

Construcción de un sistema de teclado virtual como apoyo pedagógico para personas con habilidades motrices limitadas

Eduardo Emmanuel Ochoa Gracia, José Federico Ramírez Cruz,
Miquelina Sánchez Pulido y Blanca E. Pedroza Méndez

Departamento de Sistemas y Computación,
Instituto Tecnológico de Apizaco, México

`e_8a@hotmail.es, federico_ramirez@yahoo.com.mx`

Resumen En este artículo se describe el desarrollo de un sistema que hace uso de técnicas de visión por computadora e interacción humano computadora que permita a usuarios con dificultades motrices interactuar con computadoras. El sistema está pensado para funcionar de manera integral con el sistema operativo potenciando así el proceso de enseñanza/aprendizaje en los usuarios mediante el uso de las demás aplicaciones contenidas en el sistema operativo y el acceso a internet. Para lograr la interacción se utiliza un sensor Kinect cuyas características propias facilitan el procesamiento de imágenes del entorno.

Palabras clave: visión por computadora, educación especial, nuevas tecnologías, Kinect, interactividad.

1. Introducción

Con la aparición de nuevas tecnologías tales como la realidad aumentada o la realidad virtual, nuevas oportunidades de aplicación de éstas aparecen como se ve en [1]. La educación es uno de los campos que más puede beneficiarse de estas nuevas tecnologías [2] y actualmente se pueden ver numerosos proyectos alrededor del mundo que se enfocan en la introducción de nuevos sistemas de enseñanza-aprendizaje; sin embargo, existe un área de la educación que carece de aplicaciones para estas nuevas tecnologías, la Educación Especial. Incluso, existiendo muchos sistemas enfocados a la rehabilitación como el propuesto por Da Gama y colaboradores [3], existen pocos enfocados a propósitos educacionales. Además, hay mucho menos aplicaciones enfocadas al aspecto social de la educación.

Este trabajo se centra en el desarrollo de un sistema educacional y social para personas con limitaciones motoras que son incapaces de interactuar específicamente con computadoras.

Nuestro objetivo no es solo hacer un sistema de una sola interfaz, sino uno que trabaje con todo el Sistema Operativo de manera que el usuario incapacitado sea

capaz de manipular la mayoría de las aplicaciones disponibles para éste. Todo lo anterior piensa lograrse mediante el aprovechamiento de las características particulares de las cámaras RGB-D (de color y profundidad), en éste caso un sensor Kinect de Microsoft © el cual está compuesto por una cámara RGB, un proyector infrarrojo y un sensor infrarrojo, así como un conjunto de micrófonos y un motor de inclinación para ajustar el ángulo del dispositivo como se muestra en la figura 1.

La función principal del sistema es la de facilitar la interacción usuario-computadora a usuarios con problemas motrices mediante un teclado cuyo hardware no esté en contacto directo con el usuario.

En la segunda sección de éste trabajo se describirán los procesos realizados para solucionar el problema presentado. En la subsección 2.1 describiremos el proceso de calibración de los sensores del dispositivo Kinect pues existe cierta disparidad entre las imágenes obtenidas por los sensores IR y RGB dados los ángulos de visión de ambos, en la subsección 2.2 se tratará el proceso de solución del seguimiento de la mano mediante el procesamiento de los datos de profundidad y en la subsección 2.3 se hablará de la identificación del teclado mediante el procesamiento de la imagen RGB y el uso de un algoritmo de filtrado de colores. En la tercera sección se muestran los resultados obtenidos y finalmente en la cuarta sección se darán a conocer las conclusiones a las que se ha llegado hasta éste punto del desarrollo.



Fig. 1. Dispositivo Kinect.

2. Solución propuesta

Dado que el sistema busca hacer las funciones de un teclado, la solución del problema se dividió en tres etapas: 1.- Seguimiento de las manos, 2.- Identificación del teclado físico y reconocimiento de las teclas y 3.- Integración con el sistema operativo. En éste artículo se tratarán únicamente las primeras dos etapas.

