

GERONTOLOGICAL MOBILITY SENTINEL (GeMob Sentinel)

Iván Daniel Guerra Moreno

Tutor: Francisco Mario Hernández Tejera

Instituto Universitario de Sistemas Inteligentes y Aplicaciones
Numéricas en Ingeniería

SIANI

SISTEMAS INTELIGENTES Y APLICACIONES NUMERICAS EN
INGENIERIA

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

2011/2012

Contenido

Contenido.....	2
1. Resumen	3
2. Planteamiento del problema	4
3. Justificación.....	7
4. Antecedentes	8
5. Objetivos.....	10
5.1. Objetivo general.....	10
5.2. Objetivos específicos	10
6. Metodologia.....	11
8. Limitaciones	12
7. Recursos.....	14
8. Elementos para el diseño de la solucion.....	15
kinect.....	15
weka.....	20
árboles de decisión (tree).....	21
algoritmos basados en instancias (lazy)	21
paradigma metaclasificadores	22
dtw. Alineamiento temporal dinámico (dtw)	23
9. Desarrollo	27
10. Conclusiones	38
11. Lineas futuras	39
notas y referencias	40

1. RESUMEN

Se expondrá un estudio e implementación de una solución para la detección y registro de diferentes eventos de movilidad. Dicha detección está basada en el seguimiento de una secuencia de esqueletos virtuales de un sujeto en un recinto determinado y creado a partir de video RGB y Profundidad. Los eventos pueden ser por ejemplo la secuencia de fotogramas que crean la acción de sentarse, caerse o caminar. Cada esqueleto está definido como la unión de distintos nodos y estos nodos característicos son creados a partir de los datos de profundidad y color. La selección de algunos de los nodos entre el total de nodos se utiliza como atributos para el posterior estudio mediante el uso del software de minería de datos weka. Weka permitirá un análisis de los datos para la elección del mejor clasificador y selección de las principales características. Una vez determinado el algoritmo de clasificación mas adecuado al problema se ha implementado en C#.

2. PLANTEAMIENTO DEL PROBLEMA

La población mundial está envejeciendo a pasos acelerados. Entre 2000 y 2050, la proporción de los habitantes del planeta mayores de 60 años se duplicará, pasando del 11% al 22%. En números absolutos, este grupo de edad pasará de 605 millones a 2000 millones en el transcurso de medio siglo.

La necesidad de asistencia a largo plazo está aumentando. Se pronostica que de aquí al año 2050 la cantidad de ancianos que no pueden valerse por sí mismos se multiplicará por cuatro en los países en desarrollo. Muchos ancianos de edad muy avanzada pierden la capacidad de vivir independientemente porque padecen limitaciones de la movilidad, fragilidad u otros problemas físicos o mentales. Muchos necesitan alguna forma de asistencia a largo plazo, que puede consistir en cuidados domiciliarios o comunitarios y ayuda para la vida cotidiana.

En la zona metropolitana de Madrid por ejemplo viven 616.319 personas mayores de 65 años. El 22% de ellas residen solas incrementando dicho porcentaje hasta el 31% personas mayores de 80 Años. Más de 130.000 mayores reciben el servicio de teleasistencia regional, un dispositivo que llevan los ancianos y que les tiene permanentemente conectados a un centro de coordinación. El servicio de ayuda a domicilio, con trabajadores que ayudan en tareas del hogar y aseo, tiene 80.000 usuarios en la región¹.

Según datos de la OMS las caídas son la segunda causa mundial de muerte por lesiones accidentales o no intencionales. Se calcula que anualmente mueren en todo el mundo unas 424.000 personas debido a caídas, y más de un 80% de esas muertes se registran en países de bajos y medianos ingresos. Los mayores de 65 años son quienes sufren más caídas mortales. Cada año se producen 37,3 millones de caídas cuya gravedad requiere atención médica.

Las caídas se definen como acontecimientos involuntarios que hacen perder el equilibrio y dar con el cuerpo en tierra u otra superficie firme que lo detenga. Las lesiones relacionadas con las caídas pueden ser mortales, aunque la mayoría de ellas no lo son.²

Las lesiones relacionadas con las caídas tienen un costo económico considerable. El costo medio para el sistema sanitario por cada lesión relacionada con caídas en mayores de 65 años es de US\$ 3611 en Finlandia y US\$ 1049 en Australia. Los datos procedentes del Canadá indican que la aplicación de estrategias preventivas eficaces y la consiguiente reducción de las caídas de los menores de 10 años en un 20% supondrían un ahorro neto de más de US\$ 120 millones al año. En los Estados Unidos de América un 20 a 30% de las personas mayores que se caen sufren lesiones moderadas o graves, tales como hematomas, fracturas de cadera o traumatismos craneoencefálicos. La magnitud del riesgo puede deberse, al menos en parte, a los trastornos físicos, sensoriales y cognitivos relacionados con el envejecimiento, así como a la falta de adaptación del entorno a las necesidades de la población de edad avanzada.³

Basándonos en estos hechos se puede concluir que es necesario establecer una serie de protocolos, objetivos de investigación y actuación para la geriatría y la gerontología. Para ello se deberían detectar los diez problemas básicos.⁴

-
1. Deterioro de las funciones cognitivas del anciano y problemas de comportamiento que de él se derivan. Algunas funciones motoras empiezan a deteriorarse mucho antes de la aparición de la enfermedad. Estar atento a estas señales mejora considerablemente la capacidad de diagnóstico.
 2. Depresión. La depresión puede ser identificada como uno de los principales males a seguir de manera sistemática en la población anciana.
 3. Movilidad. La geriatría empieza a observar la movilidad como una herramienta de diagnóstico que debe tenerse en cuenta. El deterioro en la velocidad de desplazamientos y reacciones del paciente es una señal de alarma de que si está produciendo un declive general.
 4. Nutrición. Entre los adultos mayores se producen cambios en los patrones nutricionales que, en algunas ocasiones, producen graves deterioros del estado físico. El descenso en la cantidad de comida ingerida y, sobre todo, la pérdida del hábito de picar entre horas generan una merma considerable en la cantidad de nutrientes.
 5. Hormonas. Una de las consecuencias mejor conocidas del paso del tiempo, sobre todo en las mujeres, es el cambio en el patrón hormonal. En teoría, el aporte extra de determinadas hormonas podría ser una buena estrategia para combatir la vejez. Pero se sabe que algunas terapias sustitutivas producen severos efectos secundarios.
 6. Fragilidad. Se trata de un importante precursor de la incapacidad funcional. Las causas de la fragilidad son demasiado numerosas incluyendo desde deterioros cognitivos hasta diabetes o problemas vasculares. La intervención ante este mal se centra en dos frentes: prevenir mediante el ejercicio físico y detectar síntomas precoces, como el aumento de los episodios de caídas.
 7. Corazón. Casi el 50 por ciento de las personas de avanzada edad muestran algún tipo de deterioro en sus funciones cardíacas por lo que la vigilancia del corazón y de la presión arterial debe ser algo rutinario.
 8. Sistema inmune. El deterioro del sistema inmune con la edad es bien conocido. Una de las causas de este mal es la disminución de aporte proteínico de la dieta.
 9. Vida a los años. La medicina no busca la longevidad banal, sino la mejora de la calidad de vida de los adultos mayores.
 10. Sistema sanitario. El último desafío de la geriatría consistiría en dotar de una infraestructura que permitiera generar programas de seguimiento de pacientes a largo plazo.

La solución parcial, que no definitiva, podría estar basada en las nuevas tecnologías y sistemas inteligentes. Un sistema capaz de detectar y reconocer patrones de conductas del movimiento, así como de actuar inteligentemente frente a situaciones aprendidas previamente. El sistema captador de información se basa en el sensor kinect⁵ de Microsoft, esta información de imagen y profundidad (RGB+deep) posibilita la creación de un esqueleto virtual basado en una serie de nodos o puntos de referencia y sus pertinentes

interconexiones. Cada coordenada XYZ de cada nodo durante un tiempo t posibilita el entrenamiento de un clasificador el cual será el encargado de detectar en una primera versión del sistema las posibles caídas del sujeto en estudio.

3. JUSTIFICACIÓN.

Primeramente destacar que se trata de un sistema no invasivo, actualmente existen multitud de sistemas de aviso y emergencias como son los colgantes y pulseras electrónicos los cuales a pesar de cumplir su misión correctamente dependen en gran medida de las capacidades cognitivas y memorísticas del usuario, tratándose de un nuevo complemento se tiende a olvidar en cualquier cajón o después de una ducha. Las pulseras añaden el problema del roce con pieles especialmente delicadas lo que puede originar ampollas y molestias. El sistema puede situarse en cuatro cuartos distintos y monitorizar los patrones de movimiento de una forma totalmente externa y sin necesidad de grabación de la imagen y no afectando de esta forma a la intimidad ni privacidad del sujeto en cuestión.

Desde el punto de vista económico es un sistema relativamente barato dado el bajo coste del equipo necesario y al poco o nulo mantenimiento necesario, sobre todo teniendo en cuenta como referencia el ahorro en teleasistencia, trabajadores a domicilio, desplazamiento por falsas alarmas y agravamiento de lesiones debido a caídas no detectadas ni avisadas. Desde el momento de una caída por rotura de cadera hasta su desplazamiento a un recinto hospitalario puede pasar de simples horas a varios días dependiendo del intervalo de visitas que tenga el anciano/a.

Con respecto a los diez puntos básicos enumerados anteriormente el sistema podría ser capaz de detectar al menos los puntos 1, 3, 4, 6, 9 y 10. Deterioro de las funciones cognitivas del anciano y problemas de comportamiento que de él se derivan. Detección de deterioro de funciones motoras debido a un incremento de caídas o tropiezos. Así mismo el deterioro en la velocidad de desplazamientos y reacciones del paciente es una señal de alarma de que se está produciendo un declive general. Posibilidad de detección de cambios en los patrones nutricionales y horarios de comida. El punto seis es la Fragilidad directamente relacionada con el ejercicio físico, el sistema podría ser ampliado mediante un sistema de monitorización y entrenamiento físico que permita detectar síntomas precoces. El sistema basado en Kinect puede mejorar la calidad de vida de los adultos mayores al crear una impresión de seguridad alejando los típicos temores que acompañan a la soledad por ultimo se dotaría de una infraestructura que permite generar programas de seguimiento de pacientes a largo plazo.

4. ANTECEDENTES

La problemática aquí tratada ha sido caso de estudio con multitud de posibles soluciones al mismo. Algunas de estas soluciones, ejemplos y antecedentes para este fin se enumeran a continuación.

- Sistemas basados en Redes Neuronales y acelerómetros.⁶

Basado en dispositivos detectores de caídas inteligentes y portátiles sobre tecnología de redes neuronales artificiales consistente básicamente en dos componentes, acelerómetros triaxiales de medidas continuas y microprocesador encargado del reconocimiento. El dispositivo se integra en una pequeña caja para sujetar el cinturón del anciano/a.

- Transformada de Fourier basados en acelerómetros y micrófonos.⁷

Sensores acelerómetros y micrófonos son conectados al cuerpo del paciente y transmiten sus sonidos y movimientos mediante redes inalámbricas hasta la unidad de monitorizado. Aplica Short Time Fourier Transform (STFT) y análisis espectral del sonido para la detección de la posible caída mientras que la clasificación se basa en máquinas vectoriales.

