

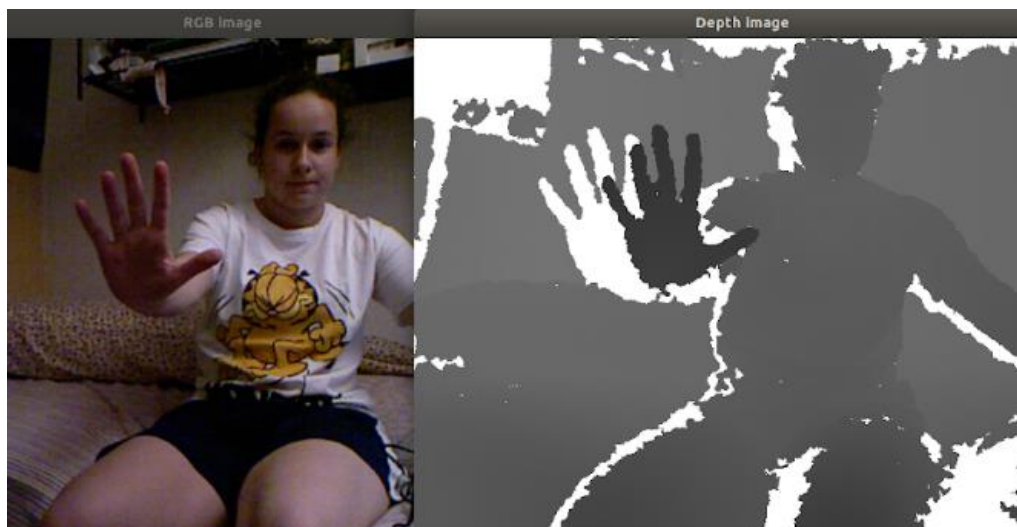


Imagen 14: Imágenes que se obtienen de la cámara Kinect

De la cámara Kinect, como ya hemos visto se pueden obtener dos tipos de imágenes, una imagen RGB y una imagen de profundidad. En este trabajo se ha trabajado con la imagen de profundidad, ya que, aporta más datos. Esto se puede ver muy bien con la imagen anterior, si nos fijamos en la imagen de profundidad, podemos ver que tiene distintos tonos de grises, esto se debe a que los objetos se encuentran a una distancia diferente, lo que permite delimitar la zona donde se realiza el movimiento de una forma más sencilla, haciendo que el procedimiento de filtrado sea más sencillo a su vez.

Para el desarrollo de este sistema se ha seguido el siguiente diagrama, en él se puede ver que, se capta la imagen de profundidad de la cámara Kinect y después se hace una segmentación de la imagen. A continuación, se realiza el filtrado de los datos que se obtienen, mediante Kalman, y se aplica HMM para generar los modelos, los cuáles van a servir para poder reconocer el movimiento.



Imagen 15: Diagrama del sistema desarrollado

Para la segmentación de la imagen lo que se ha realizado es la binarización de la misma, es decir, lo que se ha hecho es, eliminar todos los datos que no se consideran interesantes, en este caso se trata de todo aquello que no sea la mano. Para ello lo que se ha hecho uso de un método de binarización, con el que se puede delimitar la zona dónde se iba a realizar el movimiento de la mano, ya que, con la imagen de profundidad se pueden sacar los datos de a qué distancia se encuentran los objetos, como se ha podido ver anteriormente. Con esto lo que se obtiene es una imagen en la que, lo que aparece en blanco es lo que más interesa, mientras que lo negro es lo que no tiene importancia. Una vez hecho esto, el resultado obtenido se puede ver en la siguiente imagen:



Imagen 16: Ejemplo de binarización de la imagen

Cuando ya se tiene la imagen binarizada lo que se necesita es tomar unos datos de referencia, los cuáles son los que nos van a servir para realizar el reconocimiento del movimiento. Para esto lo que se pretende tomar es una serie de puntos, unas coordenadas (x,y) que se puedan obtener de la imagen.

Para la obtención de esos puntos lo primero que se ha hecho es aplicar a la imagen binaria, anteriormente hallada, el método de los contornos para que aplique este método sólo a los datos obtenidos en la propia imagen. Una vez se han hallado los contornos de la imagen, se aplica el método del centroide a los datos obtenidos por el método de los contornos, para así poder sacar un punto de referencia, como se muestra en la imagen siguiente.