Para poder realizar cualquiera de las dos funciones antes mencionadas el sensor debe estar enfocando tanto las manos del usuario como el teclado sin que existiera ningún obstáculo entre el dispositivo y estos últimos, por tanto se definió un umbral de visión para el sensor de profundidad que permitiera al sistema ver únicamente los objetos en movimiento (fig. 2.) eliminando el teclado ya que éste será identificado en

la imagen RGB. Dadas las características del sensor de profundidad, éste debe estar situado a más de 40cm de distancia del objeto a seguir de manera que éste sea visible. Una vez establecido el umbral de visión del sensor infrarrojo se debe proceder a realizar el proceso de calibración del sensor IR con el fin de empatar tanto las imágenes RGB como las obtenidas de los datos de profundidad; dicho proceso se detalla en la subsección 2.1.

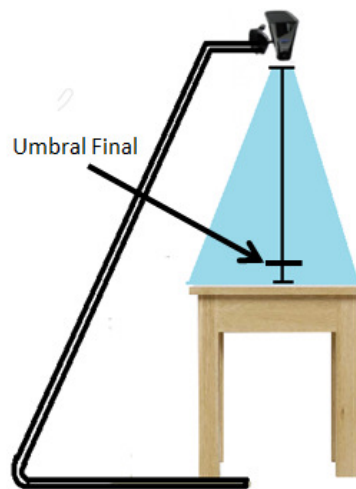


Fig. 2. Posición y distancia del dispositivo y umbral de visión final para el sensor infrarrojo.

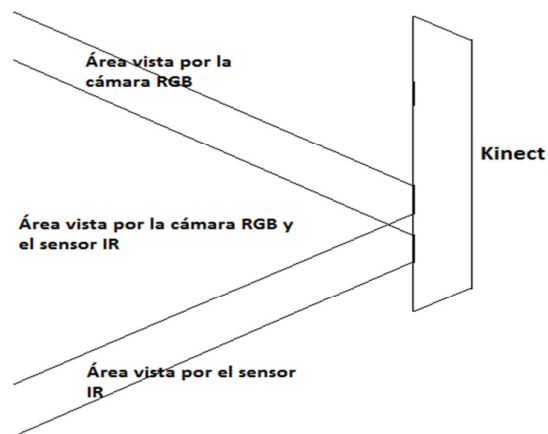


Fig. 3. Disparidad entre área de visible de la cámara RGB y el área visible del sensor infrarrojo.

2.1. Calibración

Con el fin de que los objetos tanto en la imagen de color como en la de profundidad coincidan, fue necesario realizar un proceso de calibrado propio del dispositivo Kinect dado que el sensor infrarrojo y la cámara de color captan distintos ángulos de la misma imagen como lo muestra la figura 3.

Este proceso de calibrado es crucial para el correcto funcionamiento del sistema pues si no se lleva a cabo la posición de la mano sobre el teclado no será obtenida correctamente y ocasionaría problemas en la interacción con el usuario (Esto podría considerarse como una versión simplificada del problema presentado en [4]).

Para solucionar éste problema se hace uso del método `GetAlternativeViewPointCap()` de la función `xn::DepthGenerator` contenido en la librería OPENNI que permite realizar la calibración del sensor IR en relación al sensor RGB.

Ya calibrado el sensor IR es posible iniciar el desarrollo del algoritmo de seguimiento de las manos del usuario como se describe en la subsección siguiente.

2.2. Identificación y seguimiento de la mano

En este punto se llegó a la conclusión de que lo mejor era evitar el proceso de identificación de la mano dadas las características cambiantes de los casos de discapacidad motriz, que va desde problemas de motricidad fina hasta problemas más graves que podrían influir en la forma y posición de la mano y dedos, incluyendo la posibilidad de que el usuario necesite de alguna herramienta para poder interactuar con el teclado como una prótesis, cuya forma no corresponda con la de la mano humana, por lo que el proceso de identificación podría presentar dificultades para encontrar los objetos a seguir.

