JÓVENES EN LA CIENCIA
REVISTA DE DIVULGACIÓN CIENTÍFICA

CLASIFICACIÓN DE ACTIVIDADES POR SISTEMAS AUTÓNOMOS

Rodríguez Salazar Jonathan Humberto (1), Almanza Ojeda Dora Luz (2)

1 [Licenciatura en Ingeniería en Sistemas Computacionales, Universidad de Guanajuato] | Dirección de correo electrónico: jh.rodriguezsalazar@ugto.mx

2 [Departamento de Ingeniería Electrónica, División de Ingenierías, Campus Irapuato - Salamanca, Universidad de Guanajuato] | Dirección de correo electrónico: dora.almanza@ugto.mx

Resumen

Mediante la captura de imágenes es posible encontrar patrones repetitivos, que a simple vista es sencillo distinguirlos como sentarse en una silla, sin embargo, una máquina no puede hacer este proceso de distinción tan fácilmente. Un sensor Kinect brinda la posibilidad de observar el entorno mediante la extracción de características de las imágenes RGB y de profundidad que adquiere. Las características calculadas mediante una computadora permiten detectar las actividades que una persona realiza. Así en este trabajo se capturan las acciones realizadas para su procesamiento, destacando la necesidad de encontrar las distancias entre 8 principales articulaciones del ser humano con respecto a una articulación de referencia, en nuestro caso la espina. De estas distancias y velocidades de las distancias se obtienen medias y desviaciones estándar para generar una clasificación mediante el clasificador de vecino más cercano (KNN). Además, se genera un módulo programado el cual es embebido a un sistema autónomo para en un futuro obtener una interacción real de un humano-máquina. Los resultados obtenidos presentan un asertividad de 74% para una base de datos capturada durante el desarrollo de este proyecto, que contiene 10 actividades realizadas por 27 personas y con 3 repeticiones de cada actividad.

Abstract

By capturing images, it is possible to find repetitive patterns, which at first glance is easy to distinguish as sitting on a chair, however, a machine can´t make this process of distinction so easily. A Kinect sensor provides the possibility to observe the environment by extracting characteristics of the RGB images and the depth it acquires. The characteristics calculated by a computer allow detecting the activities that a person performs. Thus, in this work, the actions taken for its processing are captured, highlighting the need to find the distances between 8 main articulations of the human being with respect to a reference articulation, in our case the spine. From these distances and velocities of the distances, means and standard deviations are obtained to generate a classification by means of the nearest neighbor classifier (KNN). In addition, a programmed module is generated which is embedded in an autonomous system to obtain a real interaction of a human-machine in the future. The results obtained have an assertiveness of 74% for a database captured during the development of this project, which contains 10 activities carried out by 27 people and with 3 repetitions of each activity.

Palabras Clave

Palabras Clave: clasificador KNN; Kinect ONE; características; articulaciones; base de datos.

INTRODUCCIÓN

La posibilidad de hacer sistemas autónomos que realicen el trabajo de un hombre siempre ha estado presente en la historia de la humanidad, y este principio es el que ha llevado a producir ramas de la ciencia enfocadas a generar y producir inteligencia artificial capaz de aprender de su entorno mediante diversos sensores.

En este trabajo tomamos como base los sensores Kinect ONE (ver IMAGEN 1) para crear un módulo programado que le permita a una computadora, a través de imágenes, aprender criterios de clasificación sobre acciones comunes en la sociedad, utilizando la librería OpenCV.

Con el fin de poder manipular juegos de una consola mediante el movimiento corporal del usuario fue inventado el dispositivo Kinect 360 en el año de 2009, y posterior mente en el año de 2014 fue lanzada la segunda versión Kinect ONE [1], el cual es el referente de esta investigación.

Las especificaciones de este dispositivo están enumeradas a continuación, haciendo una comparación con la versión Kinect 360 [2]:

1. Campo de visión: 70º en horizontal (antes 57º) y 60 en vertical (antes 43º).
2. Resolución: 1920 x 1080 Full HD (antes 640 x 480).
3. Rango de profundidad: 4,5 metros (antes 0.5 metros)
4. USB 3.0 (antes 2.0)
5. Captación de sonido: Eliminación del ruido ambiente
6. Captura de movimiento a oscuras.
7. Kinect ONE permite calcular/analizar la fuerza de nuestros músculos y medir el ritmo cardíaco.

