VNIVERSIDAD NACIONAL AVTÓNOMA DE MEXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

SISTEMA DETECTOR Y RECONOCEDOR DE GESTOS PARA INTERACCIÓN EN UN QUIRÓFANO EMPLEANDO CÁMARAS RGB-D

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA (COMPUTACIÓN)

P R E S E N T A:

GUSTAVO VERGARA GONZAGA

DIRECTOR DE LA TESIS: DR. FERNANDO ARÁMBULA COSÍO

MÉXICO, D.F. **2012.**

Jurado asignado

PRESIDENTE:	DR. JESÚS SAVAGE CARMONA
VOCAL:	DR. FERNANDO ARÁMBULA COSÍO
SECRETARIO:	DRA. MARÍA ELENA MARTÍNEZ PÉREZ
SUPLENTE:	DR. FERNANDO GAMBOA RODRÍGUEZ
SUPLENTE:	DRA. NIDIYARE HEVIA MONTIEL

Resumen

El reconocimiento de gestos (basados en la posición de las manos del usuario) representan un problema desafiante para la investigación. El presente trabajo muestra un clasificador que puede reconocer hasta cinco gestos (*zoom in, zoom out,* paneo, puntero y rotación) en base a la información que nos puede dar una cámara RGB-D, dichas gesticulaciones sirven para interactuar con un software de uso médico y así poder visualizar modelos anatómicos en un ambiente virtual. Se utilizaron un par de métodos de clasificación, se analizaron y se obtuvieron resultados favorables.

Abstract

Gesture recognition (based on the position of the user's hands) represents a challenging problem for research. This paper shows a classifier that can recognize up to five gestures (zoom in, zoom out, panning, cursor and rotation) based on the information an RGB-D camera can provide, these gestures are used to interact with medical software so it is possible to visualize anatomical models in a virtual environment. A couple of classification methods were used and analyzed on this research with favorable results.

Índice

Capítulo 1. Introducción

El trabajo que aquí se presenta está centrado en la detección y reconocimiento de gestos (en específico, los basados en la posición de las manos) a distancia con ayuda de una cámara RGB-D, las cuales han ido tomando importancia debido a los datos que nos pueden proveer.

El sistema desarrollado se aplicó en un prototipo de una interfaz gráfica para la visualización de modelos anatómicos tridimensionales.

Dada una posición de las manos, el sistema será capaz de interpretarla y traducirla en acciones equivalentes al movimiento del mouse. Los gestos que podemos lograr con una cámara que posee estas capacidades se pueden extender a los que se encuentran en 3D sin la necesidad de utilizar algún otro dispositivo para el rastreo del usuario.

El reconocimiento de gestos representa un problema retador para la visión computacional, en ocasiones por la complejidad de los ambientes en los que es comúnmente aplicado. Resolver tal problema requiere de mecanismos efectivos de rastreo, generación de descriptores y análisis de cúmulos.

1.1 Objetivos

El presente trabajo tiene como finalidad desarrollar un sistema detector y reconocedor de gestos mediante el seguimiento del esqueleto humano. Los objetivos generales son los siguientes:

- Mostrar y conocer el estado del arte referente a los sistemas multimedia con interacción hombre-computadora.
- Generar una imagen en profundidad y tener el seguimiento del esqueleto humano con la ayuda de la cámara RGB-D.
- Estudiar un método que nos ayude a reconocer gestos.
- Implementar la técnica seleccionada para el reconocimiento de patrones.

Dados los objetivos generales, los objetivos específicos son:

- Acceder a fuentes relacionadas con ésta investigación, para tenerlas en cuenta como partida de nuestro trabajo.
- Analizar las librerías que nos otorga el Kinect SDK para el control del dispositivo.
- Implementar un sistema detector de los gestos que se definirán y observar el comportamiento de los datos.
- Desarrollar una aplicación para interactuar con modelos anatómicos tridimensionales.

1.2 Motivación

El campo de la medicina se ha visto sensiblemente nutrido por los avances que la computación le ha aportado. Por ejemplo, el uso de sistemas de cómputo como parte del equipo dentro del quirófano se ha incrementado; ya que los sistemas computarizados pueden brindar valiosa información al operante antes, durante y después de una intervención.

Este tipo de sistemas incluye dispositivos y técnicas para proporcionar interfaces entre la realidad virtual de los modelos por computadora y los planos quirúrgicos y la realidad actual de los quirófanos, los pacientes y cirujanos (P. Cinquin *et al*., 1995; L. Joskowicz *et al*., 2001).

La presente investigación contribuye a los esfuerzos por la consecución de la preservación del campo estéril de una sala de intervenciones; ello a través de la creación de un sistema que permite al operante el acceso y la manipulación de una imagen médica, sin la necesidad de entrar en contacto con ningún dispositivo y los riesgos que ello podría representar en una sala de cirugía (Organización Mundial de la Salud, 2012).

Como un marco importante para esta investigación, es preciso resaltar que, adoptando las necesidades que los sistemas médicos puedan requerir, en el Laboratorio de análisis de imágenes y visualización en el Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM se están desarrollando diversos estudios para el apoyo a aplicaciones médicas que próximamente podríamos ver adaptadas en instituciones sanitarias del país.

Finalmente, y si bien el numen del estudio que aquí se presenta fue aquel cuya aplicación acontecería en el ámbito médico, es necesario aclarar que el sistema desarrollado puede encontrar campos de ejecución en otras áreas que precisen la manipulación de imágenes a distancia, como son las del entretenimiento, la automatización de procesos y la educación.

1.3 Cirugía asistida por computadora

La cirugía asistida por computadora (CAS, por sus siglas en inglés) forma – desde hace tiempo – parte de una gran integración de sistemas para cirugías mínimamente invasivas (F. Marquet *et al*., 2006). Para lograr esto, es necesario recuperar datos de la sala de operaciones con diferentes sensores, transformando así a la sala de operaciones en un ambiente cada vez más complejo.

El objetivo es entender la situación actual y adaptar automáticamente las funciones de asistencia; ser capaz de extraer automáticamente información de la sala de operaciones tales como sucesos, medidas, eventos adversos que incluso puedan permitir la gestión de sistemas sensibles al contexto (F. Lalys *et al*., 2012).

Añadiendo el análisis antes de una cirugía, se están adaptando tecnologías capaces de comunicarse a través de la red para el cuidado de la salud; las cuales han permitido que mejore la calidad de vida de las personas.

La investigación del cuidado de la salud, en condiciones fuera de un hospital, también ha provocado el interés de investigadores, así como la de los profesionales de la salud (I. Korhonen *et al*., 2003; M. Mahfouz *et al*., 2012).

Muchos de los avances – si se considera el valor de la importancia crítica de un sistema médico – necesitan procesos que se desempeñen con mayor precisión. Por ello, el desarrollo también se ha basado en simular los eventos posibles que puedan ocurrir dentro de un quirófano y cada una de las tareas que puedan tener sus intervencionistas. Así, un número considerable de los sistemas que se han desarrollando y se siguen investigando son aquellos que simulan condiciones de un ambiente quirúrgico, que sirven principalmente para entrenamiento (Y. Sun *et al*., 2011; J. Xu *et al*., 2011; A. Baumgart *et al*., 2007).

Capítulo 2. Marco teórico

2.1 Interacción hombre-máquina

La interacción puede ser definida como el intercambio de información entre dos o más participantes activos.

En términos de programación, un elemento en la interacción es un sistema computacional y el otro es el usuario. Como condición, tiene que existir la constante retroalimentación en sistemas que desean incluir la interacción entre el humano y la computadora. Sin la debida retroalimentación los sistemas no podrían autorregularse porque no sabrían lo que están haciendo.

Una característica más que distingue a este tipo de sistemas es que el usuario puede regular el flujo de información conforme al tiempo que este desee. Cuando se le da la posibilidad al usuario de completar una tarea o introducir información en el sistema que cambia de manera sustancial y darle un significado a los datos que pueda proveerle a este para que le responda al usuario, entonces se está creando interacción (J. Noble, 2009).

Todo tipo de interacción está dotada de cierto vocabulario que permite la comunicación entre los actores que en ella intervienen. Lo que es importante hacer aquí es que se entienda qué acciones pueden ejecutarse y también que sea posible advertir que el sistema entiende esas acciones en el mismo sentido, e interprete de la misma manera lo que hace el usuario. Algunos tipos de interacción son:

- La manipulación del mouse: este es el método más común de interacción con la computadora hasta el momento. Por tal motivo, gran parte de las aplicaciones de uso común han sido diseñadas utilizando este método.
- Presencia, localización e imagen: el uso de la presencia y ausencia del participante o el usuario es una forma extremadamente sencilla – pero intuitiva – de interactuar profundamente. Este puede ser detectado en peso, movimiento, luz, calor o, en ciertos casos, por sonido. La presencia del cuerpo – aunque simple – es una sólida base de la interacción, que involucra a

usuarios y pide a estos mismos participar con su presencia, su posición y su imagen. Podemos imaginar el cuerpo como un interruptor, o bien, podemos imaginar el cuerpo como la imagen de sí y analizar ésta usando fotos o videos en un número extenso de formas.

- Interfaces hápticas y *multitouch*: Recientemente se han desarrollado diversas herramientas táctiles. Dado el éxito que ha tenido esta tecnología, la velocidad de innovación es constante. Sin embargo, los fundamentos del diseño y la estructura de la interacción usada en estas interfaces manifiestan un patrón repetitivo en cuanto a los gestos en que se basa, sin presentar ningún cambio reciente. Esencialmente estos gestos están constituidos por aquellos con los que la mayoría de los usuarios – que han utilizado cualquier dispositivo de esta naturaleza – se han ido familiarizando, como son: usar dos dedos para expandir o contraer, girar dos dedos para rotar, pulsar para seleccionar. El lenguaje de estos gestos ha llegado a ser un lenguaje que es posible utilizar por los usuarios, permitiendo una interacción cada vez más común y natural. (J. Noble, 2009).

Es importante señalar que la interacción natural ha ido ganando terreno dentro de las ciencias computacionales. Representa un objetivo retador para los investigadores dentro del campo de estudio de la interacción hombre-computadora. El propósito final: que cada una de las señales digitales que se le entreguen al sistema sean traducidas por este en acciones multimedia amenas y comunes para el usuario.

2.2 Sistemas Multimedia

Dentro de los sistemas existentes es posible encontrar aquellos que mediante la mezcla de texto, imágenes, animaciones, sonido y video brindan información según la entrada de datos y el tipo de aplicación con el que se esté trabajando. Con mayor frecuencia, diferentes empresas están apostando por este tipo de sistemas; los cuales pueden ofrecer una experiencia más grata para el usuario final.

Con el actual rendimiento de los equipos de cómputo, las interfaces multimedia han permitido a los usuarios visualizar el cambio de los datos con mayor practicidad. Siendo así que se utilicen muchas de ellas para el entrenamiento de personal en diferentes

áreas, como la milicia, la parte médica, y, también, son de gran impacto en el mundo del entretenimiento. Este constante cambio ha llevado a incorporar diferentes medios de visualización, integrándolos con ambientes de inmersión cada vez más sofisticados.

Hoy en día se pueden encontrar diferentes aplicaciones multimedia; desde la oficina de trabajo, en los salones de clase, las consolas de videojuegos e interacciones cotidianas para el ámbito industrial.

Una de las piezas importantes a este respecto es el diseño del sistema y cómo se establecerá un lenguaje con el usuario; de tal modo que los mensajes enviados por este último sean entendidos por el sistema y que la respuesta que el sistema ofrezca sea entendible para el usuario. La interfaz que se desarrolle será el medio de comunicación entre el usuario y el sistema. Deberá ser un entorno amigable con colores, texto y simetría entre las imágenes que proporcione un ambiente en el que el usuario se sienta en confianza para interactuar con el sistema.

Como un ejemplo de lo anterior, en la actualidad existe un trabajo que propone un sistema multimedia utilizando una máquina de estados finita, capaz de interactuar con un sistema médico a distancia por medio del uso de un Kinect™ como el único dispositivo de entrada. El sistema permite al usuario utilizar las manos y brazos como medio de interacción 2D (L. Gallo *et al*., 2011).

2.3 Cámaras RGB-D

Las cámaras RGB-D (Rojo, Verde, Azul, más profundidad) son un desarrollo considerablemente novedoso que permite el acceso a información que las cámaras tradicionales solo podían brindar bajo la idea de una imagen 2D. Con la profundidad se puede obtener cuadros (*frames*) donde cada pixel representa la distancia cartesiana, en milímetros, desde el plano de la cámara al objeto más cercano en las coordenadas particulares X e Y del campo de visión en profundidad del sensor. Un valor de 0 en los datos de profundidad indica que no hay datos que se encuentren disponibles en esta posición, porque todos los objetos están demasiado cerca o lejos del dispositivo (Microsoft®, 2010).

Estas cámaras son capaces - aparte de brindar una imagen común RGB- de establecer un mapa de distancias de la escena (L. Gallo *et al*., 2011). La figura 1 nos muestra la información que la cámara nos puede proporcionar.

a) b)

Figura 1. a) Imagen común RGB, b) imagen en profundidad capturada a una velocidad de 30 cuadros por segundo con una resolución de 640x480. En la imagen se observa una escala de grises, la cual representa el nivel de profundidad del pixel.

Debido a que las cámaras RGB-D proporcionan una gran cantidad de información 3D del ambiente que capturan, presentan algunas desventajas que hacen aún difícil su aplicación en detección y reconocimiento de objetos. Estos dispositivos proveen información de profundidad en un rango acotado (más de 40cm); estos valores de profundidad son mucho más ruidosos que aquellos obtenidos por láser (principalmente en los contornos de los objetos) y su campo de visión (~60°) está, por mucho, más limitado que los de cámaras especializadas o escáneres láser usados en estas aplicaciones (~180°) (L. Contreras, 2011).

Para el presente trabajo se utilizó la cámara desarrollada por PrimeSense™ incluida en el dispositivo Microsoft® Kinect™, el cual está equipado con un proyector infrarrojo (IR) a base de un láser y un sensor monocromático CMOS. Así, el proyector IR es empleado para emitir un patrón moteado fijo hacia el área enfocada. El patrón antes mencionado es detectado por el sensor CMOS y utilizado para calcular la información en profundidad por triangulación contra un patrón de red manual (L. Gallo *et al*., 2011).

2.4 Detección del Esqueleto Humano con Kinect™

El seguimiento tanto del esqueleto humano como el de las manos ha representado un problema desafiante para la investigación, en cuanto a la visión computacional se

refiere; tanto que aún varios de los sistemas de visión requieren de marcadores (M. Gutiérrez *et al*., 2008; J. Lee, 2007; L. Vlaming, 2008; M. Ortega *et al*., 2008; K. Kanda *et al*., 2005; F. Peña *et al*., 2006) o de algún medio controlado (fondos dominantes, marcas características) (T. Collett *et al*., 2001; T. Fukushi, 2001; P. Graham *et al*., 2003; P. Müllier *et al*., 2001; D. Nicholson, *et al*., 1999), por lo que también es llamado rastreo asistido(E. Florian, 2009). De esta manera es más práctico detectar los puntos de interés.

Dentro del rastreo no asistido se han desarrollado diversas técnicas para el reconocimiento de gestos y múltiples tipos de cámaras para su detección (E. Florian, 2009). Estos sistemas proveen información de una forma más natural e interactiva, con la versatilidad de que no solo se puede trabajar sobre superficies táctiles, sino que también existe la posibilidad de interactuar sobre el espacio sin ningún tipo de artilugio sumado al sensor óptico (O. Hilliges *et al*., 2009; H. Benko, 2008).

Sin duda alguna, el dispositivo que ha marcado una importante tendencia en torno al desarrollo de aplicaciones con él fue el desarrollado por Microsoft®, puesto que es uno de los más baratos en el mercado; además de poseer un poder de procesamiento que puede ser calificado de bueno y novedoso.

El sistema Microsoft Kinect™ - inicialmente publicitado como la nueva experiencia en cuanto a entretenimiento sin controles se refiere - generó una notable inquietud entre muchos desarrolladores y programadores. Apenas después de algunos días de que el dispositivo había salido a la venta (Noviembre 4, 2010), ya existían librerías creadas por *hackers* con las cuales se puede interactuar con la cámara, sensor, motor y micrófonos con los que cuenta el Kinect™. A pesar de esto, aún se desconocían algunas funciones específicas con las que cuenta el Kinect™; y no fue hasta Junio del siguiente año que Microsoft® publicó cómo era que el dispositivo podía identificar a una persona, hacer el seguimiento de ésta y etiquetar dónde se encuentran las partes del cuerpo humano (J. Shotton *et al*., 2011). Ese mismo mes Microsoft® liberó su kit de desarrollo para Windows 7, el cual es capaz de darnos información en coordenadas 3D del seguimiento de hasta veinte articulaciones (*joints*). La figura 2 ilustra estas articulaciones.