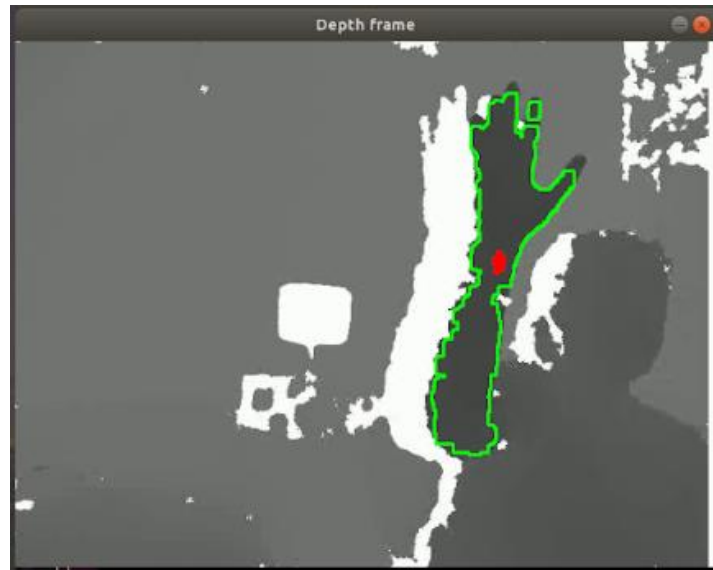


Imagen 17: Ejemplo de imagen aplican los métodos del centroide y contornos

Una vez realizado todo este proceso, lo que se realiza a continuación es el filtrado de datos y después se aplica el método de reconocimiento, esto lo veremos a continuación.

4.2. Filtrado de datos mediante filtro de Kalman

Para la aplicación del filtro de Kalman, se ha hecho uso de la librería de OpenCV, la cual permite hacer uso de una serie de métodos con los que se puede realizar los cálculos para el filtrado de los datos. (OpenCV, cv::KalmanFilter Class Reference 2019)

Para hacer este tipo de filtrado se ha tomado como referencia dos trabajos (Greg Welch and Gary Bishop 2006) y (OpenCV, Real Time pose estimation of a textured object 2019), con ambos trabajos se puede ver como se aplica el filtro de Kalman.

Lo primero de todo es decidir qué tipo de datos son los que se quieren filtrar. En este caso, se ha tomado como referencia el centroide del contorno de la mano.

Este nos da dos referencias, la coordenadas x,y , estas dos variables se asocian al número de mediciones que se van a utilizar en el método que se va a aplicar con OpenCV.

Lo que se lleva a cabo a continuación, es la declaración de una serie de variables, que son constantes y sirven para declarar la matriz A . Los parámetros a declarar son, el número de estados que va a tener, estos estados fijan la dimensión que tendrá la matriz A , y por otro lado, la frecuencia de fotogramas que se ha fijado para la captura de vídeo, porque ciertos elementos de la matriz A dependen de este valor.

Después se fijará el valor de la matriz de medición, la matriz H . Como ya sabemos, esta depende del estado y de la medición que se realiza en el mismo estado. Y, por último, lo que se hará es fijar los valores de las variables Q , covarianza del ruido del proceso, que depende del número de estados variables que se han declarado, y R , la covarianza del ruido de observación, que representa el número de mediciones que se han fijado.

Una vez ya se han fijado todas las variables que son constantes, lo que se realiza es el ciclo explicado anteriormente, en el apartado de teoría. Primero aplicaremos las ecuaciones de predicción y después las de corrección. Hay que tener en cuenta que, los valores que se refieren al estado a priori, \hat{x}_k^- , es el valor que se recibe del centroide. En este caso se trata de hacerlo con dos valores, coordenadas x,y , por lo que se utiliza un vector, en el cual se va guardando en diferentes elementos el valor, tanto de x , como el valor de y .

4.3. Obtención del modelo a reconocer con HMM

Lo primero que se tuvo en cuenta para la aplicación de HMM, es el tipo de movimientos que se iban a reconocer, porque como se ha podido ver en (Bobick 1997), dependiendo de la clasificación que le demos hay varias técnicas que se pueden utilizar. En este caso se eligió HMM, dado que lo que se tiene es una serie de puntos, los cuales se pueden clasificar como estados por los que pasa la mano durante la realización del movimiento.