- Análisis de Umbral basado en sensores de presión⁸

Dispositivo portátil basado en un sensor de presión colocado en la plantilla del zapato encargada de recopilar la información del movimiento y la presión sobre el suelo. Combina el análisis de umbral con algoritmos de máquinas de vectores.

- Algoritmos evolutivos multiobjetivos basados en visión artificial⁹

El sistema extrae la información 3D a partir de varias cámaras y realiza el seguimiento en 3D de las personas en el espacio. Basado en dos algoritmos evolutivos multiobjetivos para la estimación continua de la posición 3D en tiempo real, un primer algoritmo sobre nube de puntos 3D y un segundo sobre la base de prismas 3D. Ambos algoritmos utilizan aprendizaje de aspecto visual, color y movimiento como características.

- Reglas de evaluación basados en sensores corporales, cámara cenital y array de micrófonos.¹⁰

El sistema propone el uso de datos de vídeo, audio y movimiento capturados del cuerpo del paciente utilizando sensores acoplados al cuerpo y en su entorno mediante el uso de cámaras aéreas y array de micrófonos. El seguimiento lo realiza el componente de la percepción visual que permite el reconocimiento de la trayectoria del sujeto, mientras que el procesamiento de datos de audio y análisis de direccionalidad del sonido identifica la caída e indicar un evento de emergencia. El comportamiento postcaída y el movimiento del sujeto indica la gravedad de la caída. Mediante una representación semántica de la condición del paciente, el contexto y la evaluación basada en normas se realiza la clasificación.

- Lógica difusa basada en múltiples cámaras RGB¹¹

Varias cámaras se utilizan para ver el mismo entorno y el mundo se cuantifica en elementos

de volumen no superpuestos (voxels). Se construye una representación 3-D del sujeto en tiempo real basado en su silueta, denominada voxel. Las características se extraen del voxel, mientras que el razonamiento y clasificación se basa en lógica difusa para determinar el grado de pertenencia a un número predeterminado de estados en cada fotograma. El sistema automático es comparado con los datos de aprendizaje previamente suministrados por un entrenador humano

- K-Vecinos mas cercanos basado en cámara RGB¹²

Método para la detección y registro de diferentes eventos de interés basados en la postura del sujeto. Estos eventos incluyen en pie, sentarse, inclinarse / agacharse, acostado de lado y acostado hacia la cámara. Los histogramas de la proyección de la silueta segmentada del cuerpo humano se utilizan como vector principal característico para la clasificación de la postura. La deducción de la misma utiliza algoritmo de k-vecinos más cercanos (k-NN) y algoritmos de acumulación de evidencias. Por otra parte, se utiliza la velocidad de la caída para diferenciar entre caída real y un evento donde la persona está simplemente tumbada sin caer.

5. OBJETIVOS

5.1. Objetivo General

El objetivo general global es la detección de un individuo caído o accidentado en el suelo en cualquier zona del recinto en estudio. Basado en la posición de su esqueleto virtual, la velocidad de caída del sujeto y el tiempo de incorporación si existiera.

5.2. Objetivos Específicos

- Detección de un individuo en el área de estudio.
- Creación de la estructura de esqueleto virtual y extracción de características basado en los nodos del esqueleto.
- Exportación e importación de los datos en formato csv adaptado a la estructura de archivo legible por el software Weka, así como Matlab o Excel.
- Identificación del tipo de movimiento según patrones de movilidad distintos, gestos. Ej.: Tendido, sentado, de pie parado, caminando, caído. Otros patrones de movimiento como son la activación de entrenamiento físico, acciones como comer, leer, y demás actividades pueden ser añadidas en versiones posteriores. (Ver apartado 8 Limitaciones)

6. METODOLOGIA

Para la consecución de los objetivos a continuación se detallan procedimientos, técnicas, instrumentos, actividades, etapas y/o demás estrategias metodológicas requeridas. El diseño metodológico será la base de la planificación de todas las actividades que demandarán el proyecto.

La metodología puede ser desglosada según el esquema siguiente:

1. Configuración y ajuste de la plataforma de desarrollo.
 - a. Plataforma de desarrollo Kinect
 1. Instalación Visual Studio 2010
 2. Instalación Prerrequisitos SDK Kinect
 3. Instalación Kinect Drivers y SDK
 4. Prueba del sistema
 - b. Plataforma de Minería de datos Weka
 1. Instalación Prerrequisitos Weka
 2. Instalación Weka
 3. Prueba del sistema
2. Configuración y ajuste del sensor dentro del área en estudio.
3. Detección de un individuo en el área de estudio.
4. Creación de la estructura de esqueleto virtual.
5. Extracción de características. Cada nodo estará caracterizado por tres componentes espaciales XYZ y un componente temporal τ . ($0 < \tau < T$; Siendo T el tiempo de duración del evento)
6. Aprendizaje de los tipos de movimiento
7. Identificación del tipo de movimiento

8. LIMITACIONES

El trabajo a realizar presenta una serie de limitaciones enumeradas a continuación:

El escenario de estudio donde se encuentra el sujeto tendrá limitada su extensión al alcance del sensor kinect a fecha actual¹³ (4 m).

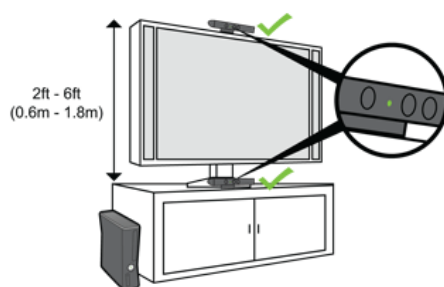
Así mismo el estudio se realizará sobre el supuesto de existir un solo sujeto dentro del área de alcance del sensor.

El sistema se diseñará para su funcionamiento con una sola cámara Kinect lo que limita el estudio a un solo recinto.

Inicialmente no se tendrá en cuenta factores o herramientas de ayuda a la movilidad como pueden ser sillas de ruedas, muletas, bastones o cualquier otro utensilio. Estos factores pueden ser tratados como trabajos futuros.

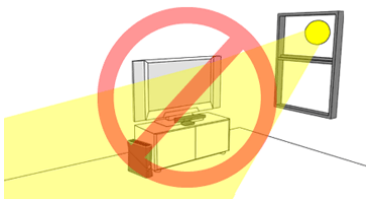
Inicialmente solo se identificarán cinco patrones de movilidad distintos, tendido, sentado, de pie, caminando, caído. Siendo el número máximo de gestos almacenables de 64 debido a restricciones propias de los arraylist de c#.

Dadas las características propias del sensor, resolución en infrarrojo IR, la posición Tendido no es implementable.



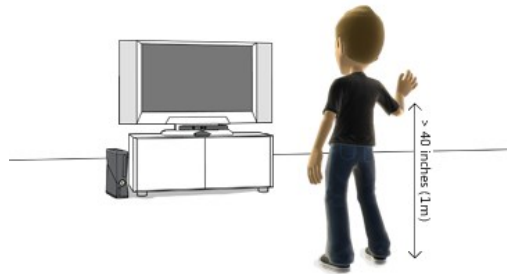
Según las recomendaciones del fabricante el sensor debe estar colocado a una distancia comprendida entre 0,6 a 1,8 m del suelo, sin que haya nada entre el objetivo y el sensor. Dicha configuración imposibilita el control de un recinto completo por lo que se ha optado por la instalación del mismo a 2,5 metros en una de las esquinas del cuarto.

El sensor no puede ser colocado detrás de un cristal.



La iluminación de la habitación debe ser la adecuada para que se vea claramente la cara del sujeto en estudio. Se debe intentar reducir la iluminación lateral o de fondo, especialmente

desde ventana. Para obtener los mejores resultados de seguimiento, evitar colocar el sensor o al sujeto donde reciban la luz directa del sol. Dichas condiciones eliminan la posibilidad de uso en exteriores por saturación de componente de luz infrarroja. Las condiciones de iluminación pueden hacer que Kinect no le identifique o que no realice un seguimiento de sus movimientos correctamente.



Los objetivos deben medir 1 m como mínimo. Por lo que no es capaz de detectar niños de corta estatura.

Algunas prendas de color negro pueden absorber la luz infrarroja dificultando que Kinect realice el seguimiento del jugador correctamente.

7. RECURSOS

Para la realización del sistema se necesitará de una serie de recursos enumerados a continuación:

- Sensor Microsoft Kinect para Windows con cable USB/power correspondiente.
- Ordenador personal o portátil con al menos procesador de doble núcleo a 2,66 GHz y 2Gbytes de memoria RAM
- Sistema Operativo Windows7 o Windows8
- Microsoft Visual Studio 2010 Express u otra edición de Visual Studio 2010
- .NET Framework 4.0
- Microsoft DirectX 9
- Microsoft DirectX R SDK June 2010 o posterior
- The Kinect for Windows SDK 1.0.3.191 (May 2, 2012)
- WEKA 3.6.6 de 64 bit y Java VM 1.6, plataforma de software para aprendizaje automático y minería de datos. GNU-GPL.

8. ELEMENTOS PARA EL DISEÑO DE LA SOLUCION

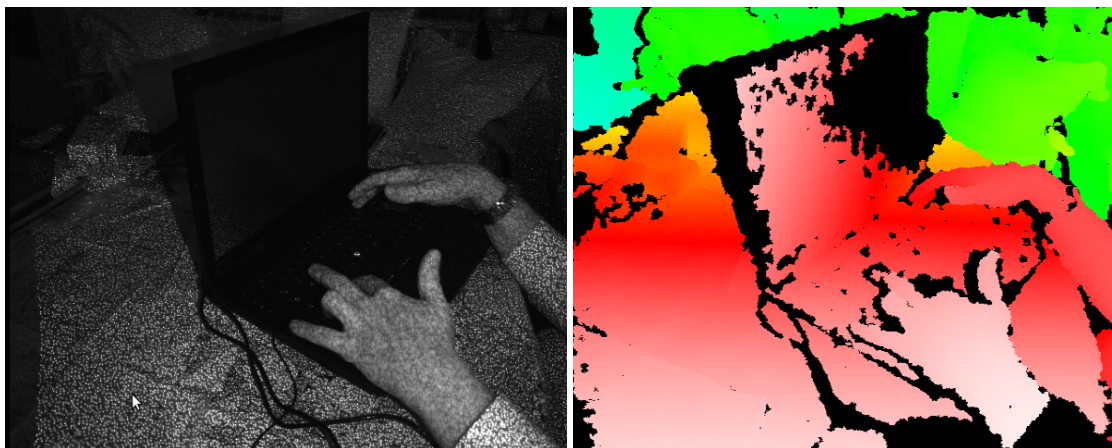
Kinect

La cámara Microsoft Kinect, inicialmente conocida por proyecto Natal, utiliza tecnología desarrollada por la empresa israelí PrimeSense, el sistema de escaneo Light Coding, capaz de interceptar la información 3D de la escena mediante la proyección continua de luz infrarroja.

El equipo consiste en una cámara RGB, una cámara IR y un emisor de Matriz IR para la estimación de la profundidad. Además posee un array de micrófonos para la localización mediante sonidos y reconocimiento del habla.