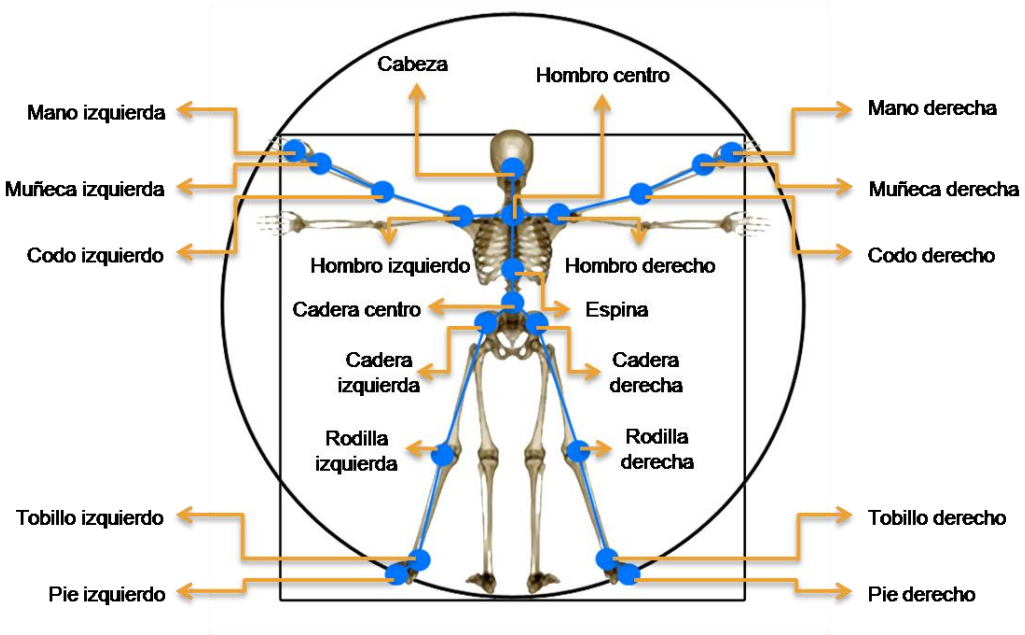


Figura 2. Imagen que representa las uniones del esqueleto detectado con el Kinect SDK de Microsoft® para Windows 7.

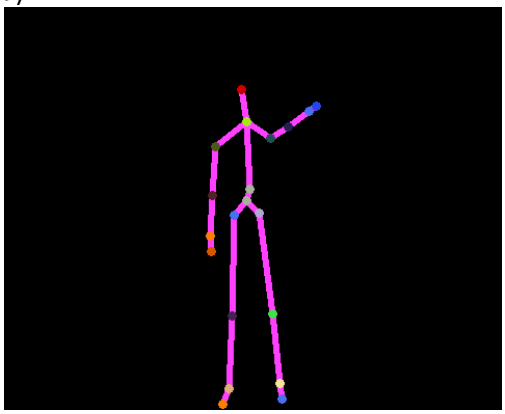
Tanto las recientes bibliotecas libres para el uso del Kinect™, como las que oficialmente se han publicado – que también son libres de descarga –, proporcionadas por Microsoft®, nos dan la posibilidad de manipular las diferentes interpretaciones de las imágenes que el dispositivo puede percibir.

Como se ha comentado, se puede obtener una imagen tradicional RGB. También hemos comentado del mapa de distancias por pixel que nos puede otorgar. Y dadas las bibliotecas, primeramente se puede identificar el cuerpo de una persona, sin importar la complexión o ropa que use. Con esto, se hace la aproximación a cada una de las uniones del esqueleto.

En este trabajo utilizamos la posición de las manos para detectar diferentes gestos de interacción a distancia con un modelo gráfico. Como se describe en la sección 3.1.

El Kinect™ y su software asociado, solo proporcionan la posición de las articulaciones y de las manos del usuario, como se muestra en la figura 3; por lo que es necesario implementar la detección de una secuencia de movimientos que definen un gesto (sección 3.2).

a) b)



c)

Figura 3. a) Imagen RGB, b) reconocimiento de una persona, c) imagen con información del esqueleto y cada una de las articulaciones detectadas.

2.5 Reconocimiento de patrones

El reconocimiento de patrones, como una rama de la inteligencia artificial, representa una investigación ardua en base a funciones matemáticas y aprendizaje de máquina. El objetivo principal de esta disciplina es reconocer patrones y asignar una etiqueta a aquellos que son de nuestro interés, en un cierto número de categorías o clases.

El reconocimiento de patrones tiene una larga historia, pero antes de la década de 1960 era mayoritariamente producción de la investigación teórica en el área de la estadística. Como con todo lo demás, el advenimiento de las computadoras incrementó la demanda de aplicaciones prácticas del reconocimiento de patrones, lo cual, en lo sucesivo, estableció nuevas demandas para más desarrollos teóricos. Mientras nuestra sociedad evoluciona de su fase industrial a la postindustrial, la automatización en la producción y la necesidad del manejo y recuperación de

información se vuelven cada vez más importantes. Esta tendencia ha empujado al reconocimiento de patrones a la cúspide de las aplicaciones de ingeniería e investigación de la actualidad. El reconocimiento de patrones es una parte integral de la mayoría de los sistemas de inteligencia artificial construidos para la toma de decisiones (S. Theodoridis *et al,.*2009).

Al patrón se le puede definir como aquel suceso u objeto que suele repetirse; entonces, al ser un fenómeno que suele repetirse, puede ser predecible. Claro está que no se podría reconocer sin antes dotar al sistema de las bases necesarias para que pueda hacerlo.

Por ejemplo, en un sistema que incluye la visión computacional se deben de sensar constantemente los movimientos de los gestos que se desean reconocer.

Una de las formas más interesantes para crear interacción utilizando visión computacional es usar reconocimiento de gestos, en donde el reconocimiento es la detección de un determinado tipo de objeto en una imagen y un gesto es un patrón de movimiento.

Ya que en muchos de los procesos que se le pueden hacer a la imagen se requiere tanto de almacenamiento, estudio y clasificación de esos mismos datos, es interesante conocer las técnicas de clasificación.

2.5.1 Selección de características

Una de las primeras tareas para el diseño de un sistema que sea capaz de reconocer patrones es la de seleccionar las mejores características que nos ayuden para encontrar y clasificar, de una forma óptima, los patrones que se quiere sean reconocidos.

La selección de características es un problema importante de la investigación en reconocimiento de patrones. En este trabajo evaluamos el desempeño tanto de un clasificador por el método de k-medias y de otro por el método de k-vecinos más cercanos, con características que se mencionarán en la sección 3.2.

2.6 Análisis de Cúmulos

El análisis de cúmulos es un método que permite segmentar datos de una imagen; de tal modo que sea posible obtener subconjuntos o cúmulos de datos que sean muy similares entre sí, que compartan un mismo perfil y que, además, se diferencien de otros subconjuntos o cúmulos con características propias.

Cada uno de estos conglomerados tendrán una etiqueta para poder distinguirle, también podemos llamarles clase.

Existen diversos métodos de clasificación, sin embargo, se pueden distinguir dos enfoques principales de este tipo de análisis:

- Clasificación no supervisada: No requiere información acerca de las clases en las cuales se desea segmentar.
- Clasificación supervisada: Forma la clasificación a partir de datos de entrenamiento.

2.6.1 Normalización

En muchas situaciones prácticas el diseñador es confrontado con características cuyos valores yacen dentro de diferentes rangos dinámicos. Así, características con valores grandes podrían tener una influencia más grande en la función costo que características con valores pequeños, aunque esto no necesariamente refleja su importancia respectiva del diseño del clasificador. El problema es superado mediante la normalización de las características, de modo que sus valores permanezcan dentro de rangos similares. Una técnica directa es la normalización vía los estimados respectivos de la media y la varianza. Para *N* datos disponibles de la k-esima característica tenemos:

$$\bar{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik}, k = 1,2, \dots, l$$

$$\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N} (x_{ik} - \bar{x}_k)^2$$

$$\hat{x}_{ik} = \frac{x_{ik} - \bar{x}_k}{\sigma_k}$$

Es decir, todas las características normalizadas resultantes ahora tendrán una media igual a cero y una variancia de uno, resultando en un método lineal. Otras técnicas lineales limitan los valores de la característica en un rango de [0, 1] o [-1, 1] en la escala apropiada. Además de los métodos lineales, los métodos no lineales pueden ser empleados también en casos en los cuales los datos no estén equitativamente distribuidos alrededor de la media; en tales casos, transformaciones basadas en funciones no lineales (esto es: logarítmicas o sigmoides) pueden ser utilizadas para mapear datos dentro de intervalos específicos (S. Theodoridis *et al,.*2009).

En la figura 4a se muestra un caso para dos dimensiones en el que el rango de variación de una de las características es notoriamente mayor respecto al otro, con lo que la distancia de un nuevo punto *p* a cualquier punto de la base está determinada principalmente por el parámetro de mayor rango. Para solucionar esto se obtiene el mismo sistema con parámetros normalizados; en el que la variación de cualquiera de los parámetros influye en la misma proporción para el cálculo de la distancia de un punto nuevo a cualquiera de la base (figura 4b).

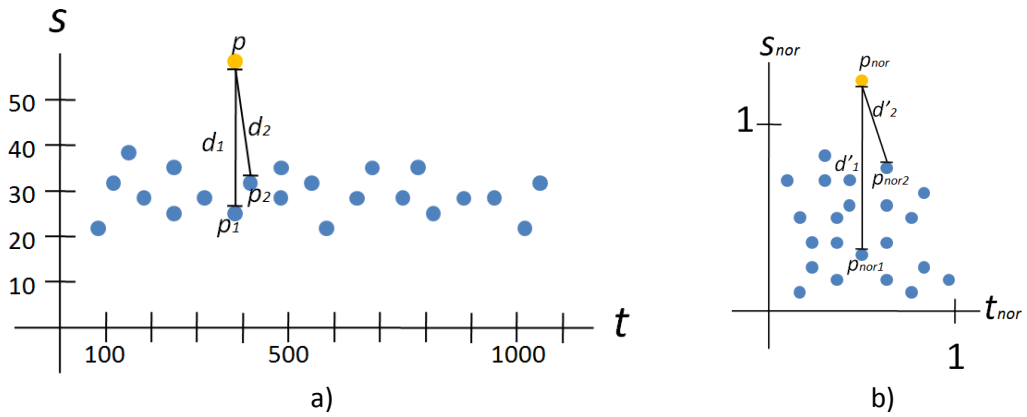


Figura 4. Variación de la distancia de un punto nuevo *p* a cualquiera de la base. a) Sin normalización, con d_1 semejante a d_2 y b) normalizado, en donde d'_1 es notoriamente diferente a d'_2.

2.6.2 K-Vecinos más cercanos

Un método de clasificación básica consiste en asignar al mismo subconjunto un caso similar referente a otro, cuya clasificación ya es conocida. Así, el método de clasificación del vecino más cercano tiene una base de datos con clases conocidas; por tanto, se dice que es un método de clasificación supervisada donde el aprendizaje de cada clase tiene una base de entrenamiento previa a la clasificación.

De tal manera que existiendo un nuevo caso (objeto o elemento), este se va a clasificar en la clase donde sus *k* vecinos más cercanos tienen una distancia (generalmente se utiliza la distancia euclidiana) menor o se puede hacer una evaluación hacia donde estos son más recurrentes.

El proceso de clasificación se puede observar en la figura 5, donde cada uno de los colores representa una clase y cada círculo un caso de la base de entrenamiento. El nuevo caso que se intenta clasificar es representado con el círculo color naranja. Primeramente se calculan todas las distancias (flechas color negro) hacia cada uno de los casos de entrenamiento, y después se obtienen los *k* vecinos más cercanos (señalados con una flecha naranja). En la figura el nuevo caso será asignado a la clase azul, dado que los vecinos más cercanos son más recurrentes hacia esta clase.

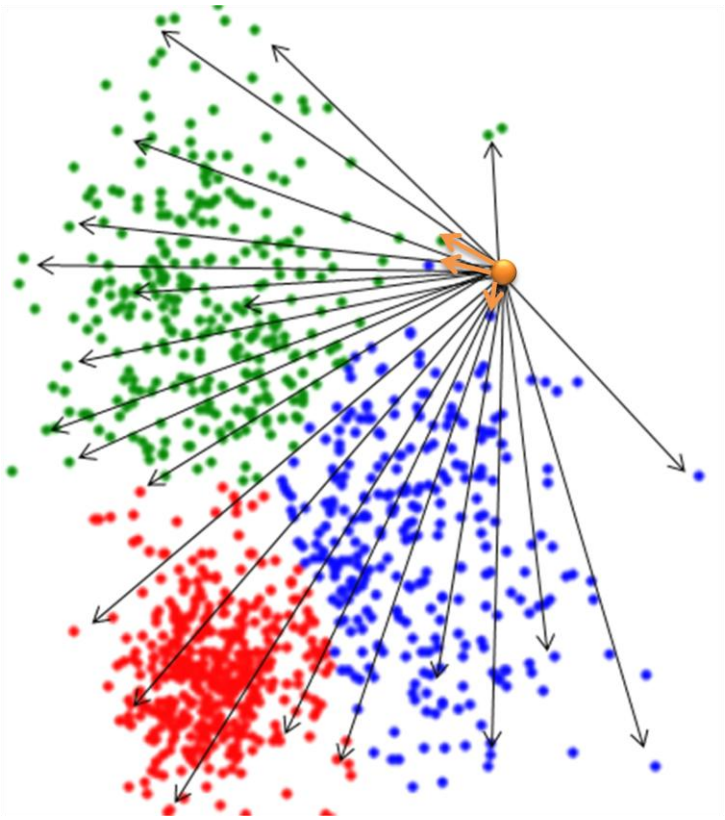


Figura 5. Ejemplo de una clasificación por *k* vecinos más cercanos, en donde *k*=3. El nuevo caso podría clasificarse dentro de la clase azul ya que dos de los vecinos más cercanos están más cerca a este.

2.6.3 K-Medias

El algoritmo de k-medias es un método de análisis de cúmulos que corresponde al método de clasificación no supervisada; es una técnica iterativa que permite agrupar valores de un grupo de datos por su parecido o similitud. Al final, se generan *k* centroides en la nube de datos que mejor agrupen conjuntos de valores.

En la figura 6 se ilustra un ejemplo de este procedimiento. Con el conjunto inicial de datos (figura 6a) se plantea una primera hipótesis aleatoria, mostrada como puntos rojos en (figura 6b) (refiriéndonos a ellos como 'centroides'); se calcula la distancia euclidiana de cada punto del conjunto a cada uno de ellos y asignándoles una pertenencia al más cercano, obteniendo así regiones de dominio. De estas regiones se obtiene el centroide de todos los puntos que pertenecen a ella y se considera como nueva hipótesis (como se muestra en figura 6c). Se repite iterativamente este procedimiento hasta alcanzar un número máximo de repeticiones o un criterio de

convergencia, como puede ser la poca variación entre el centroide actual y el siguiente (figura 6d, 6e y 6f) (L. Contreras, 2011).

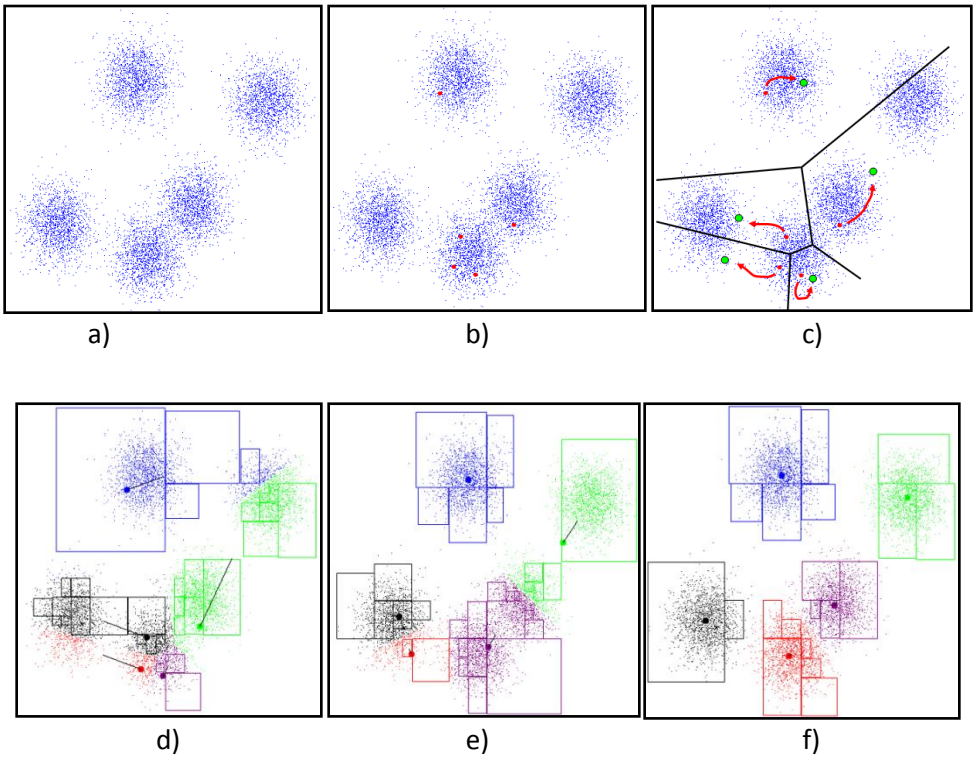


Figura 6. Etapas del algoritmo de k-medias.