Con el método de reconocimiento escogido, lo que se hace después es, elegir los movimientos de la mano que se van a reconocer, en este caso, se trata de cuatro movimientos, movimiento hacia la derecha, movimiento hacia la izquierda, movimiento hacia arriba y movimiento hacia abajo.

Para poder aplicar el método de HMM, se ha utilizado de la librería `hmmlearn`, (developers 2010), con esta librería se han fijado los valores iniciales para poder realizar el modelo de cada uno de los movimientos. Como se van a reconocer cuatro movimientos, se crearán cuatro modelos, uno por cada movimiento. Además, hay que tener en cuenta cuáles son los datos que se van a utilizar para la creación de cada uno de los modelos, en este caso serán los datos que se obtengan después de aplicar el filtro de Kalman.

Una vez que se tienen esos datos, hay que adaptarlos a los parámetros que se utilizan en la librería `hmmlearn`. Para el uso de esta librería se necesita saber el número de estados que va a tener cada modelo y el tipo de covarianza que se va a utilizar, para ello se ha cogido la referencia del trabajo de (Bobick 1997). En este estudio como se ha podido ver en el reconocimiento de ciertos movimientos, el modelo que se ha representado tenía cuatro estados, y los estados que se han elegido para inicializar el método han sido también cuatro.

Para elegir el tipo de covarianza se han tenido en cuenta dos tipos, ‘diag’ y ‘full’, con la primera opción lo que se hace es fijar la misma matriz de covarianza diagonal para todos los estados, en el caso de ‘full’, cada estado tendrá su propia matriz de covarianza. Comparando ambos tipos, finalmente se ha trabajado con el tipo de covarianza ‘diag’, dado que con este tipo se ha visto que los resultados eran mejores.

Por otro lado, hay que elegir cómo se van a tener en cuenta los datos, es decir, cómo se van a recoger los datos para hacer el modelo y después comparar los datos de nuevos movimientos con el modelo. En este caso también se han considerado dos opciones, la primera depende de la velocidad del movimiento y del frame-rate (fotogramas/s) que se haya fijado, este método se rechazó ya que dependía de que la mano pasase por ciertos puntos, fijados previamente, en un tiempo determinado y eso en algunas ocasiones sí que podría funcionar, pero en otras los datos podrían ser confusos. El otro método, el cual es el que se utiliza, se basa en dividir la imagen en un número determinado de áreas o clústers, por los que deberá pasar la mano; la gran ventaja que tiene esta opción es que no depende del tiempo.

Para realizar la separación de áreas/clústers se ha tenido en cuenta el número de pixels que nos da la imagen, de largo y ancho (imagen 480x640). El número de clústers en los que se ha dividido la imagen, tanto para el ancho, como para el largo son los mismos, ocho. En las tablas siguientes podremos la separación en clústers en cada caso, para movimientos horizontales primero y en la siguiente para movimientos verticales.

Número de clúster	1	2	3	4	5	6	7	8
Rango de pixeles	0-80	80-160	160-240	240-320	320-400	400-480	480-560	560-640

Tabla 2: Separación de clústers para los movimientos horizontales

Número de clúster	1	2	3	4	5	6	7	8
Rango de pixeles	0-60	60-120	120-180	180-240	240-300	300-360	360-420	420-480

Tabla 3: Separación de clústers para los movimientos verticales

Una vez se tienen los clústers fijados, lo que se hace es tomar los datos de cada uno de los cuatro vídeos de cada movimiento que van a servir como modelo. Con estos datos lo que se hace es clasificar cada uno de los puntos que se recogen en el clúster correspondiente; después se hace la media en cada clúster, de los puntos que se han recogido. Después se crea una matriz de los datos entrada, dado que se trata de dos datos, las coordenadas x,y. Y con esto, lo que se hallan son los estados ocultos y se obtiene una puntuación. Esta puntuación es la que se utilizará como referencia para reconocer el tipo de movimiento que se ha realizado.

Aunque para el reconocimiento de los movimientos inicialmente se pensó en hacerlo por separado, por un lado, los movimientos verticales y por el otro, los horizontales, debido a que en la imagen el número de pixels es diferente en el ancho y largo, se acabó desechando esa opción, dado que no parecía que fuera influir en gran medida en el reconocimiento. Y como se ha podido comprobar después, el que haya una diferencia de pixels no tiene ninguna influencia a la hora de reconocer el movimiento, por lo que, se reconocen los cuatro movimientos juntos, aunque para la clasificación de clústers se haga de forma distinta.