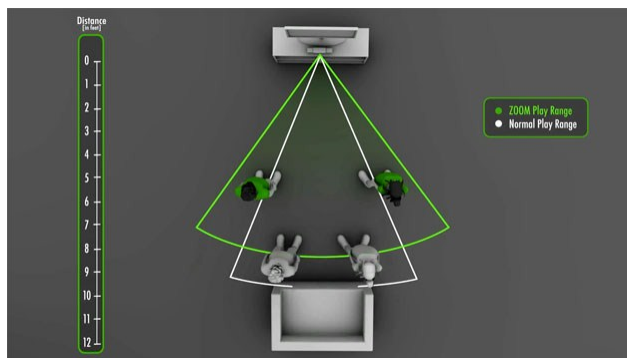


El sensor de profundidad consiste en un proyector de láser infrarrojo combinado con un sensor de imagen infrarrojo CMOS, que permite mediante la distancia de separación de ambos instrumentos y su ángulo de incidencia, calcular la información del entorno en tres dimensiones.



El sensor de profundidad presenta un rango de sensibilidad de entre 0,7 y 5 metros, utilizando una escala de grises de 11 bits -2048 niveles de intensidad desde el blanco (cerca) al negro (lejos).

El rango de visibilidad ronda los 57° horizontales y 43° verticales. La cámara está montada sobre un eje motorizado capaz de pivotar 27° tanto arriba como abajo.



La cámara color envía un fotogramas (frames) de vídeo a resolución VGA (640x480 píxeles) y 8 bits RGB, a una frecuencia de 30 frames por segundo (fps).

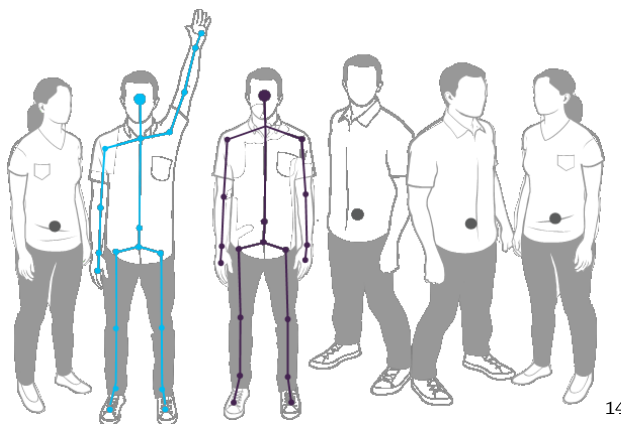
Una vez proyectado el patrón sobre el escenario, éste rebota en los objetos y es captado por la cámara de infrarrojos.

La cámara posee una profundidad de 11 bits lo que proporciona hasta 2048 niveles.

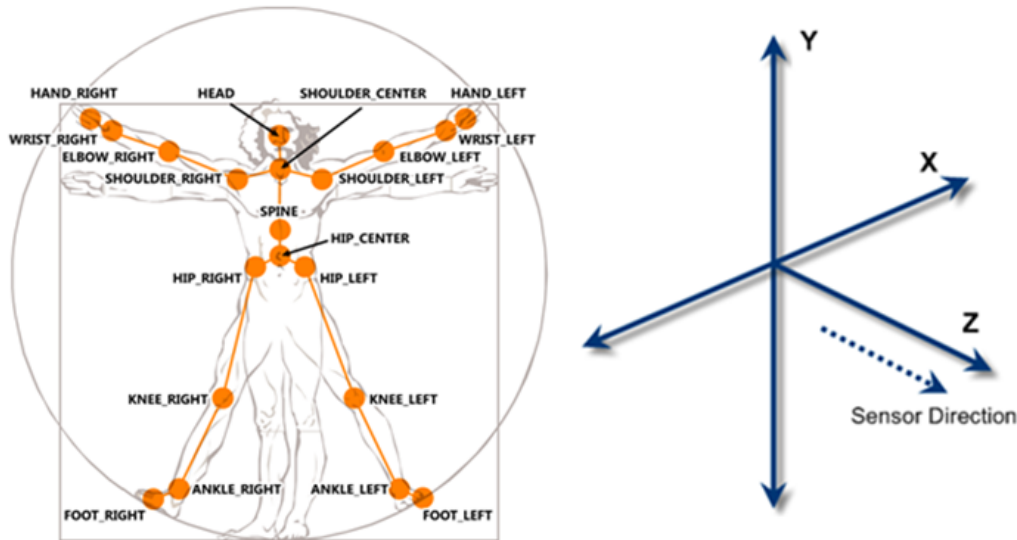
Una vez Kinect tiene la imagen infrarroja se procesa en el propio hardware. El procedimiento consiste en comparar el patrón IR con la imagen RGB que ha sido capturada por la cámara. Observando las perturbaciones en las distancias entre puntos contiguos se puede calcular cual es la diferencia en profundidad. Permitiendo de esta forma la construcción de un mapa de profundidad.

Para completar los elementos que conforman el Kinect, proporciona una matriz de 4 micrófonos además de un motor que permite levantar o bajar el punto de vista de las cámaras. El hecho de incluir 4 micrófonos permite que se pueda realizar una localización de la fuente del sonido así como una cancelación del ruido de fondo.

El sistema permite basándose en los datos RGBDeep la construcción de un esqueleto (Skeleton) para el seguimiento de un sujeto. El número máximo de sujetos al cual es capaz de realizar un seguimiento es de 6, mientras que solo reconocerá el skeleton de dos de estos sujetos en seguimiento.



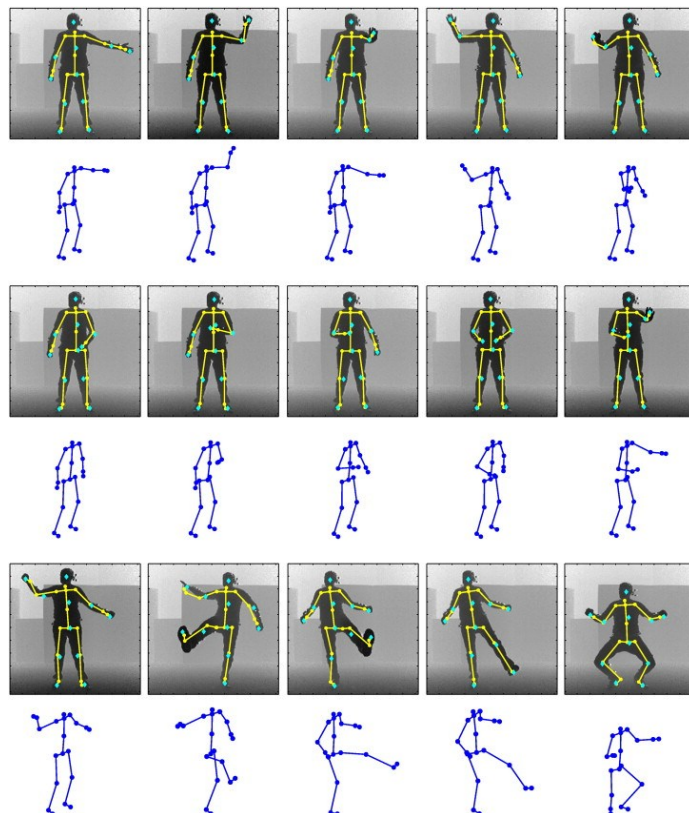
El skeleton estándar de kinect está basado en 20 puntos de referencia como se puede observar en la imagen a continuación. Dichos puntos de referencia son conocidos como Joints y cada Joint tendrá las coordenadas XYZ de la referencia. Por lo tanto se puede decir que un skeleton consiste en la unión en un espacio tridimensional de distintos Joints.



15

Cada skeleton localizado es considerado como un frame¹⁶, hacer notar que un frame de skeleton no tiene porque coincidir con un frame de video, debido al tiempo de computo la relación entre frame de video y frame de skeleton es de 3 a 1.1, un video a 30fps devuelve un esqueleto a 11 fps, por lo que 33 skeleton frames tendrían una duración de tres segundos.

En la siguiente imagen se puede observar como una secuencia de frames de skeletons describe un gesto o movimiento basandose en sus Joints de referencia.



La captura de varios frames o fotogramas crea una secuencia¹⁸. En el caso de estudio particular de este proyecto cada gesto queda caracterizado por 32 fotogramas de puntos de referencia pertenecientes al esqueleto y cada punto con sus coordenadas xyz, corregidas respecto a un punto central de referencia. Por lo tanto un gesto de aproximadamente tres segundos es representado por:

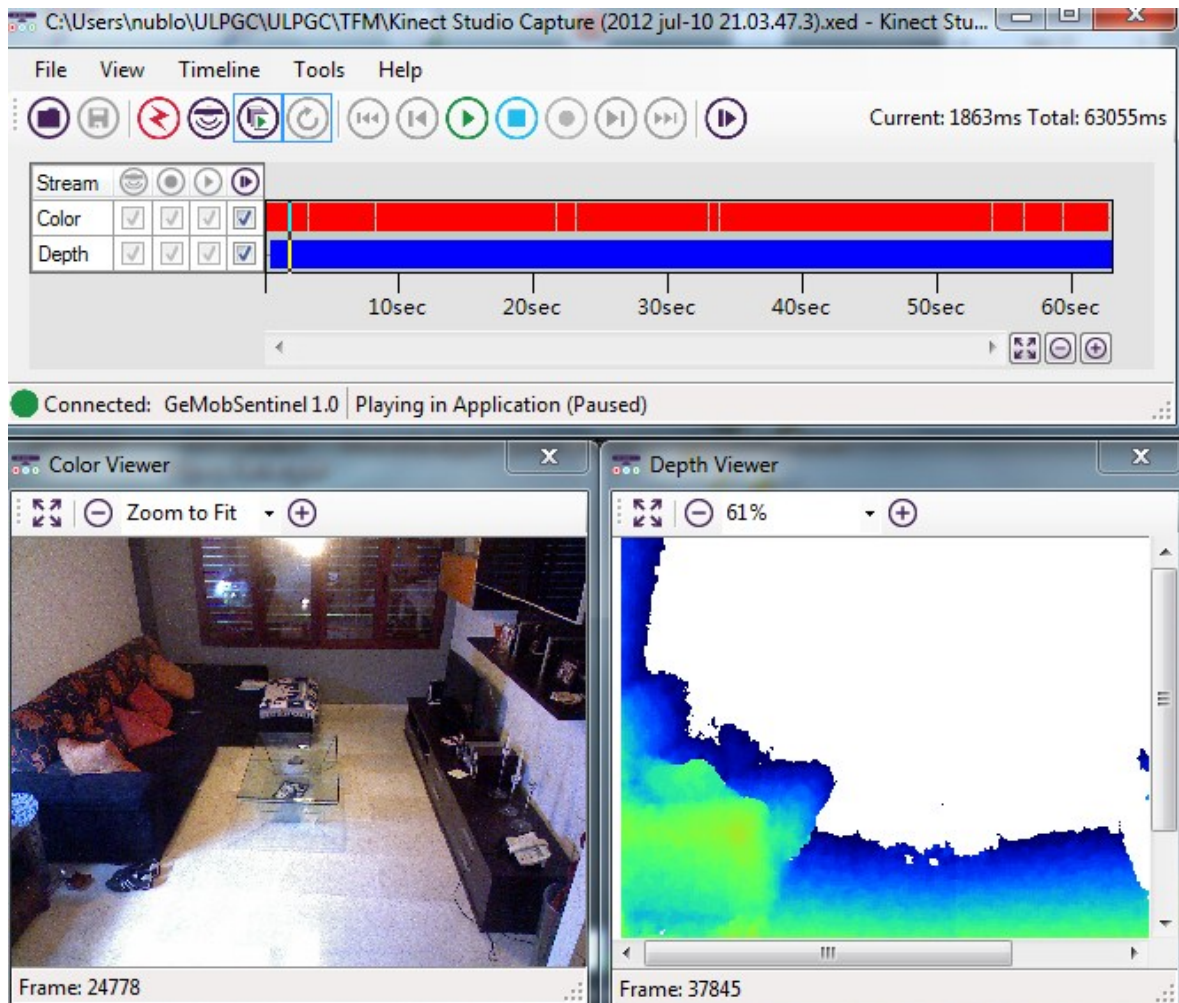
$$\text{Secuencia} = 32\text{frames} \times 6\text{Joints} \times 3\text{puntos}.$$

Kinect Studio:

Kinect es un sensor que detecta personas, por ello cada una de las pruebas a realizar requiere de un sujeto objetivo delante del sensor y realizar el gesto correspondiente.