Capítulo 3. Implementación

3.1 Definición de gestos de interacción

Para comenzar la implementación de la investigación fue necesario definir cada uno de los gestos con los cuales será posible que el usuario interactúe con el sistema.

Como se ha dicho anteriormente, para este tipo de sistemas se tiene que pensar en el usuario y cómo es que puede interactuar con la aplicación multimedia, de modo que se pueda establecer un lenguaje entre estos agentes de interacción. El usuario potencial de este sistema es un cirujano y el ambiente de trabajo entonces será una sala de operaciones.

Una premisa relevante para la asignación de los gestos fue la consecución de un sistema que sea fácilmente manipulable por el usuario.

Atendiendo a un criterio de facilidad en la operación, se pensó en gestos que no representen un esfuerzo físico, desgastante o tedio, que hagan que el usuario (un cirujano) termine cansado o sin ganas de querer interactuar con el sistema nuevamente.

La interacción consiste en que el usuario debe estar de pie frente al Kinect™, a una distancia promedio de 1.2m a 2.5m; desde esa latitud podrá realizar los diferentes gestos que a continuación se mencionan.

Se definieron gestos con las manos que pueden realizarse de arriba de la cintura, lo que ayuda a una mejor detección dentro de una sala de operaciones.

Aunque las ilustraciones siguientes presentan formas de manos con las palmas extendidas, con el puño cerrado o con el dedo índice apuntando; cabe señalar que nos basamos en las articulaciones de las manos (sección 2.4).

3.1.1 Zoom in

El método *zoom in* que comúnmente es utilizado en los dispositivos táctiles (deslizando un par de dedos en dirección contraria a cada uno de ellos) ha establecido una

tendencia de cómo puede ser ejecutado este gesto. Esta investigación emplea una aproximación similar, aprovechando la experiencia que los usuarios poseen en su aplicación.

La activación de dicho gesto es que el usuario extienda sus manos en dirección contraria de manera horizontal. Esto se ilustra en la figura 7.

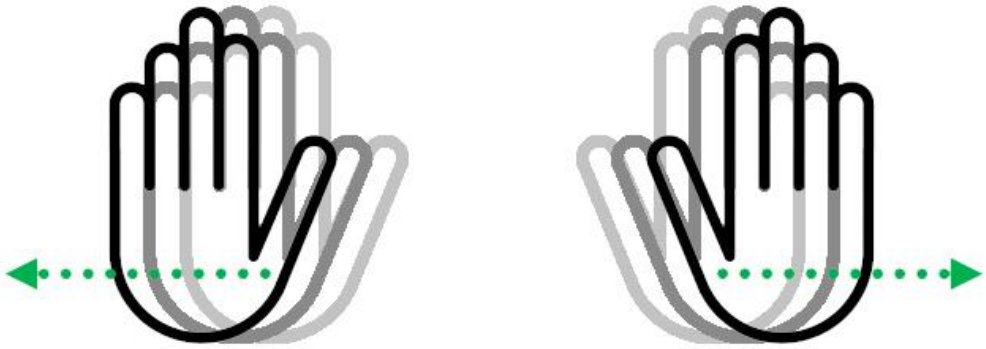


Figura 7. Representación de la interacción para el gesto de *zoom in*. Las ilustraciones con un tono más claro indican la posición inicial de las manos y las ilustraciones en un tono completamente negro indican la posición final.

Este gesto, junto con el que corresponde a *zoom out*, representan la interacción con alguna interface o aplicación en la cual se pueda acercar o alejar una imagen, conforme se está viendo o apareciendo en el *display*.

3.1.2 Zoom out

De manera contraria al gesto de *zoom in*, para la activación del gesto de *zoom out* las manos se acercarán hacia cada una de ellas (figura 8). También es de forma semejante como se utiliza en los actuales dispositivos de interacción táctil (deslizando un par de dedos hacia la intersección entre ellos).

Figura 8. Representación de la interacción para el gesto de *zoom out*. Las flechas con línea punteada indican hacia donde es que se están moviendo cada una de las manos.

3.1.3 Paneo

El gesto de paneo representa la interacción con la cual es posible mover la imagen -sobre el plano XY- que actualmente se está viendo en la pantalla.

El modo con el que es posible interactuar con el sistema para lograr esta acción consiste en mover las manos de manera paralela en cualquier dirección; siempre y cuando se respete que se muevan de esta forma y en sincronía, para desplazarse por el plano XY. Las manos deben tener una distancia mayor aproximada de 30 centímetros. La figura 9 ilustra el gesto del paneo.

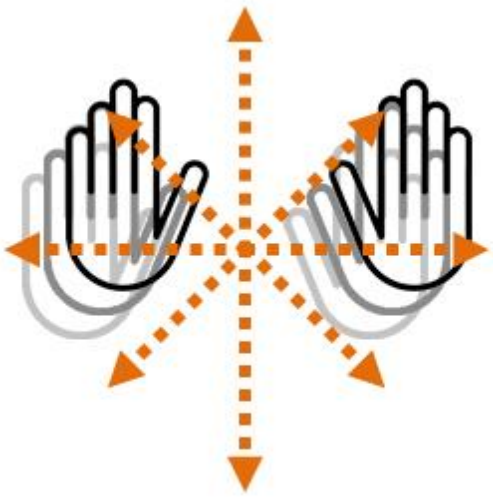


Figura 9. Representación de la interacción para el gesto de paneo. Las manos se deben mover de una forma semejante.

3.1.4 Puntero

El gesto mediante el cual es posible mover el puntero del mouse se define mediante el movimiento de la mano derecha por el plano XY (figura 10).

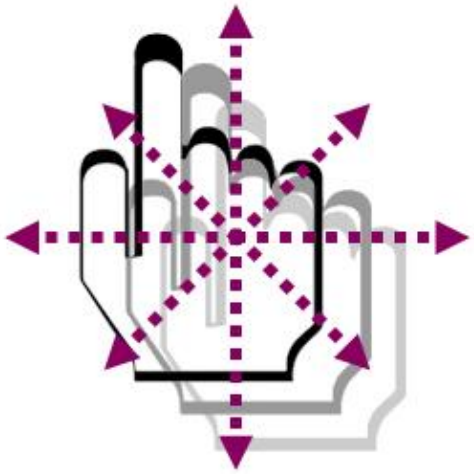
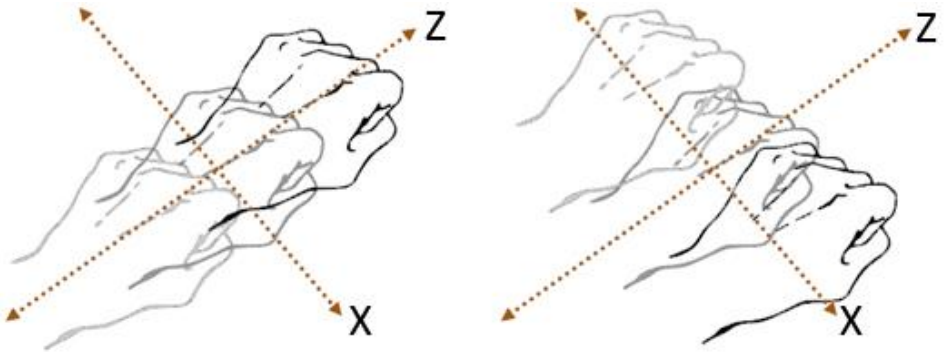


Figura 10. Representación de la interacción para el gesto de puntero.

Una anotación importante para ejecutar de forma correcta este gesto es que la mano izquierda debe permanecer relativamente pegada al cuerpo.

3.1.5 Rotación

La siguiente imagen (figura 11) representa el gesto que debe realizarse para la función de rotación. Recordemos que para todos los gestos que aquí se presentan el usuario se encuentra de frente al Kinect™. Por lo tanto, el eje Z es aquel que representa la profundidad (entre el usuario y la cámara RGB-D).



a) b)

Figura 11. a) Representación de la interacción para el gesto de rotación sobre el eje Z, b) rotación sobre el eje X.

Este gesto se pensó considerando la clásica esfera de rotación que diferentes aplicaciones 3D (de diseño o navegación) comúnmente utilizan para girar, ya sea un modelo tridimensional o la cámara del ambiente virtual.

Con la mano izquierda se realiza el movimiento a lo largo del eje Z para una rotación sobre el eje X en el modelo 3D y se mueve la mano a través del eje X para una rotación sobre el eje Y dentro de la aplicación (sección 5.1).

3.2 Análisis de Cúmulos

Para definir los gestos que se desea detectar, existen características establecidas para su reconocimiento. Por ello, se crea un patrón que será aquel arreglo que contenga los valores de cada una de las características que se piensan convenientes para el reconocimiento de los gestos antes descritos.

Dada la información que puede proveer el Kinect™ acerca del *esqueleto* detectado, las características que definen al patrón fueron basadas en las coordenadas de la ubicación de las manos y algunas de ellas utilizan el cuadro anterior al que se reconoce actualmente.

Los patrones que se conformaron para esta investigación fueron de diferentes dimensiones según la investigación avanzaba. Los vectores son los siguientes:

- 6 dimensiones:
    - *ΔxIzq, ΔyIzq, deuclXY, ΔxDer, ΔyDer, deltaZ.*
- 7 dimensiones:
    - *ΔxIzq, ΔyIzq, ΔzIzq, deuclXYZ, ΔxDer, ΔyDer, ΔzDer.*
- 8 dimensiones:
    - *ΔxIzq, ΔyIzq, ΔzIzq, deuclXYZ, ΔxDer, ΔyDer, ΔzDer, deltaZ*
- 9 dimensiones:
    - *ΔxIzq, ΔyIzq, ΔzIzq, deuclXYZ, ΔxDer, ΔyDer, ΔzDer, deltaManoDerAtorso, deltaManoIzqAtorso.*

En donde,

$\Delta x\text{Izq} = manoIzqX_{frame\_actual} - manoIzqX_{frame\_anterior}$

$\Delta y\text{Izq} = manoIzqY_{frame\_actual} - manoIzqY_{frame\_anterior}$

$\Delta z\text{Izq} = manoIzqZ_{frame\_actual} - manoIzqZ_{frame\_anterior}$

$\Delta x\text{Der} = manoDerX_{frame\_actual} - manoDerX_{frame\_anterior}$

$\Delta y\text{Der} = manoDerY_{frame\_actual} - manoDerY_{frame\_anterior}$

$\Delta z\text{Der} = manoDerZ_{frame\_actual} - manoDerZ_{frame\_anterior}$

} Representa la diferencia de cada coordenada en cada una de las manos.

deuclXY, representa la función para obtener la distancia euclidiana (para los ejes X, Y) entre las manos.

$$deuclXY = \sqrt{(manoIzqX - manoDerX)^2 + (manoIzqY - manoDerY)^2}$$

deltaZ, es la distancia absoluta en el eje Z que hay entre cada una de las manos.

$$deltaZ = |manoIzqZ - manoDerZ|$$

deuclXYZ, representa la función para obtener la distancia euclidiana (para los ejes X, Y, Z) entre las manos.

$$deuclXYZ = \sqrt{(manoIzqX - manoDerX)^2 + (manoIzqY - manoDerY)^2 + (manoIzqZ - manoDerZ)^2}$$

deltaManoDerAtorso, es la diferencia absoluta en el eje Z que hay de la mano derecha al torso.

$$deltaManoDerAtorso = |manoDerZ - torsoZ|$$

deltaManoIzqAtorso, es la diferencia absoluta en el eje Z que hay de la mano izquierda al torso.

$$deltaManoIzqAtorso = |manoIzqZ - torsoZ|$$

Las figuras 12 y 13 ilustran estos elementos.

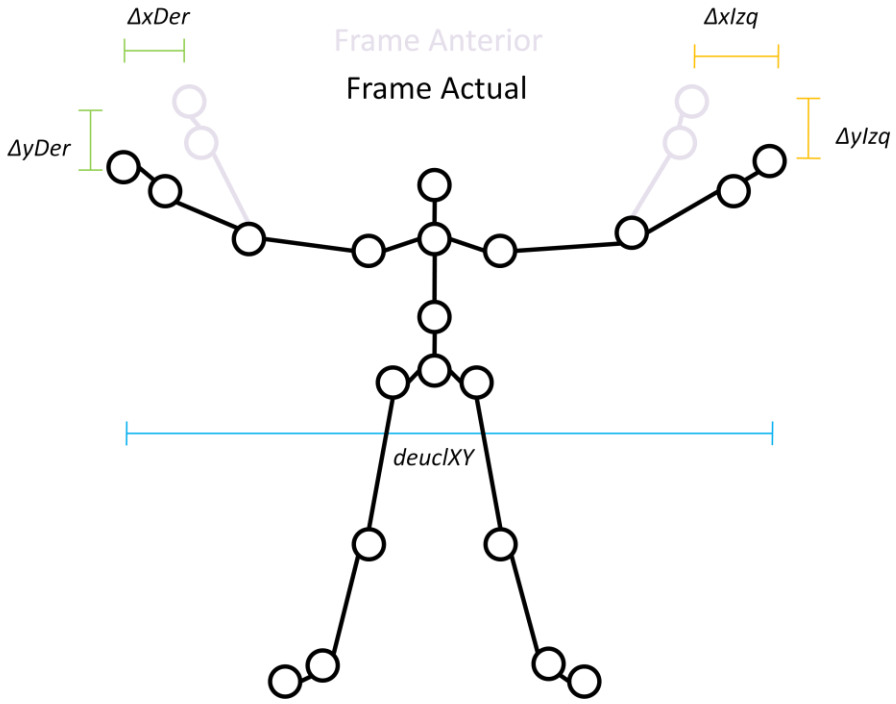


Figura 12. Representación de algunas de las características que se incluyen en los diferentes arreglos. Aquí se pueden observar las que actúan sobre los ejes X e Y

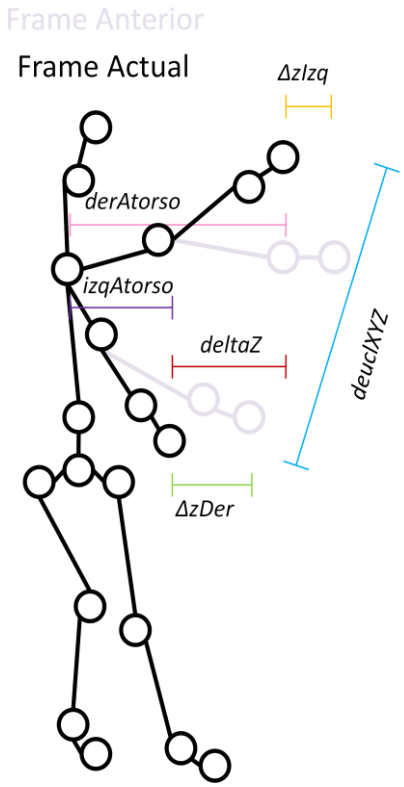
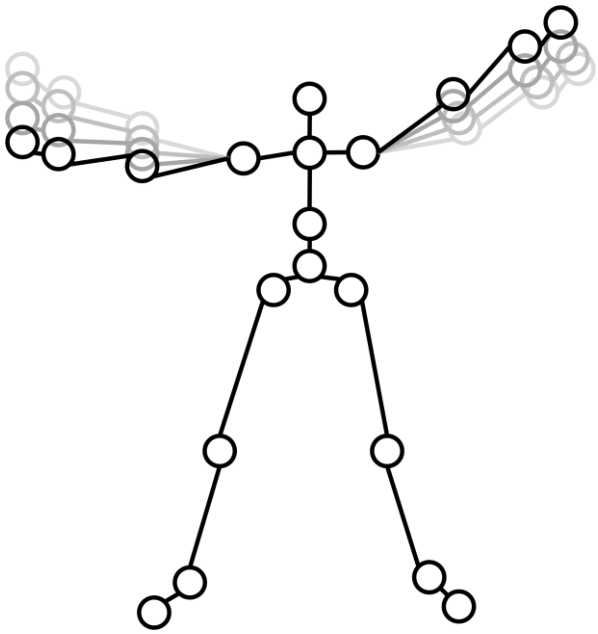


Figura 13. Imagen que representa algunas de las características que se incluyen en los diferentes arreglos. Vista en perspectiva para mostrar características en el eje Z.