Un ejemplo de cómo se hace reconocimiento es el que podemos ver en la siguiente tabla, como ya se ha dicho, lo que se hace es comparar las puntuaciones que se obtienen de los modelos con la puntuación que se ha obtenido del movimiento que se quiere clasificar.

Movimiento hacia la derecha	Estados ocultos = [1 1 1 3 2 2 2]
	Puntuación = -412.0796
Movimiento hacia la izquierda	Estados ocultos = [3 3 3 3 3 3 3]
	Puntuación = -409.1553
Movimiento hacia abajo	Estados ocultos = [1 0 0 3 3 3 2 2]
	Puntuación = -68.8970
Movimiento hacia arriba	Estados ocultos = [2 0 3 3 3 3 3 3]
	Puntuación = -110.1134

Tabla 4: Ejemplo de valores obtenidos en el reconocimiento del movimiento

En la tabla anterior se pueden ver los valores que se obtienen como resultado de reconocimiento del movimiento; como se puede observar nos da un resultado por cada movimiento. Los valores que se representan son los estados ocultos por lo que pasa la muestra de valores que se le ha pasado anteriormente y una puntuación, con esta puntuación se realiza una comparación entre las cuatro puntuaciones que se han obtenido y la más alta será la que se corresponda con el movimiento que se ha producido. Aunque, como veremos más adelante, en algunos casos no es así.

En el ejemplo que se muestra anteriormente (tabla 4), se elegiría el movimiento hacia abajo dado que es el que tiene una puntuación más alta.

5. Resultados

A lo largo de este apartado se realizará una explicación sobre los resultados que se han obtenido después de haber aplicado las técnicas anteriormente explicadas. En el apartado de resultados obtenidos se hará una explicación sobre qué es lo que se ha conseguido con este sistema y después un análisis de los mismos.

5.1. Resultados obtenidos

Primero vamos a ver cuáles son los resultados que se han podido obtener utilizando el filtro de Kalman.

Los primeros resultados obtenidos son los que se pueden ver en la siguiente imagen.

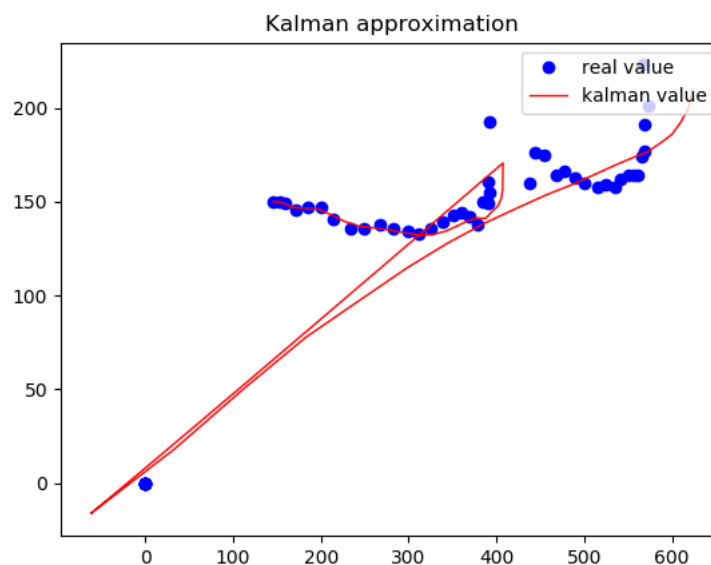


Imagen 18: Ejemplo de primera aproximación de Kalman

En la imagen anterior, se puede ver que, el valor real se corresponde con los puntos azules, que representan el valor que se obtenido del centroide, y el valor que se ha conseguido al aplicar Kalman es el que se corresponde con la línea roja.

Como se puede apreciar en la imagen los valores que se obtienen aplicando el filtro de Kalman son bastante buenos, haciendo que dichos valores no varíen tanto como lo hacen los valores reales. Lo malo, es que, como se puede ver, hay un momento en la imagen que el centroide es (0,0), esto provoca que con el filtro haya un pico, el cual no tienen ningún sentido al tratarse de un movimiento continuo. Esto se debe a que cuando el área del contorno que se halla es muy

pequeña, lo que hace que no se calcule el centroide, y por defecto, en estos casos pasa a ser (0,0); esto provoca que cuando se pase a aplicar el filtro haya incongruencias con los valores.