La versión 1.5 de Kinect SDK, ofrece una nueva herramienta llamada Kinect Studio el cual permite grabar una sesión frente al kinect y luego reproducir la misma las veces que sea necesario.

Para esto lo primero que vemos en su interfaz es una barra de grabación y reproducción.



Cuando se lanza una aplicación o se activa el botón conectar online se obtiene un listado con todos aquellos procesos que están accediendo al sensor. Una vez conectarnos a este proceso ya se permite el comienzo de la grabación

Una vez grabada una sesión permite la visualización de la misma en la aplicación, la duración y el reutilizado de la misma. La grabación almacenada suele tener 35Megas por segundo de video.

En cualquier momento es posible pausar la reproducción, detenerla o inclusive poner un modo repetición que permitirá fundir los tiempos de proceso de la aplicación.

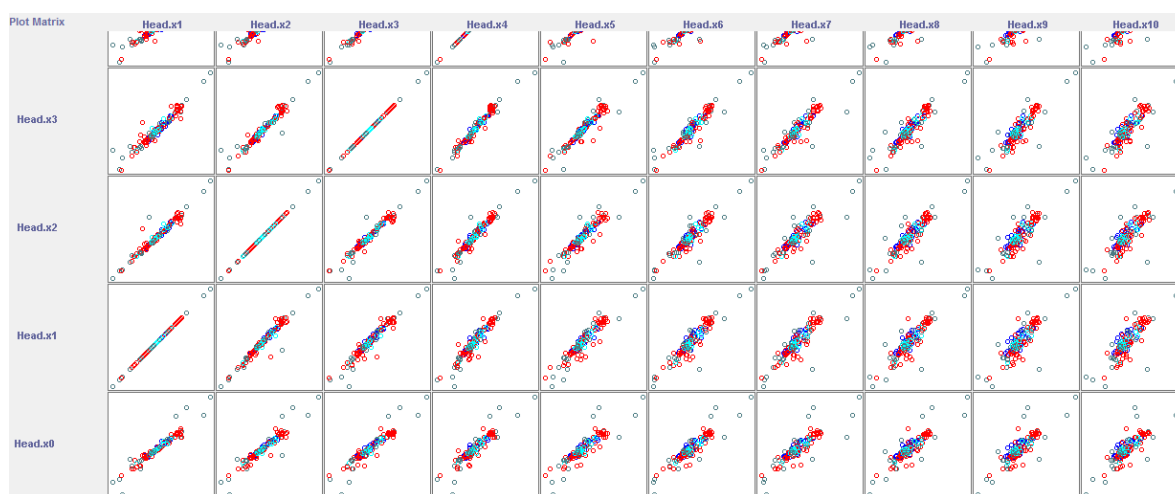
Posibilita a su vez la visualización de la cámara, del sensor de profundidad y de un visor para ver los datos que está procesando el Kinect.

WEKA

Weka (Waikato Environment for Knowledge Analysis Entorno para Análisis del Conocimiento de la Universidad de Waikato) es una plataforma de software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka es un software libre distribuido bajo licencia GNU-GPL.

El paquete Weka contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue un front-end en TCL/TK para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para preprocesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación.

En la imagen siguiente se puede observar la relación entre la coordenada x de la cabeza en función de los frames y el gesto correspondiente.



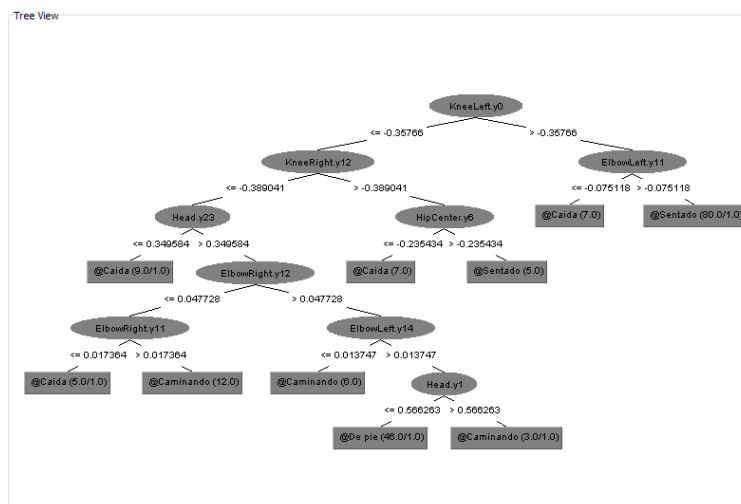
En este trabajo se compararon cuatro clasificadores pertenecientes a tres paradigmas de aprendizaje distintos. Para ello se ha utilizado la versión de software Weka 3.4

A continuación se bosquejan estos paradigmas y los clasificadores correspondientes, mencionando solo las características más distintivas. Los parámetros de los algoritmos fueron optimizados para obtener los mejores resultados. A menos que se mencione, los parámetros fueron los predeterminados.

Árboles de Decisión (tree)

Los árboles de decisión se construyen comenzando por la raíz hasta las hojas. Primeramente se escoge un atributo para discriminar y se produce un subnodo por cada valor del atributo. Si todos los ejemplos con un valor particular de atributo tienen la misma clase, el nodo se convierte en hoja, de otra forma se escoge otro atributo para seguir discriminando entre las clases. El árbol está completo cuando todos los ejemplos son representados por un nodo hoja. Para determinar cuál atributo se ramifica en cada nivel se calcula la información ganada al discriminar con cada atributo y se usa aquel que maximice la ganancia. Un árbol aprendido puede representarse también como un conjunto de reglas si-entonces, más fáciles de entender para un usuario.

- C4.5 (en Weka: J48) Genera árboles de decisión para atributos discretos y continuos, utiliza la razón de ganancia para seleccionar el atributo de cada nodo y aplica estrategias de poda para reducir el ruido de los datos de entrenamiento.
- Bosque Aleatorio (en Weka: RandomForest) Este algoritmo construye muchos árboles de decisión. Cada árbol se construye escogiendo, entre los atributos que describen a las instancias, un pequeño subconjunto aleatorio. Para clasificar un nuevo objeto, se le da el vector que lo describe a cada árbol, los cuales hacen su clasificación independientemente. Cada clasificación es un voto. El bosque selecciona la clasificación que más votos obtenga.

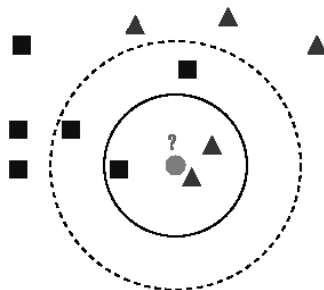


Algoritmos Basados en Instancias (Lazy)

Este tipo de Algoritmo clasifica una instancia comparándola con una base de datos de ejemplos pre-clasificados. La principal suposición que se hace es que instancias similares tendrán clasificaciones similares. Conocidos como "holgazanes" debido a que realizan poco trabajo en la etapa de aprendizaje, en los casos más simples solo se almacenan los ejemplos en memoria, transfiriendo el esfuerzo al momento de clasificar una nueva instancia, el

sistema debe decidir cuales de los ejemplos memorizados debe utilizar para hacer la clasificación.

- K Vecinos Más Cercanos (en Weka: IBk) Cuando se le proporciona una nueva instancia, este algoritmo busca -entre las que se utilizaron durante el entrenamiento las K instancias más parecidas, y clasifica la instancia en la clase de mayor frecuencia entre estas K instancias. Utiliza algún método de cálculo de distancia como puede ser la distancia Euclídea para medir similitud entre instancias.



Paradigma metaclasificadores

Además de buscar el algoritmo que individualmente rinde mejores resultados para un problema determinado existe una vía alternativa para mejorar aún más la precisión: agrupando los clasificadores en ensambles, también llamados metaclasificadores. Un ensamble es un conjunto de clasificadores cuyas predicciones individuales son combinadas de alguna manera (típicamente mediante el voto) para clasificar nuevos ejemplos. Esta es una de las áreas más activas de investigación en aprendizaje supervisado ya que los ensambles son mucho más precisos que los clasificadores individuales que los componen.

- Boosting: Se reajustan repetidamente los pesos de los ejemplos de entrenamiento, concentrándose en los ejemplos "difíciles" de muestras anteriores. Los ejemplos que son mal clasificados se le asigna mayor peso en la próxima iteración, por ejemplo, las instancias que están próximas a la frontera de decisión son a menudo más difíciles de clasificar y por tanto adquieren mayor peso tras unas pocas iteraciones.

La primera tarea en el proceso de detección del gesto es la recuperación de uno o más casos similares de la base de casos donde está almacenada la experiencia pasada en forma de casos. Un caso está formado por varios atributos (30x6x3 coordenadas) que dan una descripción del problema y una solución para el caso. Para poder llevar a cabo la tarea de recuperación es necesario tener un algoritmo de recuperación y una medida de similitud que serán usados para obtener un conjunto de casos similares.

En nuestro caso, como se mostrará en el apartado 9 Desarrollo, se ha elegido el algoritmo de los K-vecinos como algoritmo de recuperación y dos medidas de similitud: distancia Euclídea y Alineamiento Dinámico Temporal (DTW).

DTW. Alineamiento Temporal Dinámico (DTW)¹⁹

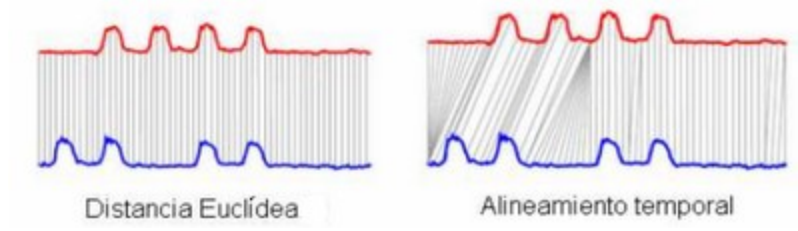
El algoritmo de alineamiento temporal dinámico nos permite medir la similitud entre dos secuencias que pueden variar en el tiempo o en el espacio.

Gracias a él, podremos calcular la distancia entre un gesto que realicemos y cada uno de los gestos patrón almacenados. El gesto reconocido será aquel que se encuentre a menor distancia.