3.2.1 Base de entrenamiento

El reconocimiento de los gestos precisa la existencia de una base de datos con la cual sea posible comparar el vector que se genere en tiempo real. Para ello, se creó una grabación de 320 muestras aproximadamente -dado que se observó que eran las muestras promedio y suficientes para utilizar cada uno de los gestos-, se almacena el patrón cuadro a cuadro con las respectivas características que antes mencionamos y se guardó en un archivo de texto (figura 14). Una vez que se completan las 320 muestras se genera un archivo (un archivo fue creado por cada uno de los gestos).



1. -0.000276 -0.000005 0.000061 0.767089 -0.000324 0.000075 0.000210 0.203482 0.214878
2. -0.000267 0.000003 0.000057 0.767037 -0.000320 0.000074 0.000206 0.203327 0.214574
3. -0.000259 0.000011 0.000054 0.766983 -0.000313 0.000072 0.000201 0.203181 0.214281
4. -0.000252 0.000017 0.000052 0.766929 -0.000306 0.000071 0.000196 0.203040 0.213996

...

320. -0.009699 0.009742 -0.043646 0.37979 0.015270 -0.043178 0.018480 0.352069 0.425527

Figura 14. Formación del archivo de grabación con un patrón de nueve dimensiones. Con los diferentes tonos de gris en la imagen se intenta mostrar que cuadro a cuadro se van grabando cada uno de los vectores.

Debido a que fue encontrado deseable que la mayoría de los datos de grabación fueran los que describieran a cada uno de los gestos, se generó una posición de grabación. Una vez que es detectada esta postura (la cual consiste en levantar ambas manos en un ángulo cercano a los noventa grados), la aplicación espera 2 segundos para la grabación (figura 15).

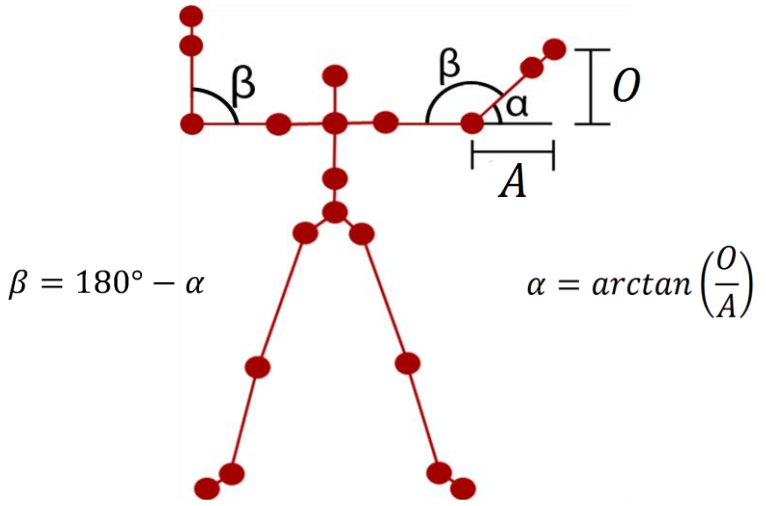


Figura 15. Gesto de grabación. Si $\beta > 75$ y $\beta < 90$ entonces se activa el gesto de grabación.

Para un mejor reconocimiento proponemos sea conveniente que el patrón tenga un historial cada cinco cuadros. Esto es que el cuadro actual incluye a los cuatro anteriores. En la figura 16 podemos observar un ejemplo de lo mencionado.

1	-0.001390	0.000374	0.000163	0.300406	0.001200	0.000186	-0.000028	0.333417	0.313511
2	-0.001368	0.000391	0.000179	0.302977	0.001221	0.000158	-0.000033	0.333274	0.313580
3	-0.001349	0.000405	0.000191	0.305565	0.001258	0.000128	-0.000043	0.333119	0.313659
4	-0.001323	0.000415	0.000201	0.308163	0.001297	0.000099	-0.000051	0.332953	0.313746
5	-0.001303	0.000423	0.000209	0.310763	0.001320	0.000073	-0.000057	0.332779	0.313837
6	-0.001293	0.000432	0.000217	0.313368	0.001336	0.000049	-0.000061	0.332597	0.313933

Fila 1
Fila 2

...

315	0.001204	-0.000081	0.000276	0.801893	0.001364	0.0002	0.000522	0.257752	0.251248
316	-0.001187	-0.000083	0.000286	0.804447	0.001366	0.000194	0.000512	0.257470	0.250741
317	-0.001206	-0.000085	0.000285	0.806992	0.001339	0.000187	0.000507	0.257189	0.250238
318	-0.001189	-0.000086	0.000294	0.809504	0.001321	0.000181	0.000502	0.256899	0.24974
319	-0.001201	-0.000092	0.000292	0.812	0.001294	0.000173	0.000494	0.256611	0.24925
320	-0.001208	-0.000102	0.000291	0.814481	0.001273	0.000167	0.000487	0.256323	0.248767

Fila 315
Fila 316

Figura 16. Obtención de los vectores con historia de cinco cuadros.

De tal modo que la matriz se reducirá en cuatro filas, pero el número de columnas se multiplicará por cinco. Entonces, si se tiene la matriz de vectores *A* de *mxn*, al integrarla con una historia de cinco cuadros se obtiene la matriz *B* de *pxq*, en donde $p = m - k + 1$ y $q = n * k$. En donde, k es igual al número de cuadros de historial (figura 17). Por ejemplo, si $m = 320$ y $n = 9$, entonces $p = 316$ y $q = 45$.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & & a_{1,\mathrm{n}} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,\mathrm{n}} \\ a_{3,1} & a_{3,2} & a_{3,3} & & a_{3,\mathrm{n}} \\ & \vdots & & \ddots & \vdots \\ a_{\mathrm{m},1} & a_{\mathrm{m},2} & a_{\mathrm{m},3} & \cdots & a_{m,\mathrm{n}} \end{bmatrix} \rightarrow$$

$$B = \begin{bmatrix} a_{1,1} & \cdots & a_{1,\mathrm{n}} & a_{2,1} & \cdots & a_{2,\mathrm{n}} & \cdots & a_{\mathrm{k},1} & \cdots & a_{\mathrm{k},\mathrm{n}} \\ a_{2,1} & \cdots & a_{2,\mathrm{n}} & a_{3,1} & \cdots & a_{3,\mathrm{n}} & \cdots & a_{\mathrm{k}+1,1} & \cdots & a_{\mathrm{k}+1,\mathrm{n}} \\ a_{3,1} & \cdots & a_{3,\mathrm{n}} & a_{4,1} & \cdots & a_{4,\mathrm{n}} & \cdots & a_{\mathrm{k}+2,1} & \cdots & a_{\mathrm{k}+2,\mathrm{n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{\mathrm{m}-\mathrm{k}+1,1} & \cdots & a_{\mathrm{m}-\mathrm{k}+1,\mathrm{n}} & a_{\mathrm{m}-\mathrm{k}+2,1} & \cdots & a_{\mathrm{m}-\mathrm{k}+2,\mathrm{n}} & \cdots & a_{\mathrm{m},1} & \cdots & a_{\mathrm{m},\mathrm{n}} \end{bmatrix}$$

Figura 17. Formación de la matriz de entrenamiento con historial.

Una vez que se ha formado la matriz de los patrones con historial para cada uno de los gestos se procede a unirlas en una sola; conformándose así la base de entrenamiento completa. Esto ayudará a tener una mejor normalización de los datos.

3.2.2Normalización

Se buscará que los datos de la base de entrenamiento sean homogéneos. Por ello, se normalizarán los datos por cada columna. Para normalizar se estudiaron dos métodos de normalización. Uno de ellos es en base al rango y otro es por media y desviación estándar:

Sea B una matriz formada por vectores fila, en el que cada una de sus columnas tiene elementos con dimensiones diferentes:

$$B_{m\times n} = \begin{matrix} \mathrm{k{=}1} & \mathrm{k{=}2} & \mathrm{k{=}3} & & \mathrm{k{=}n} \\ \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & & a_{1,\mathrm{n}} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,\mathrm{n}} \\ a_{3,1} & a_{3,2} & a_{3,3} & & a_{3,\mathrm{n}} \\ & \vdots & & \ddots & \vdots \\ a_{\mathrm{m},1} & a_{\mathrm{m},2} & a_{\mathrm{m},3} & \cdots & a_{m,\mathrm{n}} \end{bmatrix} \end{matrix}$$

Normalización por rango:

for *k*

$$rango_k = max(A_k) - min(A_k)\ ;$$

$$a_{m,k} = {}^{a_{m,k}}/_{rango_k}\ ;$$

Normalización por media y desviación estándar:

for *k*

$$\mu_k = media(A_k)\ ;$$

$$\sigma_k = std(A_k)\ ;$$

$$a_{m,k} = \frac{a_{m,k} - \mu_k}{\sigma_k}\ ;$$

3.3 Clasificador de gestos

Para la clasificación supervisada se generó una base de datos con elementos conocidos pre-clasificados de acuerdo a los gestos ya definidos. Entonces, llegado un nuevo patrón en tiempo real, se le asignará una clase según los valores que contenga.

El patrón en tiempo real también cuenta con una historia de cinco cuadros. Este va tomando el nuevo cuadro y lo adhiere al vector con historia, pero en ese momento se excluye el primer cuadro de los cinco en haber sido detectado.

$$\underbrace{frame_{actual-4} frame_{actual-3} frame_{actual-2} frame_{actual-1} frame_{actual}}_{\text{patrón con historia en tiempo real}}$$

Antes de utilizar los datos de entrenamiento para clasificarlo con k-medias o k-vecinos más cercanos, se hizo un clasificador mediante una máquina de estados, que es como propone L. Gallo *et al*. (2010) para una clasificación de gestos y así poder observar de manera práctica cómo se comporta éste sistema.

3.3.1 Máquina de estados

La máquina de estados nos brinda una clasificación rígida; dado que para reconocer los gestos que hemos propuesto se basan en condiciones determinadas y no se le da al sistema opción a reconocer más de lo que está expuesto en sus restricciones. Por ello es que su lógica de este sistema comprende solo un cierto rango de posibilidades delimitadas por ciertas distancias (en centímetros) establecidas.

Para la investigación, proponemos cómo es que la máquina de estados pueda reconocer los gestos. Cada uno de los gestos funciona bajo cierta lógica que a continuación se menciona. Para la activación de cada uno de los gestos tenemos que:

if$\Delta y \rightarrow 0$
if$\Delta z \rightarrow 0$
  if$\Delta pmX \rightarrow 0$
if$\Delta pmY \rightarrow 0$
if$\Delta pmZ < 32\text{cms}\ and\ \Delta x > 15\text{cms}$
if$\Delta x_{\text{frame\_actual}} > \Delta x_{\text{frame\_actual-1}}$

} Zoom in

if$\Delta y \rightarrow 0$
if$\Delta z \rightarrow 0$
if$\Delta pmX \rightarrow 0$
if$\Delta pmY \rightarrow 0$
if$\Delta pmZ < 32\text{cms}\ and\ \Delta x > 15\text{cms}$
if$\Delta x_{\text{frame\_actual}} < \Delta x_{\text{frame\_actual-1}}$

} Zoom out

if$\Delta x < 32\text{cms}\ and\ \Delta x > 15\text{cms}$
if$\Delta y \rightarrow 0$
if$\Delta z \rightarrow 0$
if$\Delta pmZ > 29\text{cms}$

} Paneo

if$\Delta spineToDer > 35\text{cms}$
if$\Delta spineToIzq \rightarrow 0$

} Puntero

if$\Delta spineToDerZ \rightarrow 0$
if$\Delta spineToIzqZ > 25\text{cms}$

} Rotación

en donde,

Δ*x*, Δ*y*, Δ*z*, representan la diferencia absoluta entre cada una de las manos por cada eje.

Δpm*X*, Δpm*Y*, Δpm*Z*, representan la diferencia absoluta que hay entre el punto medio que existe entre las manos y el torso, esto por cada eje.

Δ*spineToDerZ*, Δ*spineToIzqZ*, representan la diferencia absoluta en el eje Z entre el torso y cada una de las manos.

Una vez definidos cómo es que se reconoce cada uno de los gestos en tiempo real, la máquina de estados funciona según lo representado en la figura 18.

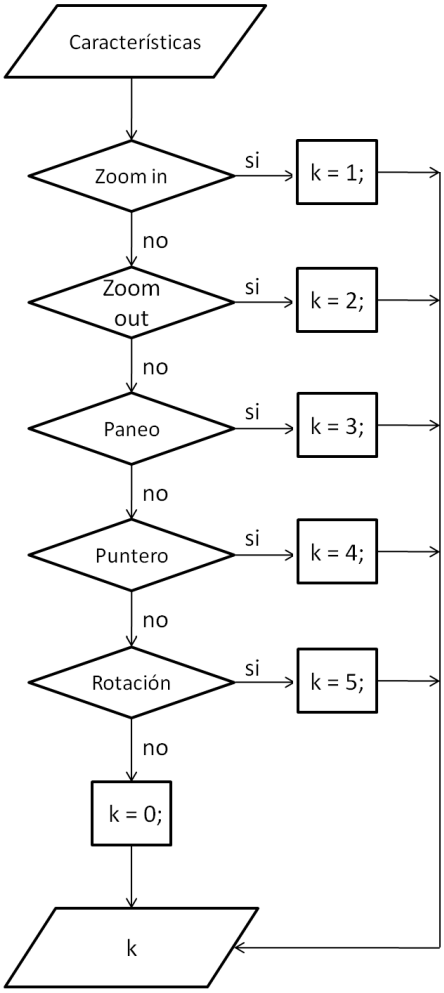


Figura 18. Máquina de estados para el reconocimiento de gestos.

3.3.2 K-Medias

La clasificación por el método de k-medias, siguiendo lo planteado por el algoritmo, se realiza a través de la utilización de la base de detección –con los datos ya normalizados–; de manera que se obtengan los *k* centroides, que son el punto medio de cada una de las clases.

Una vez que se tienen éstos, se tomará la distancia de cada uno hacia el nuevo patrón con historia -normalizado bajo la base de entrenamiento- que se esté leyendo en tiempo real.

Para el reconocimiento de gestos utilizando este método se tomó como base las herramientas de estadística con las que cuenta MATLAB; donde está incluida una función para hacer clasificación a partir de una base de datos llamada *kmeans*.

3.3.3 K-Vecinos más cercanos

El algoritmo que se desarrolló para la clasificación de k-vecinos más cercanos - como se enunció con anterioridad- se lleva a partir de una base de entrenamiento.

Como se sabe que la unión, al momento de formar la matriz general (con todos los datos de cada uno de los archivos que contenía información por cada uno de los gestos), fue de forma supervisada, es posible conocer cada uno de los límites de cada una de las clases; de igual manera, se puede advertir el índice de la columna donde inicia un gesto y también el índice donde termina.

Luego entonces – según el algoritmo –, leyendo un nuevo patrón en tiempo real que se quiere clasificar, se parte inicialmente calculando la distancia de este hacia cada uno de los patrones de entrenamiento. Posteriormente, las *n* distancias serán ordenadas; en donde *n* es el número de vectores en la base de entrenamiento, dicho ordenamiento se realizará de menor a mayor. Cabe mencionar que se trata de un ordenamiento indexado, de forma que podemos saber de cuál clase proviene cada una de las distancias. Una vez ordenadas, se obtienen las *k* distancias más cercanas.

En la implementación que se llevó a cabo en este estudio se efectuó una variación de la *k* con valores de 7, 14 y 21.

Para la clasificación se evalúa el valor de los k-vecinos más cercanos según la clase a la que corresponda. La clase que tenga mayor índice de pertenencia es con la que se etiquetará al nuevo patrón.

Existen los casos donde detectándose un nuevo patrón éste se encuentre muy alejado de la base de entrenamiento y aún así se puedan obtener los k vecinos más cercanos; sin embargo, el resultado podría representar un gesto que no se está anticipando, sería erróneo.