Para solucionar este problema lo que se ha hecho es tomar el punto anterior como referencia para así lograr tomar unos valores más coherentes y que el filtro funcione mejor.

Una vez se han aplicado los cambios anteriores, podemos ver que el resultado es mucho mejor, lo que hace que estos valores, que son los que se van a utilizar para hallar el modelo, sean mucho más exactos que antes. En la siguiente imagen podemos ver el mismo ejemplo de antes, con la diferencia de que se ha aplicado la mejora al filtro de Kalman.

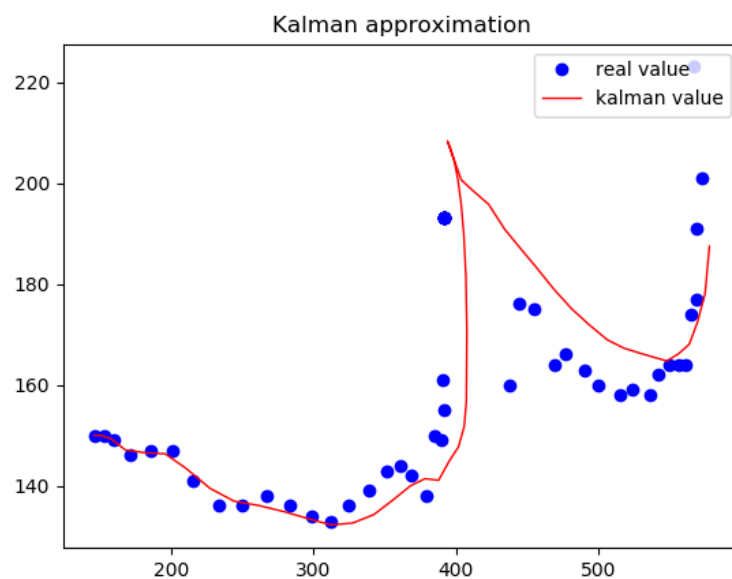


Imagen 19 Ejemplo del filtrado de Kalman más preciso

A continuación, vamos a ver qué resultados se han obtenido al aplicar el método HMM. Para comprobar que el funcionamiento del sistema y ver que realmente reconoce los movimientos se han realizado varios vídeos de prueba reproduciendo el mismo movimiento. En cada caso lo que se ha hecho es una serie de 6 vídeos, cada vídeo que se ha generado se ha intentado realizar bajo las mismas condiciones, misma distancia y posición de la cámara y sobre todo bajo las mismas condiciones de luz.

Para cada movimiento se comentará el resultado, ya que los resultados obtenidos para cada caso han sido diferentes.

Reconocimiento de un movimiento hacia la derecha:

En este caso el resultado ha sido bastante buenos, ya que se ha podido conseguir que con el modelo escogido se reconozcan todas las pruebas del movimiento que se han generado.

Reconocimiento de un movimiento hacia la izquierda:

Para el reconocimiento del movimiento hacia la izquierda se han tenido más problemas, ya que, solo se han podido reconocer correctamente 4 de los 6 vídeos de prueba. Si lo comparamos con el resto de resultados, este es bastante pobre. Más adelante se cometerá porqué se ha producido este resultado y cómo se podría mejorar.

Reconocimiento de un movimiento hacia abajo:

Para este tipo de movimiento, ha sido bastante bueno el reconocimiento que se ha realizado, ya que solo se ha fallado en uno de los vídeos a reconocer.

Reconocimiento de un movimiento hacia arriba:

En este caso, ocurre lo mismo que en el caso del reconocimiento hacia la derecha. El modelo que se ha escogido ha sido el mejor posible, dado que se ha comprobado que el acierto en el reconocimiento es del 100%.

5.2. Análisis de los resultados

Para el análisis de los resultados primero realizaremos análisis del resultado obtenido y después un breve estudio con una matriz de confusión para comprobar lo eficaz que puede ser el sistema que se ha creado.

- Movimiento hacia la derecha:

En la siguiente tabla lo que vamos a poder observar son los resultados de las puntuaciones que se han obtenido al realizar un reconocimiento de un movimiento hacia la derecha. Para este movimiento como se han reconocido todos los movimientos, se comentará el resultado en uno de los seis experimentos/pruebas, se trata del sexto vídeo.