La gráfica anterior muestra las coordenadas relativas a la cabeza respecto a algunos gestos de ejemplo. El tramo inicial referente a la coordenada “x”, el tramo central la coordenada “y” y por ultimo el trama correspondiente a la coordenada “z” en función del tiempo.

Aunque podríamos haber elegido no utilizar este algoritmo, se ha tenido en cuenta el supuesto de que, cuando realizamos un gesto, no siempre haremos un movimiento de la misma duración. Si calculásemos distancias euclídeas y tuviésemos dos vectores idénticos, pero uno desplazado ligeramente en el tiempo respecto al otro, al hacer la comparación punto a punto devolvería que ambos vectores son distintos. Mediante el DTW podremos realizar gestos con mayor o menor velocidad y escalar dichas secuencias para un mejor acierto.

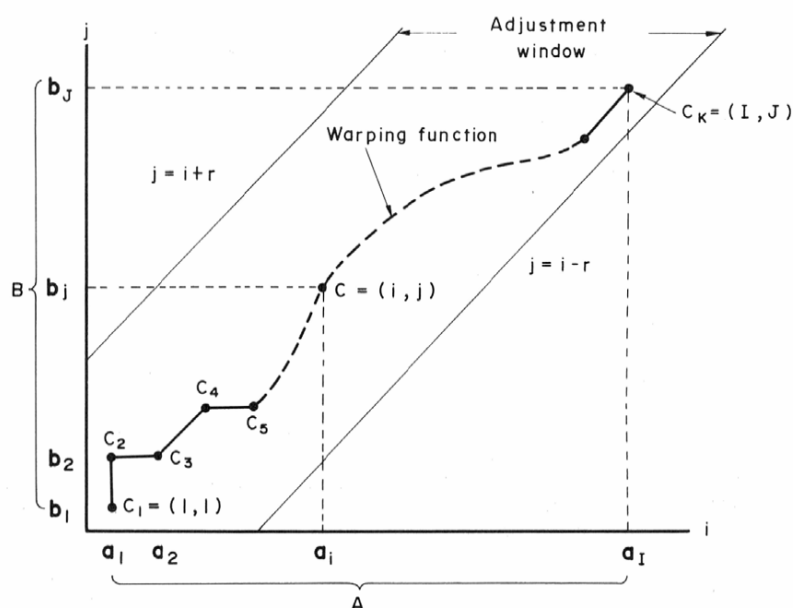


Originalmente, el algoritmo DTW se usaba para comparar distintos patrones acústicos en el reconocimiento automático del habla. Otras aplicaciones son el reconocimiento de gestos o de escritura.

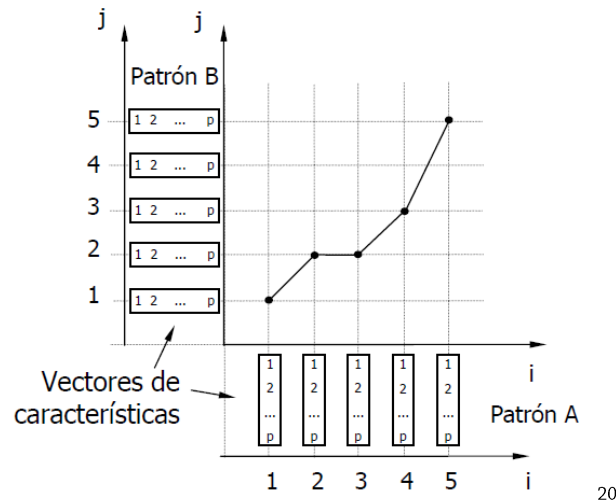
La problemática asociada hace referencia a la dificultad añadida en el proceso de medida de distancia entre patrones, puesto que se estarán comparando tramos que pueden corresponder a unidades fónicas distintas. Será necesario alinear temporalmente la locución para proceder a realizar una medida de distancia entre patrones cuyo nuevo eje temporal haya homogeneizado las variaciones iniciales.

La figura siguiente muestra dos realizaciones de la misma locución (contorno de energía localizada) antes y después de ser alineadas: Alineamiento temporal (temporal warping): en este proceso, el eje temporal de la señal de test se comprime y expande de forma no lineal para alinear los vectores de características entre patrón y test.

De forma genérica, podemos presentar la función de alineamiento a través de una representación del tipo siguiente:



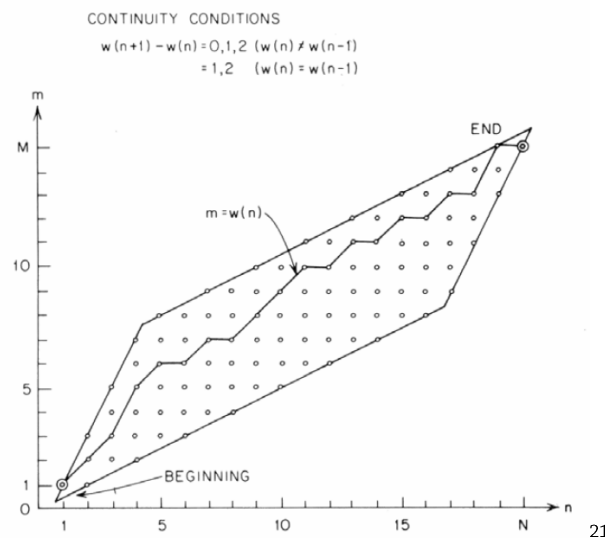
Si observamos la función de alineamiento en detalle, tendremos que, si la función de alineamiento pasa por (i,j) , comparamos el vector 'i' de A con la 'j' de B.



Generalmente existe algún tipo de límite global en la máxima cantidad de alineamiento. Esto se puede expresar como

$$i(k) - j(k) < Q$$

lo que da lugar a una restricción denominada de 'ventana' o rombo, siendo Q el ancho de la ventana:



La programación dinámica nos dice que el mejor camino entre $(1,1)$ y cualquier punto (i,j) es independiente de lo que suceda más allá de este punto.

Por tanto, el coste total del punto $[i(k), j(k)]$ es el propio de este punto más el camino más 'barato' hasta él:

$$D(Ck) = d\{c(k)\} + \min\{D(Ck-1)\};$$

$c(k-1)$ representa todos los predecesores permitidos de $c(k)$.

Sin embargo, por la limitación local tenemos un conjunto muy limitado de predecesores posibles.

Podemos resumir el algoritmo de programación dinámica como:

1. Inicialización

$$D(C1) = d\{c(1)\}$$

2. Recorremos el índice 'i':

para cada i desde i=1 hasta i=M

para cada j permitido

Calculo y guardo la distancia acumulada según (*dtw. 1*) y la posición del predecesor

fin para j

fin para i

3. Final

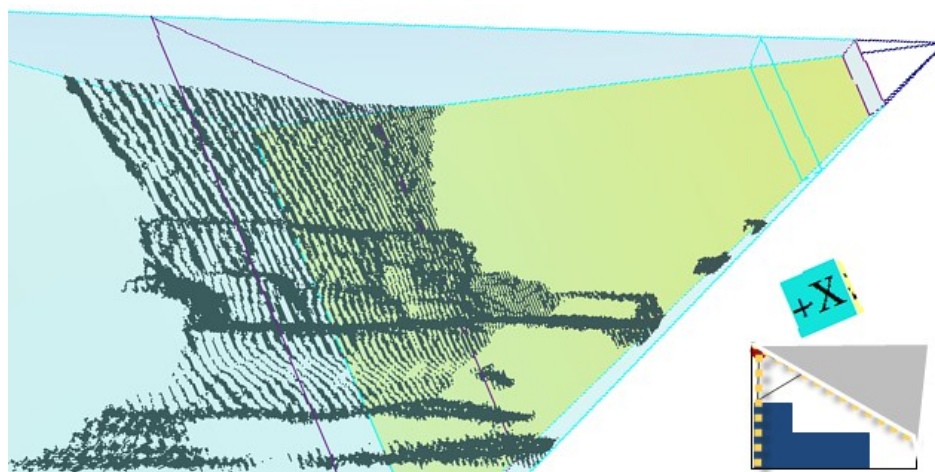
$$d(A, B) = D(CK) / K \quad c(K) = [i(M), j(N)]$$

Para implementar el algoritmo sólo necesito disponer de dos array adicionales:

- Uno de tamaño $M \times N$ para guardar el predecesor de cada punto
- Otro de tamaño $2 \times N$ para guardar los costes acumulados de la columna anterior y los de la presente, actualizándolos en cada paso de i ²²

9. DESARROLLO²³

Como se ha comentado en los fundamentos teóricos, el sensor Kinect presenta una serie de limitaciones tanto de alcance en profundidad como en ángulos de visión horizontal y vertical. Debido a estas restricciones la situación del sensor respecto al recinto en el cual se realizará el seguimiento del sujeto es determinante tanto en el entrenamiento del clasificador como en la fase de detección.



Instalación se ha realizado en una de las esquinas del cuarto con una elevación de dos metros y un ángulo de -45° respecto a la horizontal como muestra la figura.

Para sistema de reconocimiento de gestos se ha diseñado una interface gráfica que permitirá conocer el estado del sensor, lectura de las coordenadas y visualización del esqueleto. Así como tareas de captura de gestos, almacenaje y carga de los mismos. Todo el entorno ha sido implementado en c#.²⁴

El funcionamiento de la interface es bastante sencilla consistente en un menú superior basado en botones y un listado desplegable. Una vez el sistema ha detectado una kinect conectada, automáticamente genera el esqueleto correspondiente si existiera dentro de su ángulo de visión y comenzaría a intentar identificar el gesto, mostrando su nombre y su distancia respecto al gesto aprendido previamente.

Nada mas arrancar el programa deberá tenerse en cuenta que no tiene ninguna base de conocimiento cargada por defecto, por lo tanto sería necesario realizar un entrenamiento o bien la carga de un entrenamiento previamente grabada.