En este sentido, para delimitar casos como estos, se plantea una distancia máxima para cada uno de los gestos, a la que el nuevo patrón se podría clasificar. Si la distancia entre el nuevo patrón y el de la base de entrenamiento va más allá de la distancia máxima, el gesto no se considerará como un vecino más cercano (figura 19).

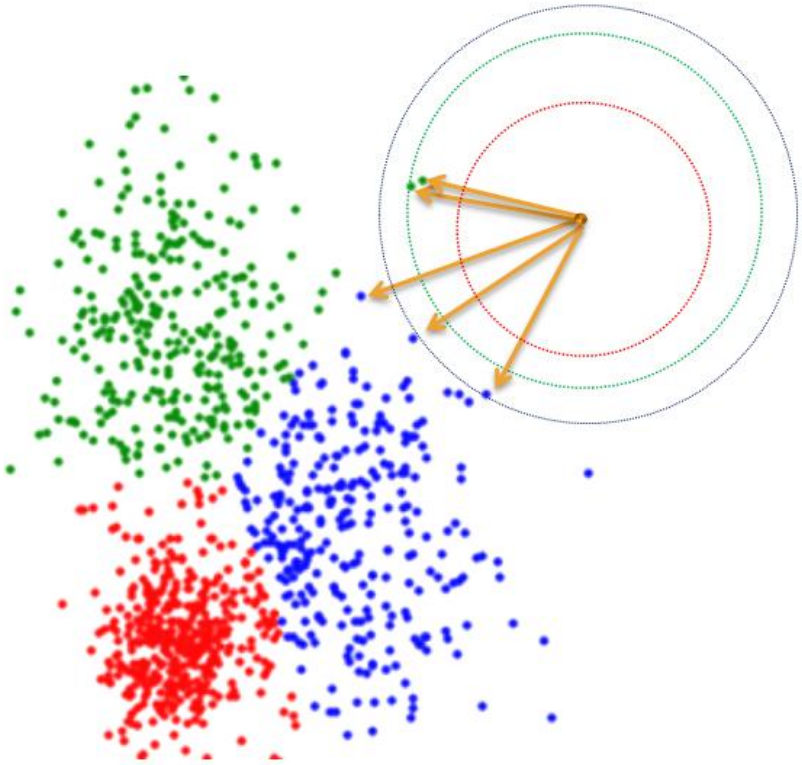


Figura 19. Los círculos de cada color representan la distancia máxima por clase. De tal manera se determinará si el vecino más cercanos puede ser considerado para la clasificación del nuevo patrón.

3.3.4 Distancia Euclidiana y distancia Coseno

En esta investigación se utilizó la distancia euclidiana y la distancia coseno que nos provee la función del programa MATLAB. Dicha función, llamada *pdist2*, es capaz de arrojar la distancia que existe entre dos conjuntos de observaciones.

Dados el vector X_s e Y_t la distancia euclidiana puede ser definida como:

$$d_{st}^2 = (x_s - y_t)(x_s - y_t)'$$

para la distancia coseno:

$$d_{st} = \left(1 - \frac{x_s y_t^{'}}{\sqrt{(x_s x_s^{'})(y_t y_y^{'})}} \right)$$

La distancia coseno no trabaja mejor que la distancia euclidiana, aunque nos ofrece la propiedad única de que el valor de la distancia está inherentemente normalizado para una determinada característica (G. Qian *et al*., 2004).

Para la implementación del método de k-vecinos más cercanos se propone multiplicar la distancia euclidiana por la distancia coseno, que se interpretó como la proyección de la distancia euclidiana en la dirección del vector de referencia.

Con ello, aunque la distancia euclidiana pueda ser la misma entre un par de puntos y el patrón de entrenamiento, la distancia coseno marca una diferencia entre ellos. Esto lo podemos observar en la figura 20.

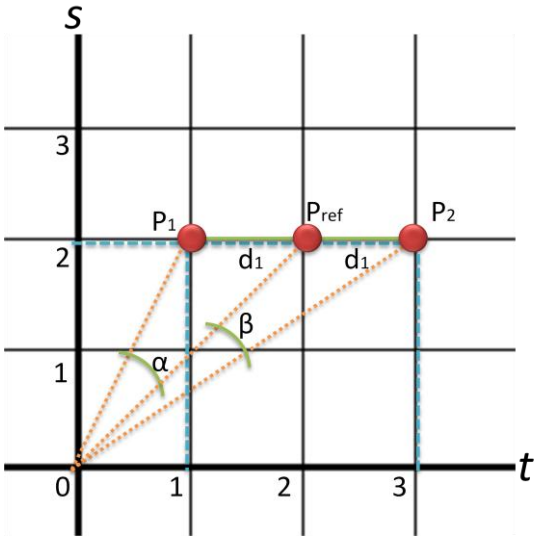


Figura 20. Representación de casos con la misma distancia euclidiana hacia un punto de entrenamiento. Pero diferente distancia coseno.

Para el ejemplo anterior, donde se desea saber cuál es la distancia menor entre un par de puntos -P_1 y P_2- hacia un punto de referencia -P_{ref}- que tienen la misma distancia euclidiana. El punto más cercano al punto de referencia será aquel en donde el ángulo que se forma entre ellos sea el menor.

Con lo anterior, cabe mencionar que el peor caso para esta implementación -donde se utilizan las dos distancias- será aquel donde tanto la distancia euclidiana como la distancia coseno sea de igual magnitud como se ilustra en la imagen (figura 21).

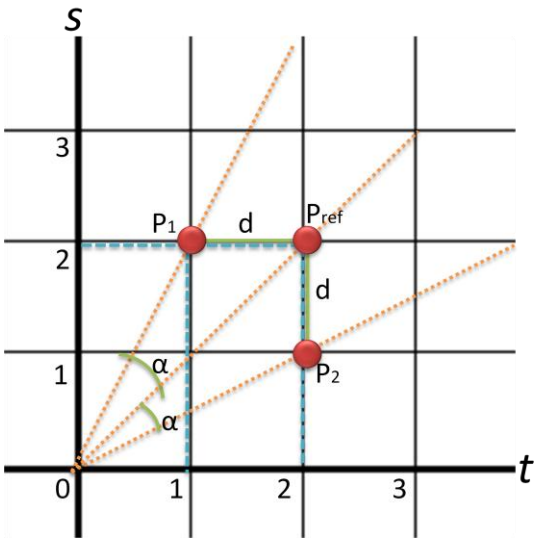


Figura 21. Peor caso donde la distancia euclidiana es la misma que la distancia coseno.

Capítulo 4. Resultados

4.1 Procedimiento

El procedimiento empleado para conseguir los resultados de esta investigación consiste en la observación del comportamiento de cada uno de los patrones con diferentes dimensiones (según las características que contiene), la normalización de sus datos (rango y media(μ)-sigma(σ)) y el método que pueda reconocer esos datos, como los gestos que son de interés para esta investigación (sección 4.1.1). Para este estudio se utiliza la clasificación por k-medias y k-vecinos más cercanos. Esto se menciona, dado que un sistema de clasificación implica una retroalimentación entre las fases de su diseño dependiendo de los resultados. Uno podría regresar para rediseñar fases anteriores con el fin de mejorar el desempeño en general (S. Theodoridis *et al,.*2009).

Agregado a lo anterior, se hicieron grabaciones según las características del patrón con el fin de tener la base de entrenamiento; así como también se realizaron grabaciones de vectores de prueba que se podrían obtener en tiempo real, para poder medir los resultados. Cabe mencionar que la base de entrenamiento es más grande que la base de prueba (sección 4.2).

Todos los resultados fueron programados y visualizados de forma numérica primeramente en MATLAB.

Tomando cada una de las bases de entrenamiento y prueba fue posible llevar a cabo la clasificación de los datos. A manera de un resumen gráfico, lo que está pasando con el sistema es lo que se muestra en la figura 22.

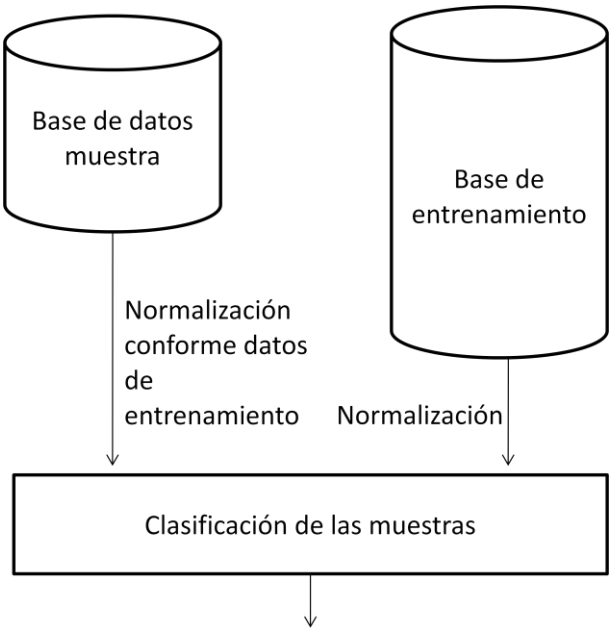


Figura 22. Resumen del sistema para observar el comportamiento de la tasa de detección.

4.1.1 Evaluación

Con el objeto de obtener el mejor vector, se consideró la tasa de verdaderos positivos/falsos positivos que existe con cada uno de estos vectores (mencionados en la sección 3.2).

A continuación, mostramos los porcentajes promedios de detección -entre los cinco gestos propuestos-, en el apéndice (véase apéndice A) se pueden observar de forma detallada las matrices de confusión con las que se llegó a dichos resultados (se presentan de forma similar a como se ilustra un ejemplo en la sección 4.2). En el apéndice se muestra en secciones por el tamaño del vector.

- Utilizando clasificación por k-medias con la distancia euclidiana, la tasa de verdaderos positivos/falsos positivos fue la siguiente:

		Dimensión del vector			
		6D	7D	8D	9D
Normalización	Rango	63.93%/36.07%	38.75%/61.25%	55.13%/44.87%	*82.85%/17.15%*
	μ , σ	72.92%/0.2708%	48.09%/51.91%	61.76%/38.24%	*83.1%/16.9%*

- Usando clasificación por k-vecinos más cercanos con la distancia euclidiana se obtienen los siguientes resultados:

		Dimensión del vector			
		6D	7D	8D	9D
Normalización	Rango	65.11%/34.89%	56.79%/43.21%	83.25%/16.75%	*86.79%/13.21%*
	μ , σ	74.45%/22.55%	76.29%/23.71%	78.3%/21.7%	*79.74%/20.26%*

- Y finalmente, se hicieron las pruebas de la misma manera; con la clasificación por k-vecinos más cercanos, pero utilizando una distancia que se ha propuesto, la cual es multiplicar la distancia euclidiana por la distancia coseno. Las tasas de detección son las siguientes:

		Dimensión del vector			
		6D	7D	8D	9D
Normalización	Rango	77.57%/22.43%	78.18%/21.82%	86.74%/13.26%	***95.12%/4.88%***
	μ , σ	74.71%/25.29%	74.98%/25.02%	*82.12%/17.88%*	78.2%/21.8%

Este análisis se realizó para, primeramente, observar el comportamiento de cada uno de los vectores con todos los gestos que se quieren reconocer. Dada la tasa de detección de cada uno de ellos se observó que el patrón de nueve dimensiones -que previamente se ha mencionado- es el mejor seleccionado. La normalización que mejor se ajusta a los datos de entrenamiento es la que utiliza el rango.

El algoritmo de k-vecinos más cercanos proporcionó la tasa de detección más alta; además de que la multiplicación de la distancia euclidiana por la distancia coseno brindó los mejores resultados.

4.1.2 Análisis estadístico de distancias (métricas)

En este punto de la investigación se eliminaron las distancias que sobrepasaron la distancia máxima establecida, para ello se hizo un análisis estadístico de distancias con algunas pruebas más.

El establecimiento de la distancia máxima se determinó mediante un análisis supervisado con una serie de patrones nuevos, conocidos por cada uno de los gestos de la base de entrenamiento.

El objetivo de este enfoque es que con la distancia máxima se pueda dejar fuera a los puntos que no se acercan a la clase y así clasificar de mejor manera a un nuevo patrón en tiempo real; también se tomo una muestra de valores aleatorios que no deberían pertenecer a ninguna clase.

Adicionalmente, se tomó la distancia de cada uno de los nuevos puntos de prueba a los de entrenamiento para con su clase correspondiente y de toda esa colección de distancias se calculó la media y desviación estándar.

Para la clase Zoom-in, por ejemplo, se tiene una base de dato con *m* elementos que pertenecen a la clase y se generó, de forma supervisada, una base de prueba con *n* elementos correspondiente a la misma clase. Se obtiene la distancia de cada elemento de este grupo de prueba a cada elemento de la base (*m*x*n* distancias en total). Para saber el comportamiento de un elemento que no pertenece a la clase respecto a la misma, se calculó la desviación estándar de la muestra de las distancias. En la figura 23 se ilustra la distribución normal correspondiente a la muestra de distancias del gesto *zoom in*. Para cada uno de los gestos (además de las muestras aleatorias) la distribución normal de las distancias se muestra en la figura 24.

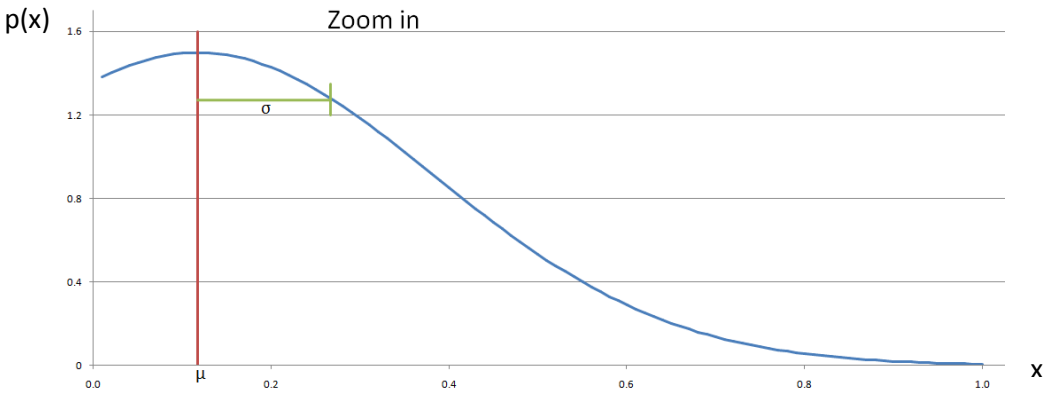


Figura 23. Distribución normal para el gesto *Zoom in*.

En la figura 24 se muestra la distribución para todos las clases de estudio así como para el grupo aleatorio en la que se ilustra el valor de la media y desviación estándar por

clase que mejor se ajusten al sistema de clasificación, de tal forma que incluyan la mayor cantidad de elementos de la clase y lo menos posible de muestras aleatorias (línea punteada).

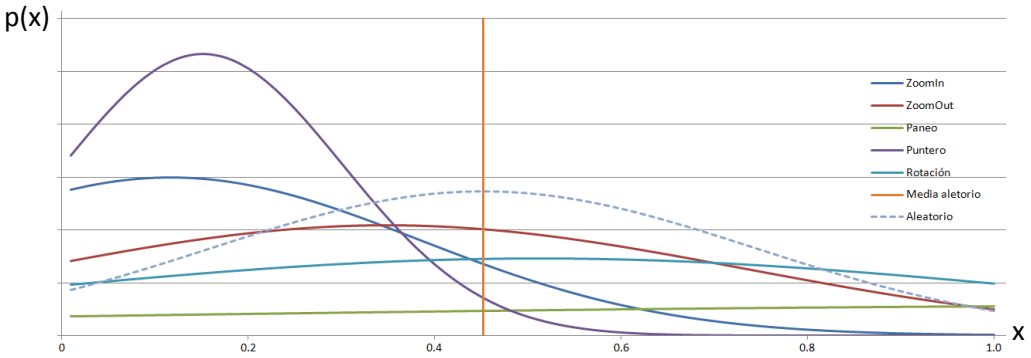


Figura 24. Distribución normalizada por cada uno de los gestos y una muestra de valores aleatorios. La línea recta representa el punto medio de la muestra aleatoria.

Esto representa solo un filtro dentro del algoritmo que plantea esta investigación para el reconocimiento de los gestos. Como se ha mencionado, también se pondera el valor de *k* para este análisis.

4.2 Medición de precisión

Por lo tanto, dada la evaluación, se utilizó el patrón de nueve dimensiones, la normalización por rango y el algoritmo de k-vecinos más cercanos (utilizando la multiplicación de la distancia euclidiana por la distancia coseno).