Movimiento hacia la derecha	Puntuación = -203.1197
Movimiento hacia la izquierda	Puntuación = -330.6671
Movimiento hacia abajo	Puntuación = -701.0816
Movimiento hacia arriba	Puntuación = -389.7568

Tabla 5: Prueba de reconocimiento de movimiento hacia la derecha

Como se puede ver en la tabla, de los valores que se obtienen, el más alto es el que se corresponde con el movimiento hacia la derecha. Un dato importante y a tener en cuenta es, que de los valores que se obtienen, el que más se acerca al que se ha obtenido para el movimiento hacia la derecha es el del movimiento hacia la izquierda, pero se diferencian en más de 100 puntos. En algunos casos esta diferencia podrá provocar que no se reconozca correctamente el movimiento, como veremos a continuación.

- Movimiento hacia la izquierda:

En este caso se ha obtenido un reconocimiento de 4/6, esto se debe a que el modelo escogido, aunque, por puntuaciones obtenidas de los vídeos de movimiento hacia la izquierda sea el mejor de los modelos posibles, se ha visto que se confunde bastante con otro modelo escogido, en este caso el escogido para el movimiento hacia arriba.

En la siguiente tabla veremos un ejemplo del reconocimiento erróneo.

Movimiento hacia la derecha	Puntuación = -462.8364
Movimiento hacia la izquierda	Puntuación = -376.1313
Movimiento hacia abajo	Puntuación = -648.8625
Movimiento hacia arriba	Puntuación = -356.1368

Tabla 6: Prueba de vídeo 1 de movimiento hacia la izquierda, reconocimiento erróneo

Si nos fijamos en la puntuación obtenida por el movimiento hacia arriba y la obtenida por el movimiento hacia la izquierda, las puntuaciones son bastante parecidas, además de que son las dos más altas. Como se puede observar son puntuaciones bastante cercanas ya que no se diferencian en más de 100, lo que lleva a pensar que, en este caso, podría tratarse por lo menos de alguno de los dos movimientos. Aunque, la puntuación más alta ha sido la de movimiento hacia arriba, por lo que se ha clasificado erróneamente.

- Movimiento hacia arriba:

A continuación, veremos un ejemplo de reconocimiento para el movimiento hacia arriba.

Movimiento hacia la derecha	Puntuación = -528.9731
Movimiento hacia la izquierda	Puntuación = -570.0037
Movimiento hacia abajo	Puntuación = -13565252.41
Movimiento hacia arriba	Puntuación = -203.0801

Tabla 7: Prueba de reconocimiento de movimiento hacia arriba, vídeo 1

Como se puede apreciar en la imagen, en este caso el sistema reconoce correctamente el movimiento hacia arriba, si nos fijamos en la diferencia de puntuaciones que se obtiene en el movimiento hacia arriba, con respecto, al resto podemos ver como la diferencia es bastante apreciable. Por tanto, se puede afirmar que este movimiento es realmente un movimiento hacia arriba, comparado con el modelo elegido.

Para realizar un estudio más exacto del grado de acierto del sistema que se ha desarrollado se utilizará un matriz de confusión. En otras palabras, lo que haremos será comprobar el rendimiento del algoritmo de reconocimiento que se ha desarrollado.

Con la matriz de confusión podemos relacionar el número de aciertos y fallos en cada caso, es decir, el número de verdaderos positivos y el número de falsos positivos. (Markham 2014)

A continuación, se muestra una tabla con lo que los resultados obtenidos.

	Valor predicho				
		Decha	Izqda	Arriba	Abajo
Valor real	Decha	6	0	0	0
	Izqda	0	4	2	0
	Arriba	0	0	6	0
	Abajo	0	0	1	5

Tabla 8: Matriz de confusión

Como se puede observar en la matriz de confusión anterior, los datos que se corresponden a la diagonal son las predicciones correctas que ha hecho el sistema, en la mayoría de los casos no se trata de pocos aciertos, en el 50% de ellos se trata de un acierto del 100%. Los valores que aparecen fuera de la diagonal principal son los datos que se han predicho de forma errónea, estos son bastante fáciles de identificar. Como se puede ver el sistema no es capaz de distinguir correctamente entre movimiento hacia arriba y los movimientos hacia la derecha y movimiento hacia abajo.

Ahora haremos un análisis más en profundidad. Para ello, lo que haremos será un análisis predictivo, el cual se hace con otra matriz de confusión, en esta matriz tenemos dos filas y dos columnas en las que podemos ver el número de falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Como se define en la siguiente tabla.