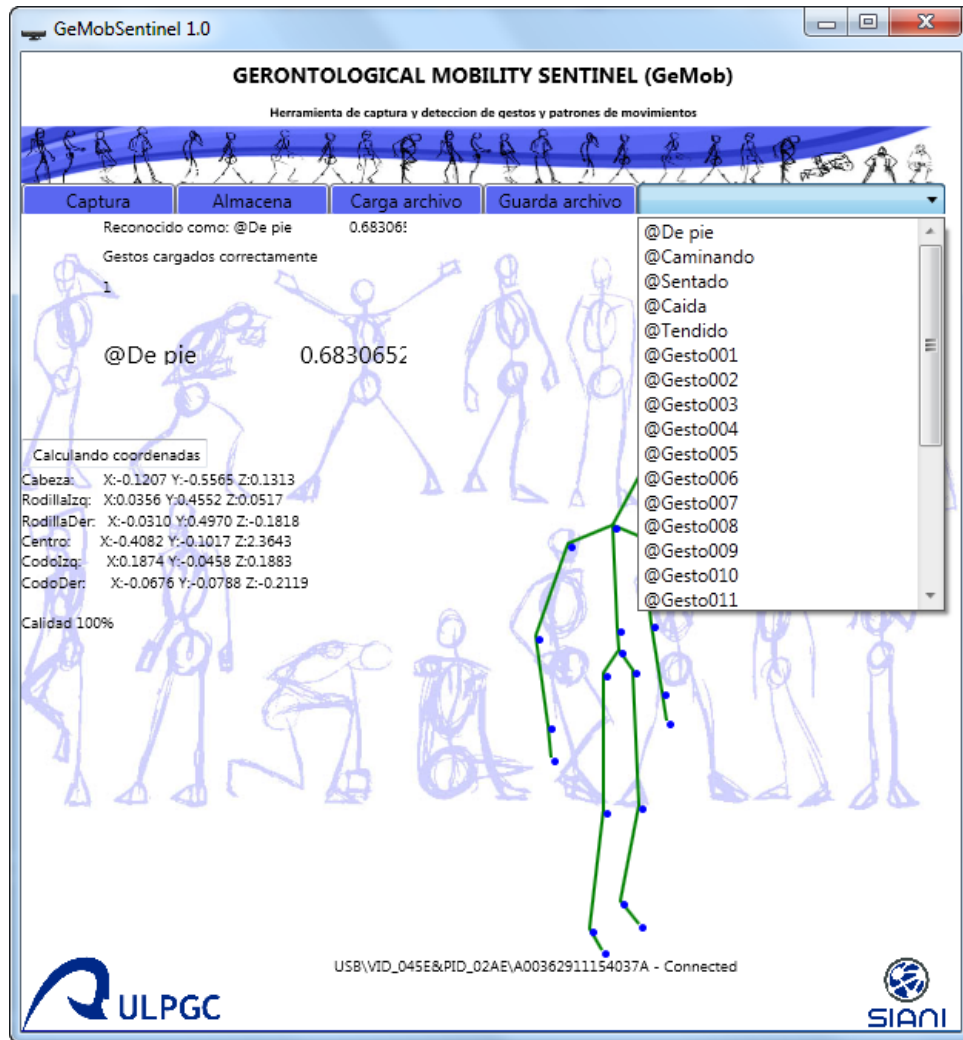
Para realizar el entrenamiento bastará con seleccionar en la lista desplegable el gesto que queremos capturar y mientras el sujeto hace el gesto realizar la captura del mismo mediante la pulsación del botón capturar. Repetir el proceso con capturas del gesto anterior en distintos lugares de la sala y ángulos respecto al sensor o seleccionando un gesto nuevo.

El número máximo de gestos es de 64 en total contando las repeticiones del mismo. Ej. 10 Caminando, 10 secuencias de gestos sentado, 10 de pie, 30 caídas. Dado que se trabaja con esqueletos es posible realizar el entrenamiento con una persona distinta al sujeto en estudio.

Una vez capturados los gestos seleccionar Carga archivo para realizar una copia de los mismos en fichero. La carga del fichero ya guardado se realizará con el fichero correspondiente. Hacer notar que una vez se ha cargado un fichero de gestos, cualquier gesto nuevo capturado se añadirá a la lista de gestos por lo que hay que tener especial atención para no superar el límite de los 64 gestos totales por matriz de gestos. Esta limitación viene dada por el tamaño máximo del tipo `ArrayList` y es un apartado a revisar en futuras versiones.

Con el entrenamiento previo en memoria el detector comenzará a calcular las coordenadas de los Joints points que forman el esqueleto y les aplica una corrección de posición respecto a su `hipCenter`, punto central del esqueleto. Los Joints no visibles son estimados, dicha predicción es indicada por (P) precediendo a las coordenadas x y z del punto en cuestión. Para el cálculo de dicha predicción se ha tenido en cuenta un suavizado o `Smoothing` del punto en sus frames anteriores, se limitan la desviación máxima del punto y su corrección y un porcentaje de posibilidad de predicción y ruido.

Otros datos de interés son la propia localización del gesto y su distancia entre cero e infinito, significando cero un acierto perfecto e infinito un gesto desconocido. Un indicador del número de frames representados y capturados del esqueleto y un nivel de calidad de esqueleto así como avisos de detección de fuera de campo de visión.



Como anteriormente se ha comentado cada Joint representa un punto de referencia del cuerpo necesario para construir un skeleton, dicho Joint es transformado en un point3D en el cual se guardan cada una de las coordenadas (x,y,z) del punto anterior.

Para la creación de un esqueleto completo se tienen 20 Joints, los cuales son representados por la aplicación pero solo 6 de esos marcadores son añadidos al vector de atributos.

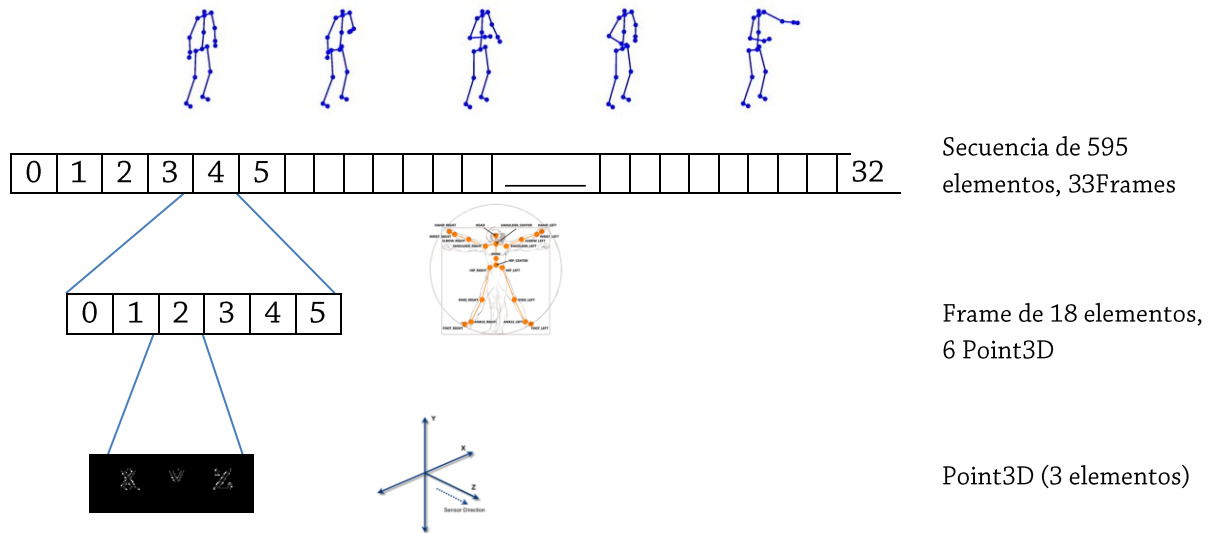
Point3D	X	Y	Z
---------	---	---	---

Cada punto de referencia es normalizado y trasladado respecto al punto central del esqueleto en sus tres componentes. La creación de un vector correspondiente a un esqueleto en el instante actual es denominado vector frame. (Frame[18])

Cabeza	CodoIzquierdo	CodoDerecho	Centro	RodillaIzquierda	RodillaDerecha
X Y Z	X Y Z	X Y Z	X Y Z	X Y Z	X Y Z

Un gesto por el contrario se podría definir como la sucesión de diferentes frames en función del tiempo, por lo que se ha estimado que un gesto o secuencia tiene una duración máxima de tres segundos, dado que se ha comentado anteriormente que se representan 11 fps esqueletos, una secuencia de 33 frames es la encargada de caracterizar al gesto completo.

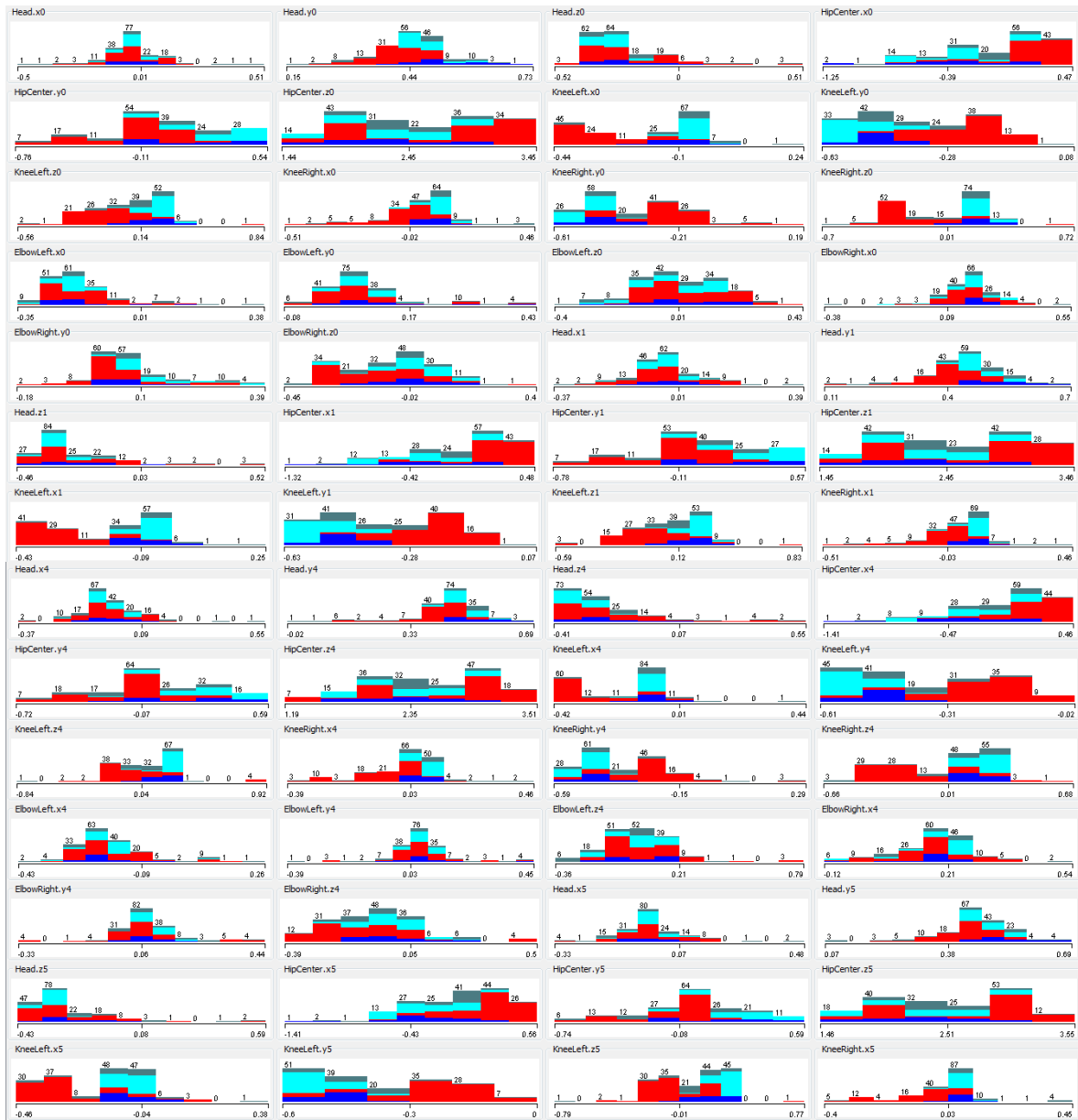
Resultando una secuencia de



Una vez realizada la elección del gesto y captura de la secuencia correspondiente se realiza el almacenaje de un máximo de 64 secuencias entre 4 gestos distintos. El fichero Gestos_Train.csv por ejemplo se compone de 5 secuencias de caídas, 13 secuencias caminando, 17 secuencias de pie y 23 secuencias sentado haciendo un total de 58 secuencias en el archivo y dejando espacio para la captura de 6 secuencias mas o nuevo gesto.