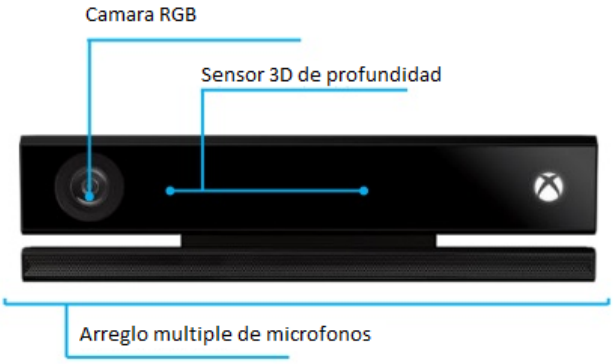

IMAGEN 1: Componentes del sensor Kinect ONE

El vecino más cercano (KNN)

Uno de los clasificadores fundamentales es el de los vecinos más cercanos o KNN [1], el cual se interesa por clasificar un nuevo objeto dependiendo del tipo de clase que sean los vecinos más próximos a él, teniendo en cuenta un parámetro K que es el margen de objetos próximos a tomar en cuenta.

Imaginemos un problema de separación de dos clases (IMAGEN 2), la Clase A llamada triángulo y la Clase B llamada cuadro, ahora pongamos atención en las líneas punteadas, están defindas por el parámetro K, el cual marca la cantidad de vecinos a considerar para tomar la decisión, dado que en K=4 no se puede tomar una decisión se extiende a K=7 y se genera el resultado del clasificador (IMAGEN 3).

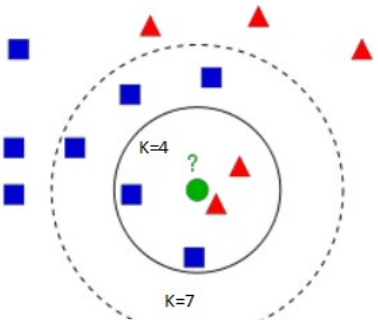

IMAGEN 2: Planteación del problema de KNN

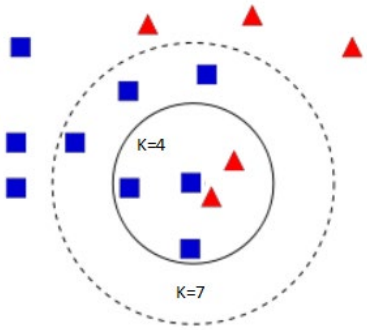

IMAGEN 3: Resolución del clasificador

OpenCV.

Es una librería open source para visión por computadora por lo tanto, es libre para uso académico y para algunas aplicaciones comerciales [1]. Tiene interfaces C++, Python y Java y es compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para una eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Escrito en C y C++ optimizado, la biblioteca puede tomar ventaja de procesamiento multi-core. Habilitado con OpenCL, puede tomar ventaja de la aceleración del hardware de la plataforma de cómputo heterogéneo subyacente.

MATERIALES Y MÉTODOS

Los materiales que fueron utilizados para la realización de este trabajo fue un sensor Kinect ONE y una computadora. Es importante especificar desde este momento que el trabajo se denomina Join a cada articulación detectada por el Kinect y ejemplificada en la IMAGEN 4, cada conjunto de datos se obtiene por cada frame, que es una captura de imagen del Kinect, de este modo cada frame cuenta con los datos (x , y , z) de 25 joins (de 0 a 24). La metodología planteada para el desarrollo del trabajo se muestra en el diagrama de la IMAGEN 5. La base de datos contiene 10 actividades realizadas por 27 personas y 3 repeticiones por persona. Las actividades planteadas fueron: Sentarse, Pararse, Aplaudir, Saludar, contestar el teléfono, Martillar, entre otras.