	Valor predicho		
		Decha	Izqda
Valor real	Decha	Verdadero positivo (TP)	Falso negativo (FN)
	Izqda	Falso positivo (FP)	Verdadero negativo (TN)

Tabla 9: Matriz de confusión para cada caso

A continuación, lo que haremos es un estudio para cada uno de los cuatro casos en los que vamos a sacar la precisión, en cada uno de los casos, es decir, cuando se predice el movimiento, con qué frecuencia es correcto. Para ello se tendrá en cuenta la siguiente fórmula, donde ACC representa la precisión:

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

Par el movimiento hacia la derecha se ha hallado la siguiente tabla.

Mov. Derecha	Valor predicho		
		Decha	Resto
Valor real	Decha	6	0
	Resto	0	18

Tabla 10: Precisión Movimiento hacia la derecha

Como podemos ver en este caso, la precisión del sistema es del 100%. En este caso la precisión para los movimientos a la derecha es la mejor que se puede obtener, por lo que cada vez que se predice un movimiento a la derecha, la predicción se considerará correcta.

En la siguiente tabla veremos los resultados del movimiento hacia la izquierda.

Mov. Izquierda	Valor predicho		
Valor real		Izqda	Resto
	Izqda	4	0
	Resto	2	16

Tabla 11: Precisión Movimiento hacia la izquierda

En este caso la precisión de acierto para este modelo es de 83%, como se puede ver en este caso la precisión baja considerablemente, esto se debe a que no se han reconocido correctamente 2 casos.

Para el movimiento hacia abajo se obtiene la siguiente matriz:

Mov. Abajo	Valor predicho		
Valor real		Abajo	Resto
	Abajo	5	0
	Resto	1	17

Tabla 12: Precisión Movimiento hacia arriba

En este caso la precisión es del 91,67%, es una precisión bastante buena, aunque como ya se ha visto en el caso anterior al no haber reconocido correctamente uno de los casos penaliza un poco, aunque se puede considerar un buen reconocedor.

Y por último veremos la matriz que se obtiene para el movimiento hacia arriba

Mov. Arriba	Valor predicho		
Valor real		Arriba	Resto
	Arriba	6	0
	Resto	0	18

Tabla 13: Precisión Movimiento hacia abajo

En este caso ocurre lo mismo que en el caso del reconocimiento del movimiento hacia la izquierda, se tiene una precisión del 100%. Por lo que en este caso el modelo elegido es el correcto.

Por tanto, como se puede observar, la precisión de cada uno de los movimientos no baja del 80%, por lo que con estos datos el reconocedor que se ha creado se puede considerar un buen reconocedor.

6. Aspectos sociales, ambientales, éticos y legales

Como se ha comentado a lo largo de este documento la información que se recabe puede ser útil a la hora de estudiar el comportamiento de las personas. Lo que se pretende es que para realizar esos estudios no haga falta llevar muchos dispositivos encima y que se pueda realizar con una simple cámara. Esto permite que no se utilicen muchos dispositivos y que incluso se puedan utilizar dispositivos pequeños para que el entorno en el que se encuentra la persona no se vea afectado por el uso de estos dispositivos.

Por otro lado, hay que tener en cuenta que la información puede ser sensible, dado que al ser una cámara si no se ha dado el consentimiento por parte de la persona, en lo que se refiere a la protección de datos, en este caso la identidad de la persona, no se estaría aplicando, por ello cuando se vaya a realizar habrá que contar con el consentimiento de la persona para que se puedan utilizar los datos recogidos. Aunque para este tipo de sistemas siempre será necesario.

7. Conclusiones

Los sistemas de captura del movimiento comienzan a tener cada vez más importancia, esto se debe a que cada vez más empresas quieren invertir en estos. Es algo muy beneficioso para los estudios que se puedan realizar sobre el comportamiento, ya que, cuanto más precisos sean los sistemas que se desarrollen, los datos, a su vez, también serán más exactos, lo que hará que los estudios que se realicen de algunos comportamientos sean más completos.

En este sistema, con el uso del filtro de Kalman para el filtrado de datos se ha podido ver que da un gran resultado y que los datos que se obtienen después de aplicarlo son con mucho menos ruido del que se recibía. Además, ha sido muy útil para después poder aplicar el método HMM y realizar un modelo de los movimientos que se han reconocido.