Para la elección del mejor algoritmo de clasificación se ha combinado tres archivos csv para aumentar el número de instancias a 180. El número de atributos corresponde a los 595 elementos de cada secuencia (instancia).

Inicialmente se ha realizado una visualización de la distribución de atributos e instancias y observar de una manera intuitiva cuan separadas están cada una de las muestras entre si.



El siguiente paso a realizar es la aplicación del entrenamiento según los cuatro algoritmos considerados en el punto ocho de fundamentos teóricos.

Relation: gestos_train_grande
Instances: 180 3 archivos csv distintos
Attributes: 595
Test mode:10-fold cross-validation

El primer resultado a tener en consideración es el tiempo de computo, la creación y calculo del propio esqueleto consume un valor cercano a tres frames de video, si añadimos el tiempo de computo el

resultado sería una ralentización de la creación del esqueleto, de la representación de este lo que origina perdida de potenciales esqueletos.

TIPO	ALGORITMO	COMANDOS WEKA			TIEMPO
lazy	IB1	-K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""			0.00 sec
meta	AdaBoostM1	-P 100 -S 1 -I 10 -W weka.classifiers.trees.DecisionStump		Iteraciones: 3	0.24 sec
trees	J48	-C 0.25 -M 2		10 Hojas 19 Nodos	0.36 sec
trees	RandomForest	.-I 10 -K 0 -S 1			0.32 sec

Según los resultados de la tabla el algoritmo IBK aka KNN es el que consume un menor tiempo de procesado. Por lo que teniendo ese dato en cuenta se ha revisado las estadísticas de los clasificadores resultados según instancias o secuencias de gestos

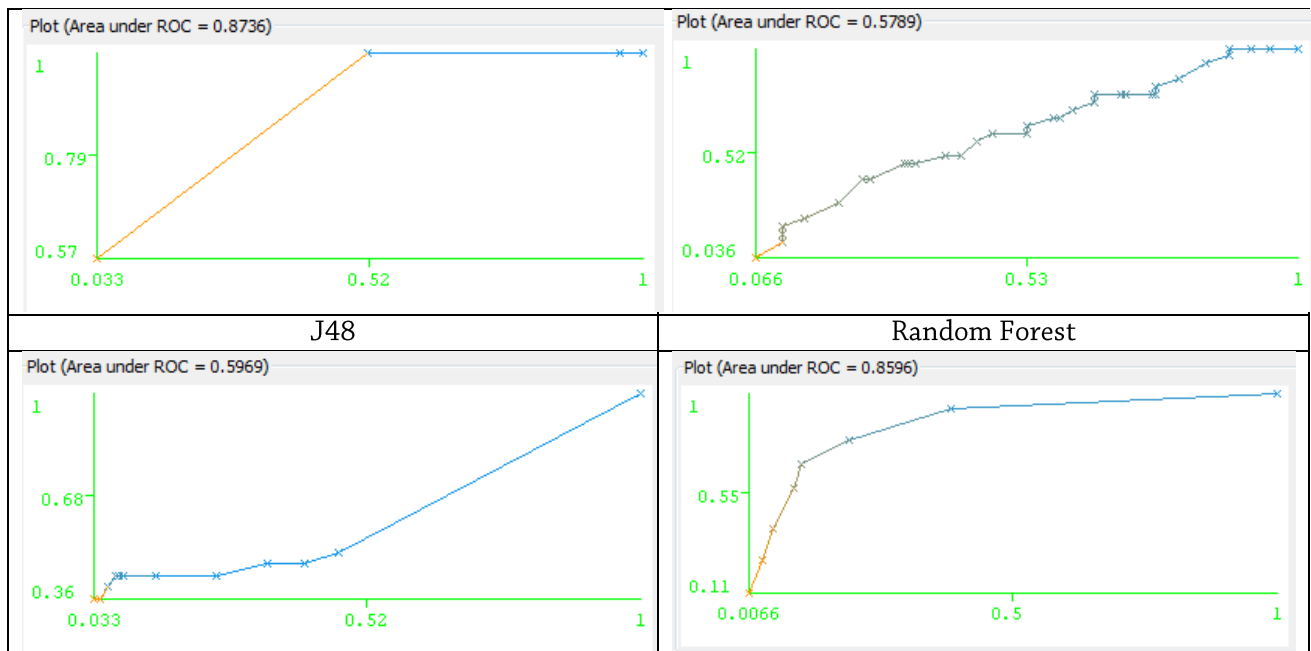
IBK			AdaBoost		
Correctly Classified Instances	154	85.5556%	Correctly Classified Instances	122	67.7778%
Incorrectly Classified Instances	26	14.4444%	Incorrectly Classified Instances	58	32.2222%
Kappa statistic		0.7838	Kappa statistic		0.4959
Mean absolute error		0.0777	Mean absolute error		0.3176
Root mean squared error		0.2657	Root mean squared error		0.3774
Relative absolute error		22.8063%	Relative absolute error		93.2449%
Root relative squared error		64.4512%	Root relative squared error		91.5455%
Total Number of Instances		180	Total Number of Instances		180
J48			Random Forest		
Correctly Classified Instances	135	75%	Correctly Classified Instances	138	76.6667%
Incorrectly Classified Instances	45	25%	Incorrectly Classified Instances	42	23.3333%
Kappa statistic		0.6201	Kappa statistic		0.6526
Mean absolute error		0.128	Mean absolute error		0.1353
Root mean squared error		0.343	Root mean squared error		0.2667
Relative absolute error		37.5809%	Relative absolute error		39.718%
Root relative squared error		83.1966%	Root relative squared error		64.6835%
Total Number of Instances		180	Total Number of Instances		180

La tabla anterior muestra un acierto del 85.5% para el k=1 vecino mas cercano como se puede ver con mas detalle en la próxima tabla

IBK						
TPRate	FPRate	Precision	Recall	F-Measure	ROC Area	Class
0.522	0.045	0.632	0.522	0.571	0.82	@Caminando
0.976	0.063	0.932	0.976	0.953	0.976	@Sentado
0.978	0.059	0.846	0.978	0.907	0.979	@De pie
0.571	0.033	0.762	0.571	0.653	0.874	@Caida
0.856	0.055	0.846	0.856	0.846	0.941	Media
AdaBoost						
TPRate	FPRate	Precision	Recall	F-Measure	ROC Area	Class
0	0	0	0	0	0.711	@Caminando
0.94	0.135	0.859	0.94	0.898	0.928	@Sentado
0.956	0.333	0.489	0.956	0.647	0.837	@De pie
0	0	0	0	0	0.579	@Caida
0.678	0.147	0.523	0.678	0.581	0.824	Media
J48						
TPRate	FPRate	Precision	Recall	F-Measure	ROC Area	Class
0.261	0.089	0.3	0.261	0.279	0.581	@Caminando
0.964	0.146	0.853	0.964	0.905	0.921	@Sentado
0.844	0.074	0.792	0.844	0.817	0.92	@De pie
0.357	0.046	0.588	0.357	0.444	0.597	@Caida
0.75	0.105	0.726	0.75	0.731	0.827	Media
Random Forest						
TPRate	FPRate	Precision	Recall	F-Measure	ROC Area	Class
0.435	0.102	0.385	0.435	0.408	0.897	@Caminando
0.94	0.094	0.898	0.94	0.919	0.982	@Sentado
0.844	0.067	0.809	0.844	0.826	0.963	@De pie
0.393	0.053	0.579	0.393	0.468	0.86	@Caida
0.767	0.082	0.76	0.767	0.76	0.947	Media

Resultado de la curva ROC (Receiver Operating Characteristic, o Característica Operativa del Receptor) en representación gráfica de la sensibilidad para cada sistema clasificador según se varía el umbral de discriminación. Observando la representación del ratio de verdaderos positivos frente a falsos positivos según se varía el umbral de discriminación, se relaciona de forma directa y natural con el análisis de coste/beneficio en toma de decisiones.

IBK	AdaBoost
-----	----------



Y se ha concluido el estudio de las características con la matriz de confusión representando como ha sido reconocida cada instancia según cada algoritmo.

IBK						AdaBoost					
a	b	c	d		Classified as	a	b	c	d		Classified as
12	0	6	5	a	@Caminando	0	2	21	0	a	@Caminando
1	82	1	0	b	@Sentado	0	79	5	0	b	@Sentado
1	0	44	0	c	@De pie	0	2	43	0	c	@De pie
5	6	1	16	d	@Caida	9	1	9	0	d	@Caida

J48						Random Forest					
a	b	c	d		Classified as	a	b	c	d		Classified as
6	5	7	5	a	@Caminando	10	1	8	4	a	@Caminando
1	81	0	2	b	@Sentado	1	79	0	4	b	@Sentado
5	2	38	0	c	@De pie	7	0	38	0	c	@De pie
8	7	3	10	d	@Caida	8	8	1	11	d	@Caida

Despues de seleccionar el algoritmo KNN (IBK) las muestras quedan como el siguiente grafico de clusters.

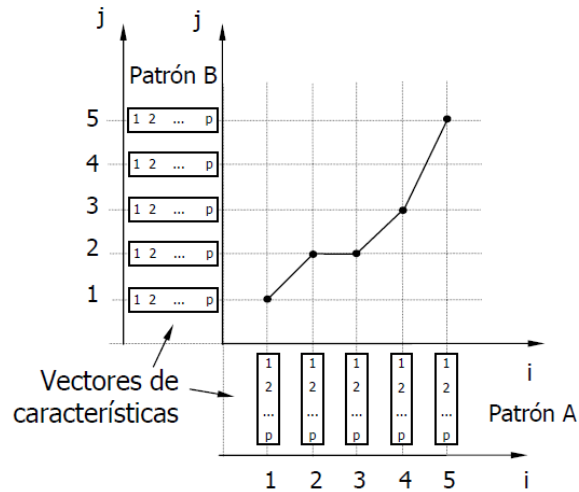


Una vez implementado en el programa el algoritmo se pasa a cargar el archivo de entrenamiento y comienza a capturar secuencias de gestos continuamente llamada secuencia de test, esta secuencia de test se compara la distancia entre cada valor del vector con cada uno de los 64 vectores de entrenamiento calculando su distancia.

Una vez determinado que la distancia entre la secuencia de entrenamiento y la secuencia actual de test están lo suficientemente cerca, son vecinos cercanos, se pasa un segundo filtrado DTW encargado de alinear dinámicamente y seleccionar entre todas las secuencias su secuencias mas cercana.

Tal como se comentó en el apartado ocho de fundamentos teóricos, se ha elegido el algoritmo de los K-vecinos como algoritmo de recuperación y dos medidas de similitud: distancia Euclídea y Alineamiento Dinámico Temporal(DTW).