Finalmente, para observar cómo se comportaba la clasificación, se efectuaron pruebas cambiando la distancia máxima (denotada por la media más una variación -de 0.0 hasta 0.9- de la desviación estándar obtenida por cada gesto), el valor de *k* (con valores igual a 7, 14 y 21) y también la afinidad de los vecinos clasificados, con variaciones que van desde un 50% hasta el 100% del valor de *k*.

Como se comentó anteriormente, se hicieron grabaciones de muestras supervisadas -con historial de cinco cuadros- por cada una de las clases y así visualizar el comportamiento de la clasificación de cada uno de ellos. Para mostrar dichos resultados se tienen 372 patrones, en donde 53 son de zoom in, 43 de zoom out, 75 de paneo, 109 de puntero y 92 de rotación.

Para unos resultados más significativos también se realizó una prueba con gestos aleatorios (ruido) que no deberían de corresponder a ninguna clase. Se hizo una toma de 307 patrones de este tipo.

A fin de visualizar los resultados, se muestran matrices de confusión, con lo cual se obtienen valores de precisión y sensibilidad (*precision and recall*). Estos están definidos en términos de Verdaderos Positivos (VP) (casos correctamente clasificados), Falsos Positivos (FP) (casos incorrectamente clasificados) y Falsos Negativos (FN) (casos sin clasificar).

La precisión está definida como:

$$Precisión = {(VP)}\Big/{(VP + FP)}$$

y la sensibilidad como:

$$Sensibilidad = {(VP)}\Big/{(VP + FN)}$$

ambas métricas van de 0 a 1, en donde una precisión igual a 1 significa que todas los casos detectados fueron correctamente etiquetados y una sensibilidad igual a 1 significa que todos los casos presentados fueron exitosamente detectados y etiquetados (A. Chávez *et al.*, 2011). En la figura 25 se ilustra un ejemplo de lo anterior.

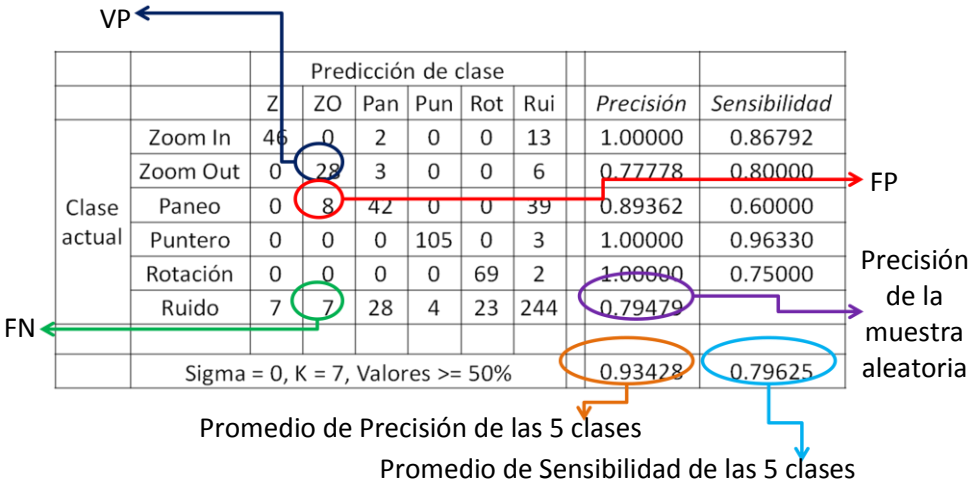


Figura 25. Matriz de confusión para el caso donde la distancia máxima es igual a la media. Se detectan los 7 vecinos más cercanos. Y la afinidad para la clasificación debe ser mayor al 50 por ciento.

Se obtuvieron algunas tablas como la que se muestra en la figura anterior para las diferentes variaciones -antes mencionadas- para que un dato sea clasificado. Solo se tomaron algunas muestras para visualizar los resultados que marcan cómo se comporta la clasificación bajo estas condiciones (figura 26) y así podamos elegir la mejor para el sistema. Cabe mencionar que la sensibilidad no se puede calcular para las muestras de gestos aleatorios puesto que ellos no pertenecen a una ninguna clase (véase apéndice B).

		Distancia máxima									
		μ + 0.0σ	μ + 0.1σ	μ + 0.2σ	μ + 0.3σ	μ + 0.4σ	μ + 0.5σ	μ + 0.6σ	μ + 0.7σ	μ + 0.8σ	μ + 0.9σ
Afinidad	50%										
	75%										
	80%										
	90%										
	100%										

Figura 26. Elección de casos representativos -celda verde-. Se eligieron éstos para evaluar en 7, 14 y 21 vecinos más cercanos.

En total tomamos quince casos, se les evaluó y los resultados se ilustran en la siguiente gráfica (figura 27). Los valores son ordenados por sensibilidad en el eje horizontal y se muestra la precisión en el eje vertical. En la gráfica se pueden observar los resultados, en donde cada uno de los puntos de la línea azul representa el promedio de sensibilidad y precisión de cada uno de los casos. La línea roja representa las muestras aleatorias (a pesar de que la sensibilidad no pueda calcularse para el ruido, se asigna la

sensibilidad del caso que se está tratando). Y la línea verde representa la media que existe entre las dos anteriores.

El modo en que quedan ordenados los casos en la gráfica (figura 27) según su sensibilidad es como sigue: 6, 5, 3, 4, 2, 1, 9, 15, 8, 7, 14, 12, 13, 11 y 10, de izquierda a derecha, respectivamente. En el apéndice B -como se ha comentado- puede observar los detalles de cada uno de los casos.

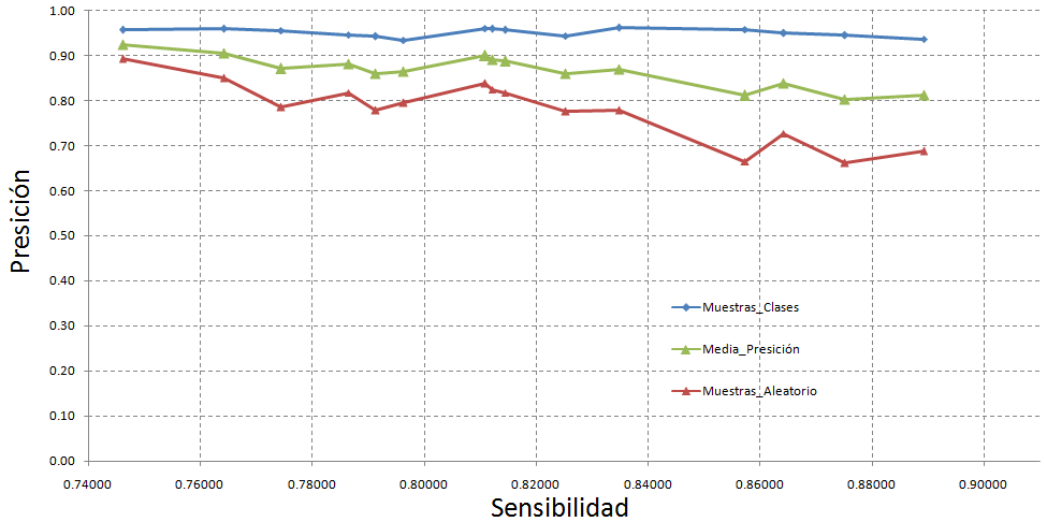


Figura 27. Sensibilidad y precisión del promedio -de clases- de las muestras, tanto de las de prueba -por cada gesto- como la toma aleatoria, además de la media entre estas.

Mientras el punto se encuentre más hacia la derecha, más sensible será y en tanto el punto se encuentre más alto, más preciso será.

Dados estos resultados se observa que el caso 9 (Sigma = 0.4, K = 21, Valores = 80%) se encuentra bien posicionado tanto en valores de precisión como de sensibilidad. Por otro lado, el caso 6 (Sigma = 0, K = 21, Valores = 100%), según las muestras de ruido (gestos aleatorios), mostró el valor más alto de precisión. Ésta es la configuración que se ha elegido para la aplicación del sistema, dado que puede distinguir de mejor manera los gestos que no son los que se quieren reconocer.

Una vez observados los resultados, tomamos las métricas que nos arrojan la mejor tasa de reconocimiento. De esta manera, se programó en Microsoft® Visual C# 2010 un clasificador de gestos en donde se leen los archivos que contienen los datos de

entrenamiento, la normalización de los datos y la clasificación del patrón que se detecta en tiempo real (figura 28).

Figura 28. Implementación del clasificador. Con circunferencias se muestran algunas articulaciones, además de mostrar que un gesto es reconocido (en este caso es el puntero) y los cuadros por segundo.

La aplicación se ejecuta en una laptop Intel® Core™ i5 64bits 2.67GHz y 6GB de memoria RAM. La detección y el reconocimiento de los gestos funciona con el ciclo de video de 27-32 cuadros por segundo (fps, por sus siglas en ingles), aproximadamente.

Capítulo 5. Aplicación en la manipulación de modelos anatómicos

En concordancia con lo planteado en los capítulos anteriores de esta investigación, se establece que el sistema es capaz de proporcionar, en tiempo real, la identificación de cuál es el gesto que ha sido reconocido; ello según el patrón que se genera en tiempo real.

Para mostrar la aplicación de los gestos se utilizó un software de uso libre y de código abierto llamado "*3D Slicer*"; el cual es un paquete para visualización y análisis de imágenes médicas, que es empleado generalmente para aplicaciones médicas en todas sus ramas (figura 29).

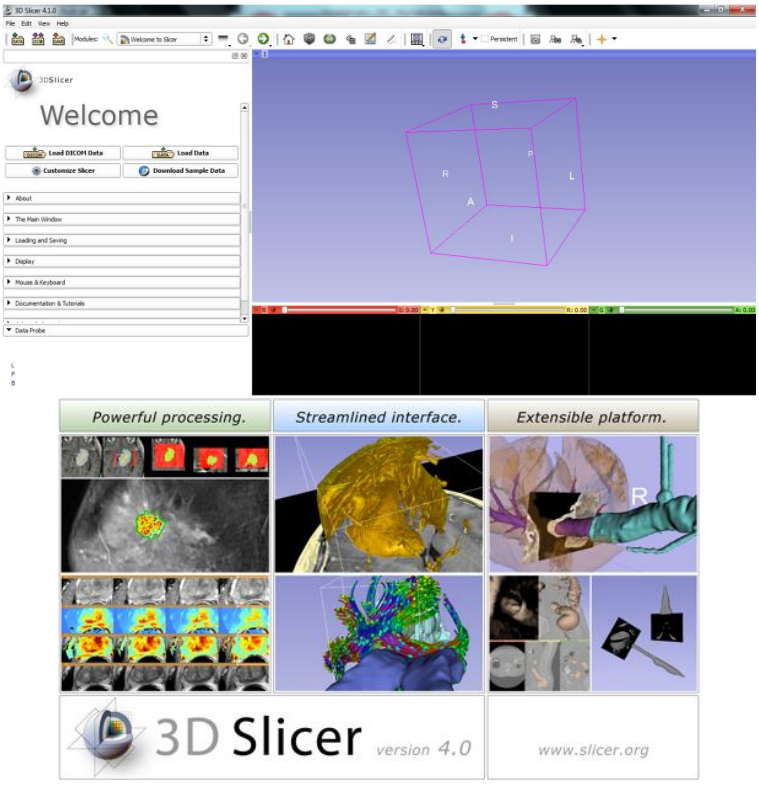


Figura 29. Interfaz de *3D Slicer* e imágenes destacadas del software (www.slicer.org).

De esta manera, en *3D Slicer* – como en muchos de los diferentes programas de desarrollo en 3D – se tienen varias vistas disponibles para observar los modelos desde distintos ángulos.

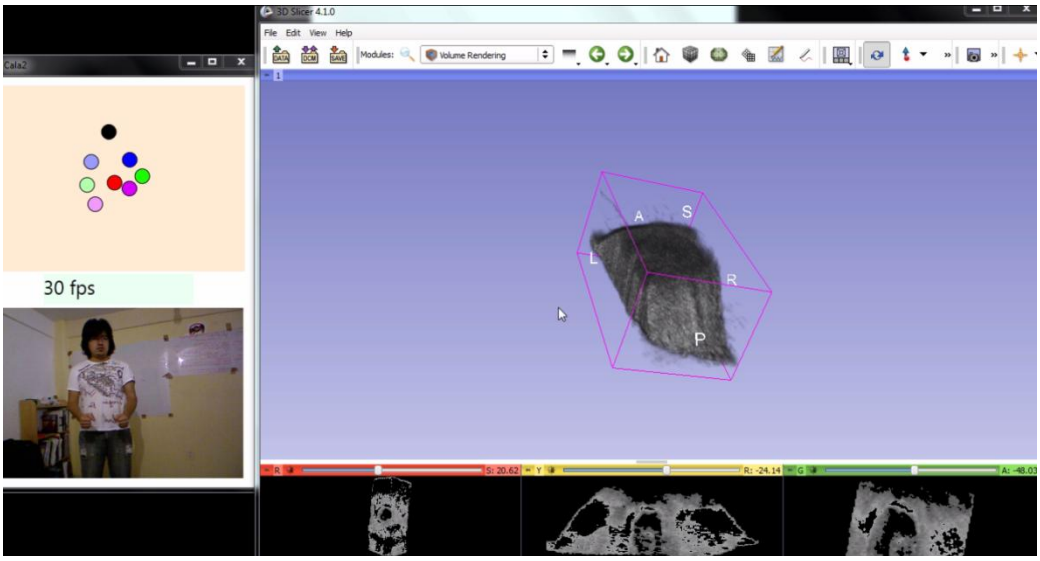
Asimismo, como en diferentes modeladores y programas de visualización de objetos en 3D, la interfaz interactúa con el mouse para cada uno de los gestos, como se describe a continuación.

- *Zoom*: Deslizar el ratón hacia arriba mientras se presiona el clic derecho para hacer *zoom out* y hacia abajo para lograr la interacción de *zoom in*.
- Paneo(traslación): Con el fin de lograr que el sistema pueda modificar la perspectiva de la cámara, se mantiene el clic central presionado y se mueve hacia la posición deseada.
- Rotación: La interacción de rotar el objeto se logra haciendo clic izquierdo mientras se mueve el mouse hacia la posición que el usuario desee.

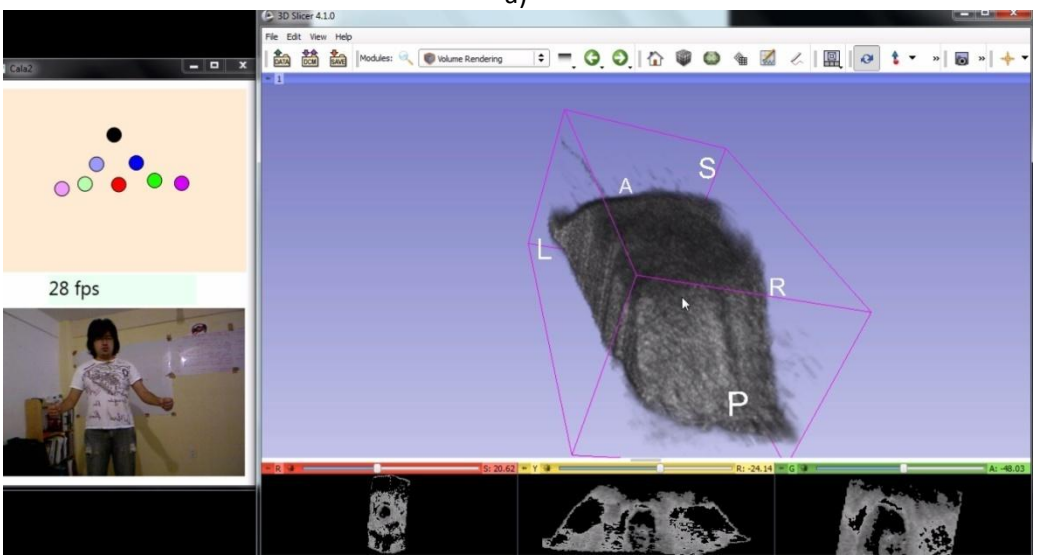
5.1 Driver para mouse

La comunicación entre los gestos reconocidos por el sistema que se ha diseñado y las acciones del mouse se llevan a cabo mediante la utilización de la API de Windows *user32.dll* , con la que se puede llamar a los eventos del ratón. De esta manera, lo que se envía del sistema reconocedor es el gesto que se ha reconocido y alguna posición de las manos. Así, por ejemplo, en el caso del gesto *zoom in* se le envía a la clase la activación del clic izquierdo del mouse y se acciona el mouse a que se mueva hacia abajo.