Con el uso de ambos métodos, Kalman y HMM, se ha podido desarrollar un sistema bastante fiable. Aunque, se podría mejorar la precisión generando modelos mejores, ya que, como se ha podido comprobar, uno de los movimientos tiene un índice de acierto un poco más bajo con respecto al resto de los movimientos.

A pesar de esto, el sistema desarrollado ha podido reconocer los cuatro movimientos con bastante precisión, lo que lleva a pensar que los resultados son bastante buenos y que con un poco de mejora se podría llegar a una precisión mayor.

8. Futuros trabajos

Desarrollar un sistema que sea más exacto cuando reconozca los cuatro movimientos, se basaría en realizar un modelo con más muestras de cada movimiento, para así, poder hacer modelos más completos para cada movimiento.

Por otro lado, lo que se podría realizar es una ampliación de los movimientos a reconocer e incluso realizar un sistema más completo, el cual pudiera reconocer, tanto movimientos como poses estáticas de la mano.

Otro estudio que se podría realizar es la implementación de este sistema, pero con otro tipo de sensores, utilizando algún tipo de sensor que se pueda aplicar a la mano, de esta forma se podría realizar un estudio mucho más preciso.

9. Bibliografía

- (BSD), hmmlearn developers. 2010. *hmmlearn*. Último acceso: 2019. <https://hmmlearn.readthedocs.io>.
- Bobick, Aaron F. 1997. «Movement, activity and action: the role of knowledge in the perception of motion.» *MITMedia Laboratory, Cambridge*.
- Daniel Jurafsky & James H. Martin. 2018. «Hidden Markov Models.» *Speech and Language Processing*.
- David J. Anderson and Prieto Perona. 2014. «Toward a Science of Computational Ethology.» *Neuron*.
- Diana-Margarita Córdova-Esparza, Juan R. Terven, Hugo Jiménez-Hernández y Ana-Marcela Herrera-Navarro . 2017. «A multiple camera calibration and point cloud fusion tool for.» *El Sevier*.
- Eyal Itskovits, Amir Levine, Ehud Cohen and Alon Zaslaver. 2017. «A multi-animal tracker for studying.» *BMC Biology*.
- Greg Welch and Gary Bishop. 2006. «An Introduction to the Kalman Filter.» *University of North Carolina at Chapel Hill, Department of Computer Science*.
- s.f. *Hidden Markov model*. Wikipedia. https://en.wikipedia.org/wiki/Hidden_Markov_model.
- Hueihan Jhuang , Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D. Steele y Thomas Serre. 2010. «Automated home-cage behavioural phenotyping.» *Nature Communications*.
- Hui-Shyong Yeo, Byung-Gook Lee and Hyotaek Lim. 2013. «Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware.» *Springer Science+Business Media New York*.
- John Darby, María B. Sánchez, Penelope B. Butler and Ian D. Loram. 2016. «An evaluation of 3D head pose estimation using the Microsoft Kinect v2.» *El Sevier*.
- Markham, Kevin. 2014. *Simple guide to confusion matrix terminology*. Último acceso: 2019. <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>.
- OpenCV. 2019. *cv::KalmanFilter Class Reference*. Último acceso: 2019. https://docs.opencv.org/trunk/dd/d6a/classcv_1_1KalmanFilter.html.
- OpenCV. 2019. *Real Time pose estimation of a textured object*. OpenCV. Último acceso: 2019. https://docs.opencv.org/3.4/dc/d2c/tutorial_real_time_pose.html.
- Pasquale Coscia, Francesco A.N. Palmieri, Francesco Castaldo and Alberto Cavallo. 2016. «3-D Hand Pose Estimation from Kinect’s Point Cloud Using Appearance

Matching.» *Seconda Universit' a di Napoli (SUN), Dipartimento di Ingegneria Industriale e dell'Informazione, Aversa (CE) - Italy.*

Pierre Plantard, Edouard Auvinet, Anne-Sophie Le Pierres y Franck Multon. 2015. «Pose Estimation with a Kinect for Ergonomic Studies: Evaluation of the Accuracy Using a Virtual Mannequin.» *MDPI.*

Rokoko. 2017. <https://www.rokoko.com/en/>. Smartsuit Pro.

Simon Hummel, Victor Häfner, Polina Häfner y Jivka Ovtcharova. 2014. «New Techniques for Hand Pose Estimation Based on Kinect Depth Data.» *Conference and Exhibition of the European Association of Virtual and Augmented Reality.*