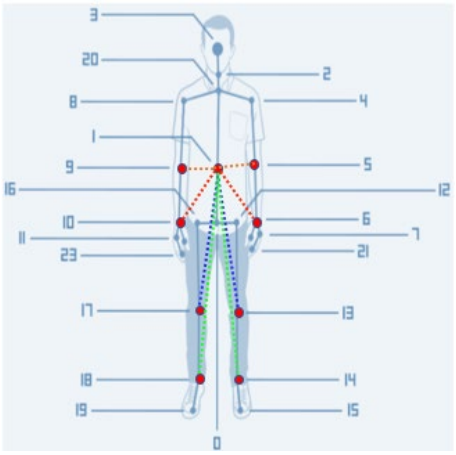

IMAGEN 4: Establecimiento de Joins y Distancias principales

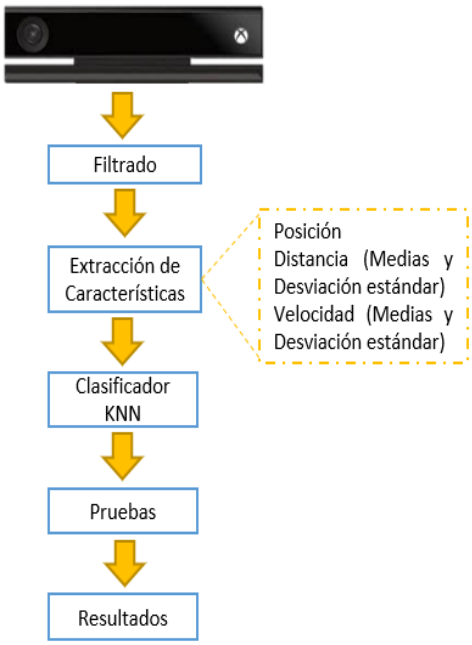

IMAGEN 5: Diagrama de la Metodología

Captura de Kinect

Para este apartado es necesaria la instalación de los controladores del Kinect (SDK) para poder mandarle instrucciones desde la

computadora, una vez teniendo la conexión establecida entre la computadora y el Kinect fue necesario elaborar un programa que genere un exoesqueleto sobre la persona capturada y para guardar las capturas de los datos que genera el Kinect, los cuales son imágenes en RGB, imágenes de profundidad, y los datos de posición en 3D de cada Join.

Filtrado

Esta etapa consiste en eliminar los frames para minimizar la cantidad de información que se capturo. Primero se crean archivos de intervalos de frames que si sirven con su respectiva separación de persona, actividad y repetición; y una segunda actividad de corrección de datos el cual consiste en tomar 5 frames para obtener promedio de los Joins y generar un nuevo punto de inicio con datos corregidos, tal cual la IMAGEN 6 lo muestra.

IMAGEN 6: Filtro de la Postura

Extracción de Características

Si volvemos a la IMAGEN 4 podemos observar que hay líneas marcadas desde el Joint central hasta los Joints de las extremidades, los cuales son: codos, muñecas y tobillos tanto izquierdo como derechos.

Con estos puntos obtenemos las distancias por frames y guardamos en un vector los promedios de las distancias como en la ecuación 1 y luego las desviaciones estándar de cada promedio de distancias como lo muestra la ecuación 2

$$d_j = [d_{1J}, d_{2J}, d_{3J}, d_{4J}, d_{5J}, d_{6J}, d_{7J}, d_{8J}] \quad (1)$$

$$\mu_j = [\mu_{d1}, \mu_{d2}, \mu_{d3}, \mu_{d4}, \mu_{d5}, \mu_{d6}, \mu_{d7}, \mu_{d8}] \quad (2)$$

$$\sigma_j = [\sigma_{d1}^2, \sigma_{d2}^2, \sigma_{d3}^2, \sigma_{d4}^2, \sigma_{d5}^2, \sigma_{d6}^2, \sigma_{d7}^2, \sigma_{d8}^2] \quad (3)$$