En la figura 30 se muestra la interacción en el software y a la izquierda - de cada una de las imágenes- se ve el sistema reconocedor de gestos. A continuación se presenta la interacción con el gesto de *zoom in* y la transformación que se efectúa en un modelo anatómico 3D.



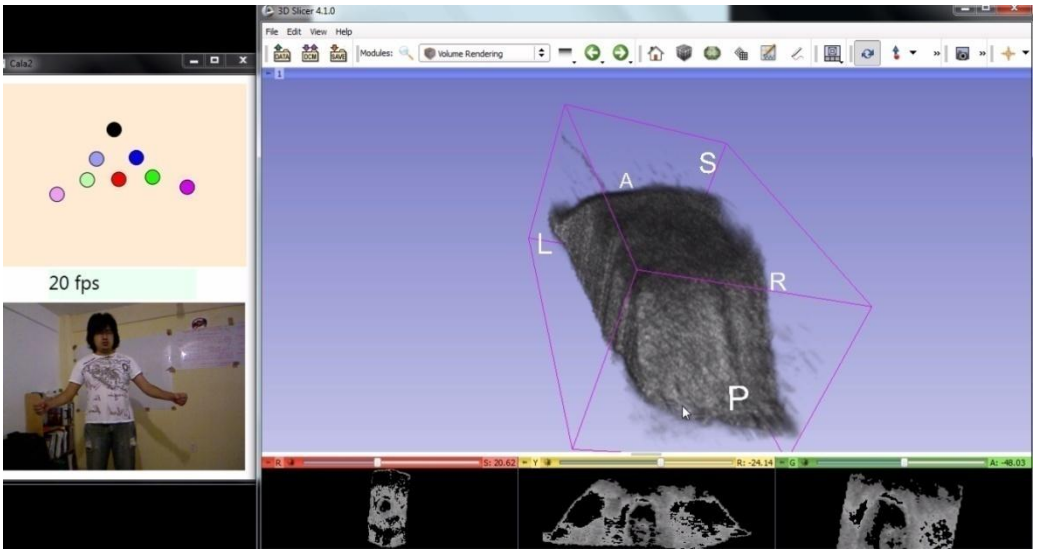
a)



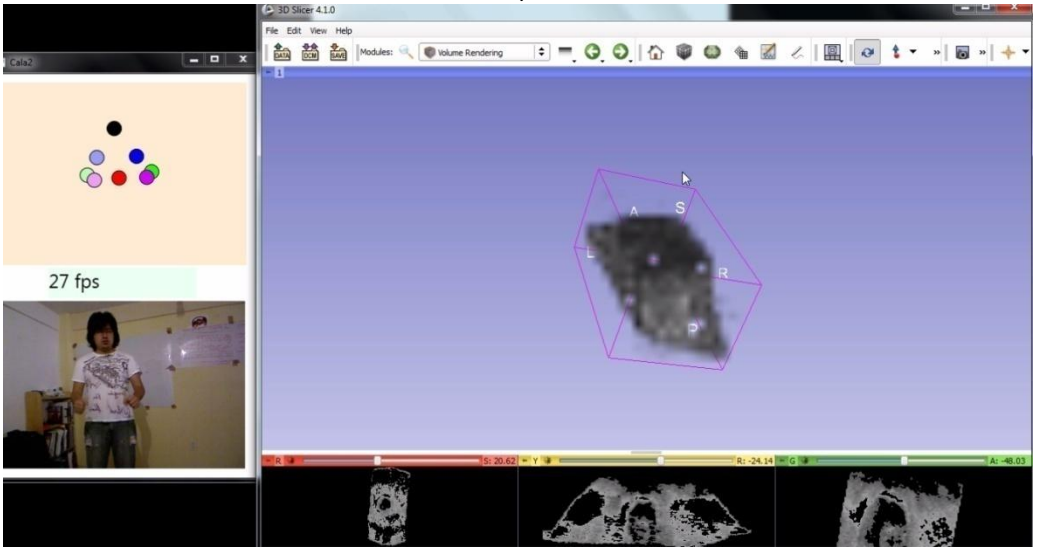
b)

Figura 30. Interfaz de reconocimiento de gestos interactuando con los eventos del mouse en el software *3D Slicer.* Se muestra cómo funciona el gesto y la transformación de *zoom in.* Se observa que el modelo a) crece después de interactuar con el gesto reconocido b).

En la figura 31 se puede observar de igual manera la interacción del sistema reconocedor con el software médico. Se muestra el gesto de *zoom out*. *3D Slicer* primero hace la transformación que uno desea y después vuelve a generar la imagen del modelo.



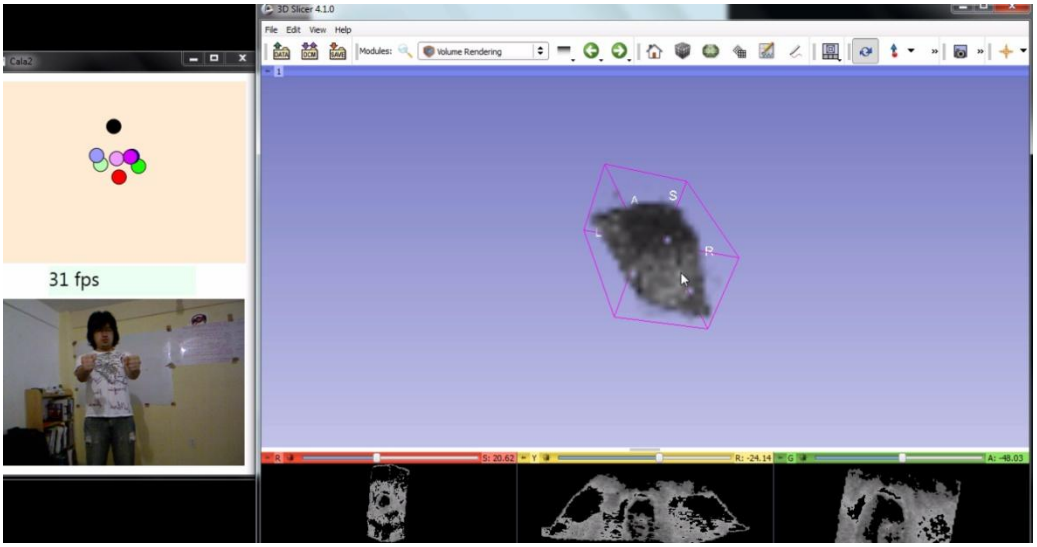
a)



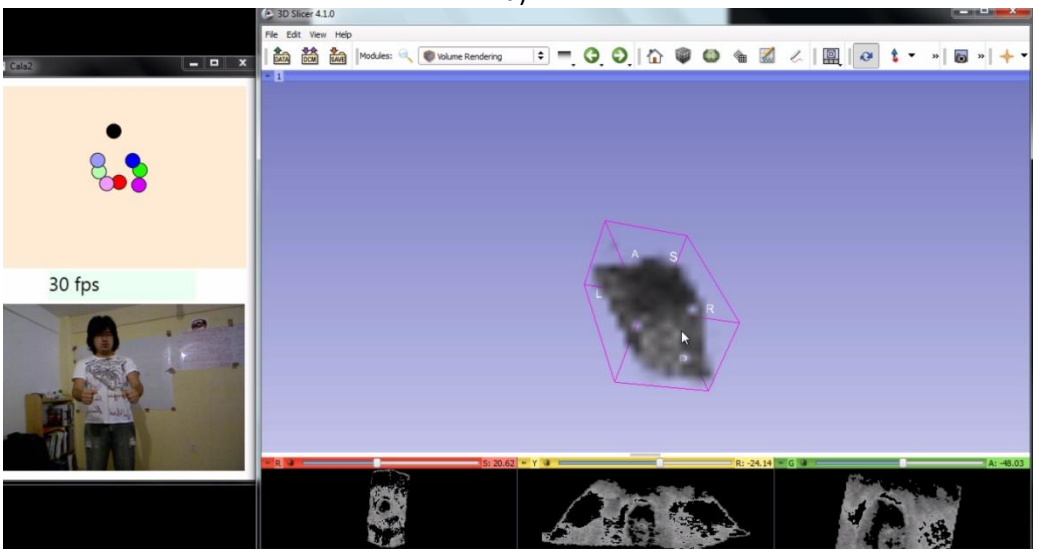
b)

Figura 31. Secuencia de *zoom out* del modelo 3D con la interacción del reconocedor de gestos. El modelo primeramente se encuentra con una dimensión a), después de que es reconocido el gesto de *zoom out* el modelo cambia de tamaño b).

En la figura 32 también mostramos los resultados con el gesto de paneo. El a) modelo se encuentra en un posición, una vez que sucede el reconocimiento se lleva a cabo la b) transformación del objeto.



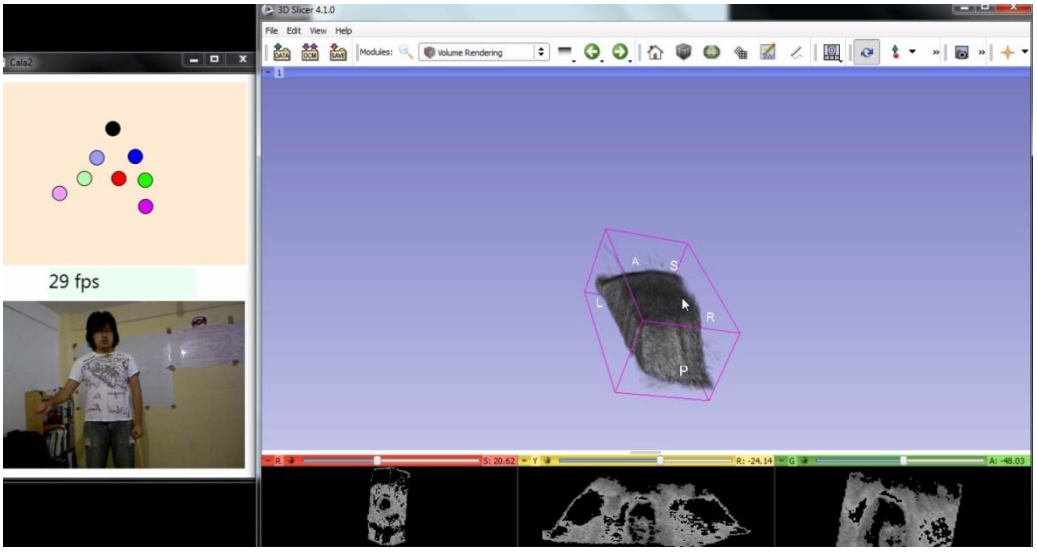
a)



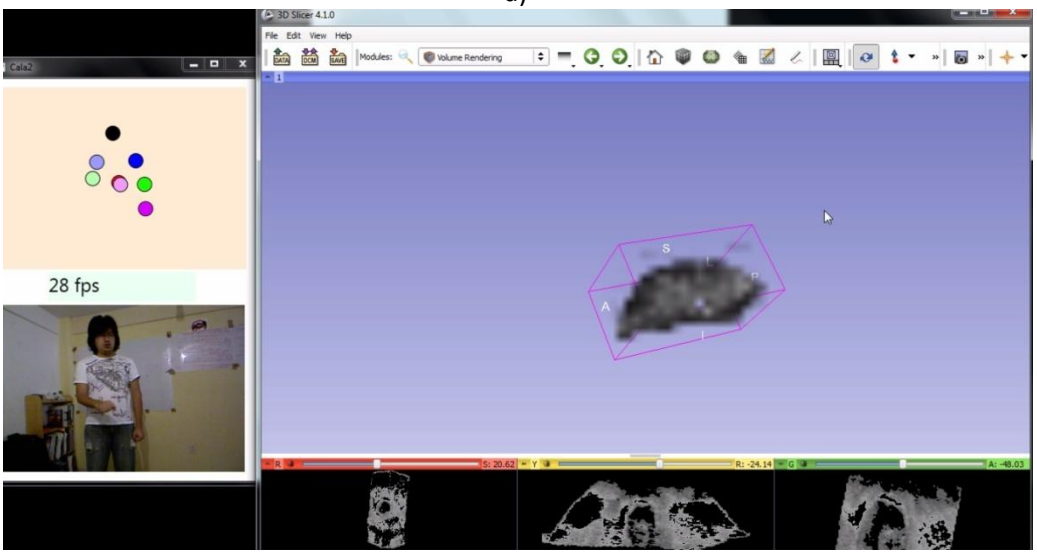
b)

Figura 32. El modelo a) se traslada una vez que es reconocido el gesto -paneo- y la dirección a donde se quiere mover b).

La imagen RGB del mundo real que aparece en el sistema reconocedor es de tipo espejo. Esto quiere decir que nos dará la percepción de que es un reflejo del usuario. Por ello, aunque en la figura 33 parece ser que se interactúa con la mano derecha, en realidad se está interactuando con la mano izquierda, reconociéndose así el gesto de la rotación.



a)



b)

Figura 33. Al reconocer el gesto de rotación, el modelo anatómico a) es transformado sobre su eje Y b).

Como se mencionó, para el reconocimiento de los gestos: la velocidad de los cuadros por segundo no desciende utilizando el equipo de cómputo antes mencionado. Sin embargo, al cargar un modelo 3D y hacer las transformaciones en *3D Slicer* con el reconocimiento de los gestos: la velocidad de los cuadros si decreció, variando de los 20 a los 30 cuadros por segundo aproximadamente.

Se cree que a pesar de que el ciclo de video decrece con la carga del software médico, la apreciación del cambio en el modelo para el usuario es de buena manera y no afecta la percepción de la interacción con el sistema.

Conclusiones

La presente investigación da a conocer un análisis que permite clasificar gestos, que sirven para una interacción multimedia, emulando la comunicación con el mouse basado en un algoritmo de clasificación.

Al mismo tiempo, se presentó la aplicación en un software de uso médico.

Como parte integral de este estudio, se muestran los antecedentes y marco teórico con el fin de abordar los temas que se relacionan con la tesis aquí propuesta; con ello se busca familiarizar y aclarar al lector sobre algunas definiciones.

Una vez que se han dado las bases del conocimiento, prosigue la definición de los gestos que interactúan, el procedimiento de cómo se va formando el patrón con las articulaciones por persona detectada y el clasificador del vector que se crea en tiempo real.

La implementación se fue desarrollando en base a evaluar el vector que arrojara la mejor tasa de detección. El mejor vector fue el que consistía en 9 dimensiones (*ΔxIzq, ΔyIzq, ΔzIzq, deuclXYZ, ΔxDer, ΔyDer, ΔzDer, deltaManoDerAtorso, deltaManoIzqAtorso*). También fue evaluada la normalización de los datos. En base a las pruebas, la mejor para nuestra clasificación de gestos fue la normalización por rango.

Es pertinente mencionar que se hicieron diferentes modificaciones al sistema, con tal de obtener mejores resultados y evitar los falsos positivos que el sistema pudiera arrojar. Lo anterior puso de manifiesto que la distancia propuesta -aquella donde se multiplicó la distancia euclidiana por la distancia coseno- marcó gran diferencia para poder clasificar con el algoritmo de k-vecinos más cercanos y obtener los mejores resultados.

Se expone que el mejor caso para distinguir entre falsos positivos, o de otra manera aquel que identifica de mejor manera a los gestos que no deben ser clasificados, es el

caso nueve; donde se utiliza una distancia máxima igual a la media, una *k* igual a 21 y con afinidad del 100% de los vecinos más cercanos.

Aún cuando el proceso de clasificación es de gran labor para la computadora, ésta presenta un alto rendimiento en los cuadros por segundo (27~32 fps); no obstante, cargando un modelo 3D en el software de *3D Slicer* y ejecutando el reconocedor de gestos el rendimiento de ciclo de video si descendió a 20~30 fps.

La clasificación mostró resultados satisfactorios, sin embargo, durante la aplicación mostrada en el capítulo 5, existen aún falsos positivos considerables en el gesto de rotación que pueden representar un cambio visible para el usuario.

Se concluyeron satisfactoriamente los objetivos propuestos para esta investigación.

Trabajo a futuro

Gracias al poder de las cámaras en profundidad, el desarrollo tecnológico está obteniendo gran importancia en el ámbito científico. Sin embargo, aún queda un amplio campo de investigación por realizar, que permita tener mejores aplicaciones con el uso de la visión computacional.

La investigación desarrollada por esta tesis deja la futura posibilidad de implementar el sistema aquí mostrado dentro de un quirófano o en un laboratorio de análisis radiológico y buscar que la aplicación sea más confiable para el usuario, teniendo así una porcentaje de error que sea cada vez menos perceptible. Además de que se pueda implementar en diferentes sistemas operativos.

Una futura mejora al sistema aquí mostrado podría realizarse en el sentido de agregar más gestos de interacción, para uso específico de la aplicación en la que se desee implementar.