Recordemos que Aunque podríamos haber elegido no utilizar este algoritmo, se ha tenido en cuenta el supuesto de que, cuando realizamos un gesto, no siempre haremos un movimiento de la misma duración. Si calculásemos distancias euclídeas y tuviésemos dos vectores idénticos, pero uno desplazado ligeramente en el tiempo respecto al otro, al hacer la comparación punto a punto devolvería que ambos vectores son distintos. Mediante el DTW podremos realizar gestos con mayor o menor velocidad y escalar dichas secuencias para un mejor acierto.



```
Funcion Recognize(Secuencia_actual)
```

```
    mínima Distancia = ∞;
```

```
    Para (secuencia_ejemplo_primera hasta secuencia_ejemplo_ultima)
```

```
    {
        example = Secuencia[i];
```

```
        Si (Distancia_euclidea(Secuencia_actual y SecuenciaExample) <
nivelCorte)
```

$$distanciaDTW = \frac{AlineamientoTemporalDinamico(seq, example)}{numero\ de\ examples}$$

```
        Si (distanciaDTW < mínima Distancia)
```

```
            mínima Distancia = distanciaDTW;
            classification = labels[i];
```

```
    }
```

```
    Si (mínima Distancia < nivel de corte2)
```

```
    return (classification )
```

```

Función AlineamientoTemporalDinamico(Secuencia1, Secuencia2)
{
// Inicializar
    TablaDTW      [Tamaño_Secuencia1+1, Tamaño_Secuencia2+1];
    Pendiente_I   [Tamaño_Secuencia1+1, Tamaño_Secuencia2+1];
    Pendiente_J   [Tamaño_Secuencia1+1, Tamaño_Secuencia2+1];

    inicializamos TablaDTW; Pendiente_I; Pendiente_J

    TablaDTW[0, 0]=0;

// Alineamiento Temporal Dinámico (Dynamic Time Warping, DTW) de la matriz.

    Para (i=:)
    Para (j=:)
        Si (TablaDTW      [i, j-1]    < TablaDTW[i-1, j-1]      y
            TablaDTW      [i, j-1]    < TablaDTW[i-1, j]        y
            Pendiente_I   [i, j-1]    < MaxPendiente)
        {
            TablaDTW      =Distancia( Secuencia1, Secuencia2)+TablaDTW[i, j-1];

            Pendiente_I   =Pendiente_J[i, j-1]+1;
            Pendiente_J   =0;
        }

    Sino Si( TablaDTW      [i-1, j]    < TablaDTW[i-1, j-1]      y
            TablaDTW      [i-1, j]    < TablaDTW[i, j-1]        y
            Pendiente_J   [i-1, j]    < MaxPendiente)
        {
            TablaDTW      = Distancia( Secuencia1,Secuencia2)+TablaDTW[i-1, j];

            Pendiente_I   =0;
            Pendiente_J   =Pendiente_J[i-1, j]+1;
        }

    sino {
        TablaDTW          = Distancia(Secuencia1,Secuencia2)+TablaDTW[i-1,j-1];

        Pendiente_I   =0;
        Pendiente_J   =0;
    }

// Buscamos el mejor acierto entre la Secuencia2 y algo que termine como la
secuencia1.
    Acierto =Infinity;

    Para (i = 1:Tamaño secuencia)
        Si TablaDTW[i, Tamaño Secuencia] < Acierto
            Acierto = TablaDTW[i, Tamaño Secuencia];

    return Acierto;
}

```

10. CONCLUSIONES

A pesar de presentar un buen resultado a la hora de clasificar los gestos el sistema no se recomendaría para su instalación con fines comerciales para el control de movilidad de personas mayores y detección de caídas.

Los reducidos ángulos de visión imposibilitan el control total de una sala de estar típica de más de 16 metros cuadrados a partir del cual el sensor se podría considerar ciego e imposibilitado para construir el esqueleto.

Como punto positivo destacar que con unas mínimas mejoras se podría adaptar para realizar un seguimiento de ejercicio físico y de puesta en forma para el mismo grupo de personas de avanzada edad.

La posibilidad de extraer los esqueletos a archivo csv ofrece una herramienta a futuros estudiantes para un estudio rápido en Matlab, Excel, pagina web o como es el caso presente weka.

El anuncio de una versión kinect2 y de otros dispositivos parecidos pero de mucha mayor resolución es probable que resuelvan los problemas apuntados en los párrafos anteriores.

11. LINEAS FUTURAS

A lo largo del desarrollo del proyecto se ha realizado una primera aproximación a la herramienta Kinect y se ha comprobado que sus campos de aplicación van mucho más allá que el campo de los videojuegos para el que estaba originalmente diseñado.

Algunas ideas para mejorar la aplicación serian las siguientes:

- Estudiar la posibilidad de añadir factores o herramientas de ayuda a la movilidad como pueden ser sillas de ruedas, muletas, bastones o cualquier otro utensilio.
- La activación de entrenamiento físico, acciones como comer, leer, y demás actividades pueden ser añadidas en versiones posteriores.
- Implementación de otros tipos de algoritmos clasificadores y posibilidad de selección desde el menú.
- Ampliación del límite de 64 gestos o secuencias máximas.
- Comunicación externa de aviso de actividad y estadísticas de movimiento.
- Posibilidad de seleccionar los joints de construcción de esqueletos y almacenamiento de sus coordenadas desde la aplicación.
- Dado que se conoce la posición de la cabeza, podría usarse identificación del rostro.

Kinect se ha mostrado como una herramienta con infinidad de campos de aplicación, es una tecnología reciente y debe mejorar mucho algunos aspectos técnicos como ajustes de ángulos de visión y saturaciones por luz solar, cada vez hay más equipos trabajando con esta tecnología y las mejoras del SDK oficial son periódicas, en el plazo de un año es muy probable que veamos un avance muy importante en la cantidad y calidad de los desarrollos

Notas y Referencias

- 1 http://elpais.com/diario/2011/04/30/madrid/1304162657_850215.html. Diario El Pais, 30-04-2011.
 - 2 En la base de datos de la OMS sobre la Carga Mundial de Morbilidad, las muertes y las lesiones no mortales relacionadas con caídas excluyen las caídas debidas a agresiones y lesiones auto provocadas, las caídas desde animales, edificios en llamas o vehículos de transporte y las caídas en fuegos, agua o maquinas
 - 3 Caídas, Nota descriptiva N°344 <http://www.who.int/mediacentre/factsheets/fs344/es/index.html>. OMS, Organización Mundial de la Salud. Agosto 2012.
 - 4 Journal of Gerontology: Medical Sciences: John E. Morley The Top 10 Hot Topics in Aging ,J Gerontol A Biol Sci Med Sci (2004) 59(1): M24-M33 doi:10.1093/gerona/59.1.M24
 - 5 Periférico que permite interactuar con el PC utilizando los movimientos de nuestro propio cuerpo o mediante comandos de voz. Incorpora un emisor de infrarrojos, dos cámaras RGB y cuatro micrófonos. Requisitos: procesador de doble núcleo a 2,66 GHz o superior y 2 Gbytes de memoria RAM. <http://www.microsoft.com/en-us/kinectforwindows/>
 - 6 Web of Science (ISI) Título: A neural network-based sensor for elder falling detection Autor: Pan, JI. Yung, CJ Liang, ICINCO 2006: PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS : 203-206 2006
 - 7 Web of Science (ISI) Advanced Patient or Elder Fall Detection based on Movement and Sound Data Autor: Doukas, C Autor/Editor alt.: Maglogiannis, I 2008 2ND INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING TECHNOLOGIES FOR HEALTHCARE : 96-100 2008
 - 8 Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument Volume 33, Issue 3, March 2012, Pages 575-580 Design of a wearable fall detection device Shi, X. , Zhang, T. College of Automation, Chongqing University, Chongqing 400030, China
 - 9 Recurso: Web of Science (ISI) Título: Evolutive Algorithms for Visual Fall Detection in Intelligent Spaces Autor: Canas, JM Autor/Editor alt.: Marugan, S Marron, M Garcia, JC Cita: IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE 2010) : 2307-2313 2010 Año: 201
 - 10 Recurso: Web of Science (ISI) Título: Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound, and Visual Perceptual Components Autor: Doukas, CN Autor/Editor alt.: Maglogiannis, I Cita: IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE 15 (2): 277-289 MAR 2011 Año: 2011
 - 11 Recurso: Web of Science (ISI) Título: Modeling Human Activity From Voxel Person Using Fuzzy Logic Autor: Anderson, D Autor/Editor alt.: Luke, RH Keller, JM Skubic, M Rantz, MJ Aud, MA Cita: IEEE TRANSACTIONS ON FUZZY SYSTEMS 17 (1): 39-49 FEB 2009 Año: 2009
-

12 Recurso: Web of Science (ISI) Título: Intelligent video surveillance for monitoring elderly in home environments Autor: Nasution, AH Autor/Editor alt.: Emmanuel, S Cita: 2007 IEEE NINTH WORKSHOP ON MULTIMEDIA SIGNAL PROCESSING : 203-206 2007 Año: 2007

13 septiembre de 2012

14 Title: Start here! Learn Microsoft Kinect API / Rob Miles. Publisher: Sebastopol, Calif. : O'Reilly Media, Creation Date: c2012. Identifier: (ISBN) 9780735663961 (ISBN) 0735663963 Language: English

15 Title: Natural user interfaces in .NET : WPF 4, Surface 2, and Kinect / by Joshua Blake. Greenwich, Conn.: Manning; London : Pearson Education [distributor], Creation Date: 2011. Identifier: (ISBN) 9781935182818 (pbk.) (ISBN) 1935182811 (pbk.) Language: English

16 Recurso: Web Kinect SDK 1.0 – 3 – Track users with the SkeletonStream by RENAUD on APRIL 19, 2012 <http://www.renauddumont.be/en/2012/kinect-sdk-1-0-3-tracker-les-mouvements-avec-le-skeletonstream>

17 Title: Programming With the Kinect for Windows Software Development Kit. Add Gesture and Posture Recognition to Your Applications.Contributor(s): [Catuhe, David](#).Publisher: Microsoft Pr. Creation Date: 2012. Identifier: (ISBN) 9780735666818 (ISBN) 0735666814 Language: English

18 Loren Arthur Schwarz, Artashes Mkhitarian, Diana Mateus, Nassir Navab, Human skeleton tracking from depth data using geodesic distances and optical flow, Image and Vision Computing, Volume 30, Issue 3, March 2012, Pages 217-226, ISSN 0262-8856, 10.1016/j.imavis.2011.12.001. (<http://www.sciencedirect.com/science/article/pii/S026288561100134X>)

19 Vintsyuk, T.K. "Speech discrimination by dynamic programming". Kibernetika, Vol. 4, pp. 81-88, Jan.-Feb. 1968 Sakoe, H. and Chiba, S., Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, 26(1) pp. 43– 49, 1978, ISSN: 0096-3518

20 L. R. Rabiner and B. Juang. Fundamentals of speech recognition. Prentice-Hall, Inc., 1993

21 Muller, M., Information Retrieval for Music and Motion, Ch. 4 (available online at http://www.springer.com/cda/content/document/cda_downloadaddocument/9783540740476-1.pdf?SGWID=0-0-45-452103-p173751818), Springer, 2007, [ISBN 978-3-540-74047-6](#)

22 C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal, 60(7):1389-1409, September 1981.

23 MSDN Library. Demonstrates how to implement Kinect for Windows in a variety of C# samples. <http://msdn.microsoft.com/en-us/library/hh855377.aspx>

24 Microsoft. Introduction to creating applications with a natural user interface.
<http://www.microsoft.com/en-us/kinectforwindows/develop/tutorials/tutorialsdesc.aspx?tutorialid=Tutorial2>