A continuación, sigue el mismo procedimiento, pero ahora calculando las velocidades como muestra la ecuación 3:

$$v_{i(j+1)} = \left|d_{i(j+1)} - d_{ij}\right| \quad i = 1,2,3,\dots,nd \quad j = 1,2,3,\dots,N\ frames \quad (4)$$

$$v_j = [v_{1J}, v_{2J}, v_{3J}, v_{4J}, v_{5J}, v_{6J}, v_{7J}, v_{8J}] \quad j = 1,2,3,\dots,N\ frames \quad (5)$$

$$\mu_j = [\mu_{v1}, \mu_{v2}, \mu_{v3}, \mu_{v4}, \mu_{v5}, \mu_{v6}, \mu_{v7}, \mu_{v8}] \quad (6)$$

$$\sigma_j = [\sigma_{v1}^2, \sigma_{v2}^2, \sigma_{v3}^2, \sigma_{v4}^2, \sigma_{v5}^2, \sigma_{v6}^2, \sigma_{v7}^2, \sigma_{v8}^2] \quad (7)$$

Así al final obtenemos un vector de 32 elementos (8 promedios de distancias y 8 de desviaciones estándar e igual con la velocidad)

Clasificador Vecino más Cercano (KNN)

Los datos obtenidos de la extracción de características ecs. (2), (3), (6) y (7) son procesadas por el clasificador KNN para ello se requiere un programa con la capacidad de leer los archivos generados y desarrolla el método en código para el procesamiento. Al ejecutar el KNN se necesitan dos partes, la primera es el *training*, que como su nombre lo especifica es la etapa en la que se entrena y se busca el mejor parámetro K y la segunda es el *tests* que es con el cual probamos la eficiencia del clasificador.

RESULTADOS Y DISCUSIÓN

Las clases evaluadas son: Sentarse (0), Pararse (1), Aplaudir (2), Saludar (3) y Martillar (4), podemos observar en los resultados obtenidos en la Tabla 1 que se tiene una precisión de 91% y 82%, para las clases Aplaudir y Martillar, respectivamente. Las clases Pararse y Sentarse son las que más confunde el clasificador. Al probar con 10 clases este mismo clasificador, pero con 27 personas por actividad, se obtiene un porcentaje del 74%.

Tabla 1: Matriz de confusión para 5 clases usando KNN

	0	1	2	3	4
0	7 64%	4 36%	0 0%	0 0%	0 0%
1	4 36%	7 64%	0 0%	0 0%	0 0%
2	1 9%	0 0%	10 91%	0 0%	0 0%
3	1 9%	0 0%	0 0%	8 73%	2 18%
4	1 9%	0 0%	0 0%	1 9%	9 82%

CONCLUSIONES

El trabajo ha tenido un resultado esperado, en el cual se clasifican 5 clases de manera automática y que puestas en un dispositivo será capaz de discriminar las acciones entrenadas, hace falta crecer la base de datos y agregar más acciones a clasificar, asi mismo probar otros clasificadores, cabe mencionar que se probó tambien el clasificador SVM y no se obtuvieron resultados acordes a lo esperado. puesto que se hace con la misma mano agrega confusión

AGRADECIMIENTOS

Los autores expresan su agradecimiento a todos los voluntarios que participaron en la captura de las actividades durante la realización de este trabajo. Jonathan Rodríguez agradece a DAIP y a la Dra. Dora Luz Almanza Ojeda, asesora del proyecto, por sus apoyos y por permitir su participación en el verano UG.

REFERENCIAS

[1] Meisener Jeffrey (2013). Collaboration, expertise produce enhanced sensing in Xbox One. The Official Microsoft Blog. News & Perspectives. 20 de Junio de 2018.Recuperado de https://blogs.technet.microsoft.com/microsoft_blog/2013/10/02/collaboration-expertise-produce-enhanced-sensing-in-xbox-one/
[2] Murillo Alejandro, Marcal Montserrat (2013). Características Kinect 2. Kinect for Devolpers. 20 de Junio de 2018. Recuperado de http://www.kinectfordevelopers.com/es/2014/01/28/caracteristicas-kinect-2/
[3] OpenCV (2018). Understanding K-Nearsted. Docs OpenCV 3.4.1. 27 de Junio de 2018.Recuperado de https://docs.opencv.org/3.4.1/d5/d26/tutorial_py_knn_understanding.html
[4] OpenCV (2018). Home. OpenCV. 15 de Julio de 2018.Recuperado de https://opencv.org/
[5] Oros Flores Marvella Izamar (2018). Diseño e implementación de una librería para la caracterización de movimiento en 3D usando Kinect. Tesis Profesional de la Universidad de Guanajuato.