BIBLIOGRAFÍA

A. Baumgart, A. Zoeller, C. Denz, H. Bender, A. Heinzl y E. Badreddin. "Using Computer Simulation in Operating Room Management: Impacts on Process Engineering and Performance". Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS) 2007, pág. 131.

H. Benko y A. Wilson. "DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface". TechReport. IEEE Workshop on Tabletops and Interactive Surfaces 2008. MSR-TR-2009-23.

P. Cinquin, E. Bainville, C. Barbe, E. Bittar, V. Bouchard, L. Bricault, G. Champleboux, M. Chenin, L. Chevalier, Y. Delnondedieu, L. Desbat, V. Dessenne, A. Hamadeh, D. Henry, N. Laieb, S. Lavallee, J. M. Lefebvre, F. Leitner, Y. Menguy, F. Padieu, O. Peria, A. Poyet, E. Promayon, S. Rouault, P. Sautot, J. Troccaz y P. Vassal. "Computer Assisted Medical Intervetions". IEEE Engineering in Medicine and Biology Magazine 1995. Vol. 14, núm. 3, págs. 254-263.

A. Chávez, R. Laganière y P. Payeur. "Vision-Based Detection and Labelling of Multiple Vehicle Parts". 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) 2011, págs. 1273-1278.

T. Collett, M. Collett y R. Wehner. "The guidance of desert ants by extended landmarks". Journal of Experimental Biology 2001. Vol. 204, núm. 9, págs. 1635–1639.

L. Contreras. "Detección y reconocimiento de objetos para aplicaciones en robots móviles empleando técnicas de visión computacional". Tesis de maestría. Universidad Nacional Autónoma de México, Posgrado de Ingeniería 2011. México.

E. Florian. "Tangible Information Displays". PhD thesis. Technische Universität München, Institut für Informatik, 2009. Alemania.

T. Fukushi. "Homing in wood ants, Formica Japonica: Use of the skyline panorama". Journal of Experimental Biology 2001. Vol. 204, núm. 12, págs. 2063–2072.

L. Gallo, A. P. Placitelli, y M. Ciampi. "Controller-free exploration of medical image data: experiencing the Kinect". 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS) 2011, págs. 1–6.

P. Graham, K. Fauria y T. Collett. "The influence of beacon-aiming on the routes of wood ants". The Journal of Experimental Biology 2003. Vol. 206, núm. 3, págs. 535-541.

M. Gutiérrez, F. Vexo y D. Thalmann. "Stepping into Virtual Reality". Springer, 2008, págs. 214. ISBN 1848001169.

O. Hilliges, S. Izadi, A. Wilson, S. Hodges, A. Garcia-Mendoza y A. Butz. "Interactions in the Air: Adding Further Depth to Interactive Tabletops". Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST) 2009, págs. 139-148.

L. Joskowicz y R. Taylor. "Computers in Imaging and Guided Surgery". Computing in Science and Engineering 2001. Vol. 3, núm. 5, págs. 65-72.

K. Kanda, Y. Miyamoto, A. Kondo y M. Oshio. "Monitoring of earthquake induced motions and damage with optical motion tracking". 1st International Conference on Structural Health Monitoring and Intelligent Infrastructure, Smart Materials & Structures 2003. Vol. 14, núm. 3, págs. S32-S38.

I. Korhonen, J. Pärkkä y M. Gils. "Health Monitoring in the Home of the Future". IEEE Engineering in Medicine and Biology Magazine 2003, Vol. 22, núm. 3, págs. 66-73.

F. Lalys, L. Riffaud, D. Bouget y P. Jannin. "A Framework for the Recognition of High-Level Surgical Tasks From Video Images for Cataract Surgieries". IEEE Transactions on Biomed Engineering 2012. Vol. 59, núm. 4, págs. 966 - 976.

J. C. Lee. "Tracking Your Fingers with the Wiimote". Página web johnnylee.net/projects/wii/, 2008, consultado en Julio del 2011.

M. Mahfouz, G. To and M. Kuhn. " Smart instruments: Wireless technology invades the operating room". IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS) 2012, págs. 33 - 36.

F. Marquet, M. Pernot, J. Aubry, M. Tanter, G. Montaldo y M. Fink. "In-vivo non-invasive motion tracking and correction in High Intensity Focused Ultrasound theraphy". 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS) 2006, págs. 688 - 691.

Microsoft®. "Project Natal". Página web www.microsoft.com/presspass/events/ces/docs/ProjectNatal101FS.doc, consultado en Agosto del 2011.

P. Müllier y D. Robert. "A shot in the dark: the silent quest of a free-flying phonotactic fly". Journal of Experimental Biology 2001. Vol. 204, núm. 6, págs. 1039–1052.

D. Nicholson, S. Judd, B. Cartwright y T. Collett. "Learning walks and landmark guidance in wood ants (Formica rufa)". Journal of Experimental Biology 1999. Vol. 202, núm. 13, págs. 1831–1838.

J. Noble. "Programming Interactivity: A Designer's Guide to Processing, Arduino, and openFrameworks". O'Reilly Media, Inc. 2009, págs. 736. ISBN: 978-0-596-15414-1.

Organización Mundial de la Salud. "Una atención más limpia es una atención más segura" Página web http://www.who.int/gpsc/background/es/index.html, consultado en Enero del 2012.

M. Ortega y L. Nigay. "AirMouse: Finger Gesture for 2D and 3D Interaction". Proceedigns Part II of the 12th IFIP TC 13 International Conference on Human-Computer Interaction, Human-Computer Interaction - INTERACT 2009. Vol. 5727, págs. 214-227.

F. Peña, P. Lourenço y J. Lemos. "Modeling the Dynamic Behavior of Masonry Walls as Rigid Blocks". In: Mota Soares *et al*. (eds), III European Conference on computational Mechanics Solids, Structures and Coupled Problems in Engineering 2006.

G. Qian, S. Sural, Y. Gu y S. Pramanik. "Similarity between Euclidean and cosine angle distance for nearest neighbor queries". Proceedings of the ACM symposium on Applied computing (SAC) 2004, págs. 1232 - 1237.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, y A. Blake. "Real-Time Human Pose Recognition in Parts from a Single Depth Image". IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2011, págs. 1297 - 1304.

Y. Sun y X. Li. "Optimizing surgery start times for a single operating room via simulation". Proceedings of the Winter Simulation Conference (WSC)/Conference on Modeling and Analysis for Semiconductor Manufacturing (MASM) 2011, págs. 1325-1332.

S. Theodoridis y K. Koustroumbas. "Pattern Recognition, Fourth Edition". Academic Press 2008. págs. 1-7, 263. ISBN-10: 1597492728

L. Vlaming. "Presenting using Two-Handed Interaction in Open Space". Proceedings of the 3rd Annual IEEE International Workshop on Horizontal Interactive Human-Computed Systems (TABLETOP) 2008, págs. 31-34.

J. Xu, Z. Liu, H. Li, J. Liu, Y. Li. "Study on the Simulation of Outpatient Operating Room Process and Capacity Planning in Terms of Cost Optimization". IEEE 18Th International Conference on Industrial Engineering and Engineering Management (IE&EM) 2011. Vol. Part 3, págs. 1781 - 1785.

Apéndices

A. Matrices de confusión

Seis dimensiones

k-medias, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	13	0	0	0	0
Zoom Out	14	2	0	0	50
Paneo	0	32	125	0	0
Puntero	0	0	0	213	17
Rotación	0	0	0	0	128

k-medias, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	27	0	0	0	27
Zoom Out	0	4	0	0	50
Paneo	0	30	125	0	5
Puntero	0	0	0	213	10
Rotación	0	0	0	0	103

k-vecinos, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	13	0	0	0	0
Zoom Out	14	4	0	0	50
Paneo	0	30	125	0	0
Puntero	0	0	0	213	17
Rotación	0	0	0	0	128

k-vecinos, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	15	0	0	0	0
Zoom Out	12	16	0	0	20
Paneo	0	18	125	0	0
Puntero	0	0	0	213	10
Rotación	0	0	0	0	165

k-vecinos, rango, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	17	0	0	0	0
Zoom Out	10	10	5	0	0
Paneo	0	24	120	0	1
Puntero	0	0	0	213	0
Rotación	0	0	0	0	194

k-vecinos, μ , σ, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	10	0	0	0	0
Zoom Out	17	18	0	0	19
Paneo	0	16	125	0	1
Puntero	0	0	0	213	12
Rotación	0	0	0	0	163

Siete dimensiones

k-medias, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	8	0	0	6	3
Zoom Out	0	0	0	15	33
Paneo	0	33	28	17	17
Puntero	0	0	0	13	33
Rotación	19	1	0	107	109

k-medias, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	27	0	0	21	26
Zoom Out	0	0	0	55	53
Paneo	0	33	28	20	18
Puntero	0	0	0	8	29
Rotación	0	1	0	54	69

k-vecinos, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	25	0	0	21	17
Zoom Out	0	0	0	21	13
Paneo	0	12	28	7	0
Puntero	0	1	0	56	56
Rotación	2	21	0	53	109

k-vecinos, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	27	0	0	3	0
Zoom Out	0	20	0	18	5
Paneo	0	14	28	13	0
Puntero	0	0	0	90	62
Rotación	0	0	0	34	128

k-vecinos, rango, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	25	0	0	7	7
Zoom Out	0	29	0	18	5
Paneo	0	1	28	12	0
Puntero	0	0	0	74	54
Rotación	2	4	0	47	129

k-vecinos, μ , σ, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	25	0	0	0	1
Zoom Out	0	15	0	0	0
Paneo	0	19	28	15	4
Puntero	0	0	0	113	60
Rotación	2	0	0	30	130

Ocho dimensiones

k-medias, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	14	0	0	3	0
Zoom Out	13	0	0	5	51
Paneo	0	34	95	0	0
Puntero	0	0	0	80	19
Rotación	0	0	0	46	125

k-medias, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	27	0	0	8	23
Zoom Out	0	0	0	0	39
Paneo	0	34	95	0	9
Puntero	0	0	0	75	21
Rotación	0	0	0	51	103

k-vecinos, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	21	0	0	3	0
Zoom Out	0	22	0	3	25
Paneo	6	12	95	0	0
Puntero	0	0	0	116	0
Rotación	0	0	0	12	170

k-vecinos, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	16	0	0	0	0
Zoom Out	0	16	0	0	10
Paneo	11	18	95	6	0
Puntero	0	0	0	121	0
Rotación	0	0	0	7	185

k-vecinos, rango, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	20	0	5	1	0
Zoom Out	0	30	7	0	5
Paneo	7	4	83	6	0
Puntero	0	0	0	116	0
Rotación	0	0	0	11	190

k-vecinos, μ , σ, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	17	3	8	1	0
Zoom Out	0	27	7	0	5
Paneo	10	4	80	6	0
Puntero	0	0	0	116	0
Rotación	0	0	0	11	190

Nueve dimensiones

k-medias, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	53	27	9	0	0
Zoom Out	0	13	3	0	0
Paneo	0	0	63	0	0
Puntero	0	0	0	109	0
Rotación	0	3	0	0	92

k-medias, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	53	8	14	0	1
Zoom Out	0	25	9	0	10
Paneo	0	10	52	0	0
Puntero	0	0	0	109	0
Rotación	0	0	0	0	81

k-vecinos, rango, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	45	1	14	0	0
Zoom Out	3	36	7	0	0
Paneo	5	6	49	0	0
Puntero	0	0	1	109	0
Rotación	0	0	4	0	92

k-vecinos, μ , σ, euclidiana

	ZI	ZO	Pan	Pun	Rot
Zoom In	40	0	5	0	0
Zoom Out	0	28	5	0	0
Paneo	0	10	55	0	8
Puntero	13	5	7	109	6
Rotación	0	0	3	0	78

k-vecinos, rango, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	50	0	2	0	0
Zoom Out	0	41	3	0	0
Paneo	3	2	61	0	0
Puntero	0	0	1	109	0
Rotación	0	0	4	0	92

k-vecinos, μ , σ, euclidiana_coseno

	ZI	ZO	Pan	Pun	Rot
Zoom In	44	0	4	0	1
Zoom Out	0	30	4	0	18
Paneo	0	10	58	0	11
Puntero	9	3	5	109	6
Rotación	0	0	4	0	56

B. Tablas de resultados de muestras representativas

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.77778	0.80000
Paneo	0.89362	0.60000
Puntero	1.00000	0.96330
Rotación	1.00000	0.75000
Ruido	0.79479	
Dmax=μ+0.0σ, K = 7, Valores >= 50%		
Caso 1	0.93428	0.79625

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.77778	0.80000
Paneo	0.93182	0.58571
Puntero	1.00000	0.96330
Rotación	1.00000	0.73913
Ruido	0.77850	
Dmax=μ+0.0σ, K = 14, Valores >= 50%		
Caso 2	0.94192	0.79121

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.82353	0.75676
Paneo	0.95238	0.55556
Puntero	1.00000	0.96330
Rotación	1.00000	0.72826
Ruido	0.78502	
Dmax=μ+0.0σ, K = 21, Valores >= 50%		
Caso 3	0.95518	0.77436

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.77778	0.80000
Paneo	0.95349	0.56164
Puntero	1.00000	0.96330
Rotación	1.00000	0.73913
Ruido	0.81759	
Dmax=μ+0.0σ, K = 7, Valores = 100%		
Caso 4	0.94625	0.78640

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.81818	0.72973
Paneo	0.97561	0.54054
Puntero	1.00000	0.95413
Rotación	1.00000	0.72826
Ruido	0.85016	
Dmax=μ+0.0σ, K = 14, Valores = 100%		
Caso 5	0.95876	0.76412

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.80645	0.67568
Paneo	0.97436	0.51351
Puntero	1.00000	0.94495
Rotación	1.00000	0.72826
Ruido	0.89251	
Dmax=μ+0.0σ, K = 21, Valores = 100%		
Caso 6	0.95616	0.74607

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.90566
Zoom Out	0.77778	0.80000
Paneo	0.94000	0.65278
Puntero	1.00000	0.96330
Rotación	1.00000	0.80435
Ruido	0.77524	
Dmax=μ+0.4σ, K = 7, Valores >= 80%		
Caso 7	0.94356	0.82522

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.90566
Zoom Out	0.82353	0.75676
Paneo	0.95918	0.65278
Puntero	1.00000	0.96330
Rotación	1.00000	0.79348
Ruido	0.81759	
Dmax=μ+0.4σ, K = 14, Valores >= 80%		
Caso 8	0.95654	0.81440

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.90566
Zoom Out	0.82353	0.75676
Paneo	0.97917	0.63514
Puntero	1.00000	0.96330
Rotación	1.00000	0.79348
Ruido	0.83713	
Dmax=μ+0.4σ, K = 21, Valores >= 80%		
Caso 9	0.96054	0.81087

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.94118
Zoom Out	0.78378	0.82857
Paneo	0.89655	0.80000
Puntero	1.00000	0.96330
Rotación	1.00000	0.91304
Ruido	0.68730	
Dmax=μ+0.9σ, K = 7, Valores >= 50%		
Caso 10	0.93607	0.88922

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.92308
Zoom Out	0.78378	0.82857
Paneo	0.94340	0.76923
Puntero	1.00000	0.96330
Rotación	1.00000	0.89130
Ruido	0.66124	
Dmax=μ+0.9σ, K = 14, Valores >= 50%		
Caso 11	0.94544	0.87510

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.92308
Zoom Out	0.82857	0.78378
Paneo	0.96154	0.72464
Puntero	1.00000	0.96330
Rotación	1.00000	0.89130
Ruido	0.66450	
Dmax=μ+0.9σ, K = 21, Valores >= 50%		
Caso 12	0.95802	0.85722

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.92308
Zoom Out	0.78378	0.82857
Paneo	0.96154	0.71429
Puntero	1.00000	0.96330
Rotación	1.00000	0.89130
Ruido	0.72638	
Dmax=μ+0.9σ, K = 7, Valores = 100%		
Caso 13	0.94906	0.86411

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.82353	0.75676
Paneo	0.98039	0.69444
Puntero	1.00000	0.96330
Rotación	1.00000	0.89130
Ruido	0.77850	
Dmax=μ+0.9σ, K = 14, Valores = 100%		
Caso 14	0.96078	0.83475

	Precisión	*Sensibilidad*
Zoom In	1.00000	0.86792
Zoom Out	0.81250	0.70270
Paneo	0.97917	0.63514
Puntero	1.00000	0.96330
Rotación	1.00000	0.89130
Ruido	0.82410	
Dmax=μ+0.9σ, K = 21, Valores = 100%		
Caso 15	0.95833	0.81207