Para la identificación de los objetos a seguir se definió un umbral de distancia de visibilidad de los objetos en la imagen y en base a éste se realizó la segmentación de los objetos, eliminando todo aquello que estuviese más allá de la distancia definida con el umbral de visibilidad a fin de evitar interferencias en la imagen. Una vez teniendo los objetos segmentados en la imagen éstos son etiquetados para así poder trabajar cada uno por separado; después, se crea una envoltura convexa para cada objeto con el fin de reducir sus características y encontrar el centro geométrico (centroide) de cada objeto [5] mediante la aplicación de la ecuación (1), en donde x y y son las coordenadas de cada pixel i en la región de la imagen en que se trabaja; mientras que k corresponde al número de pixeles contenidos en la región. El centroide de cada región se utiliza como punto de referencia para eliminar el resto del brazo en la imagen dejando sólo el área de la imagen correspondiente a la mano o, en su defecto, al extremo del objeto a seguir (ver la figura 4).

$$\bar{x} = \frac{\sum_{i=0}^k x_i}{k}, \bar{y} = \frac{\sum_{i=0}^k y_i}{k} \quad (1)$$

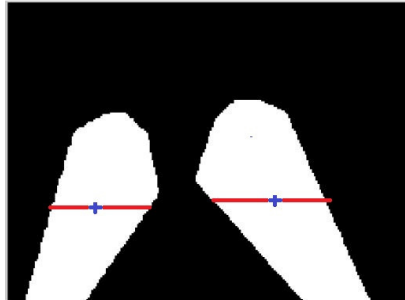


Fig. 4. Primeros centroides encontrados en cada región segmentada, utilizados para recortar las imágenes.

Después de este proceso se obtiene por segunda vez el centroide de las regiones obtenidas tras reducir el área a seguir como se muestra en la figura 5.

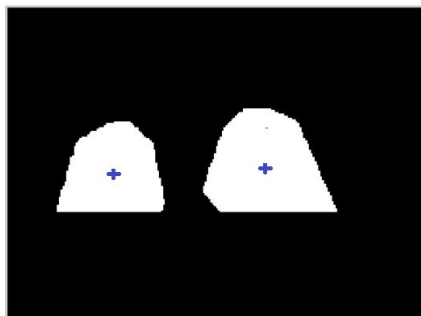


Fig. 5. Segundos centroides encontrados en las imágenes recortadas y utilizados como puntos a seguir en la imagen.

Al ser éste proceso realizado independientemente para cada región en la imagen logramos seguir individualmente cada objeto y no interfieren el uno con el otro a menos que los objetos se unan en cierto momento o que alguno desaparezca del cuadro, en cuyo caso sólo se seguirá el objeto restante en la imagen.

Una vez solucionados los problemas de la calibración del sensor IR y del seguimiento del usuario es posible adentrarse en el desarrollo del algoritmo de identificación del teclado y su integración con el seguimiento del usuario como se detalla en la subsección 2.3.

2.3. Identificación del teclado físico

En el caso del teclado se necesitan definir características que servirán para reconocerlo, a partir de dichas características se establecen los puntos de anclaje de los que se obtendrán las proporciones para cada tecla, siendo éste método invariable al escalamiento.

Mediante un filtro de color se elimina de la imagen todo aquello que no sea de utilidad dejando únicamente las áreas que cumplen con los requisitos necesarios para permanecer en la imagen como se observa en la figura 6b en cuyo caso se optó por filtrar los tonos rojos, eliminando toda mezcla de colores en las que el color rojo sea menor que el verde y azul, así como cuando la diferencia entre los tonos azules y verdes y los tonos de rojo sea superior a 100 en una escala de 255 dado el esquema de color RGB, dejando visibles únicamente los cuadros a las esquinas del patrón de prueba.

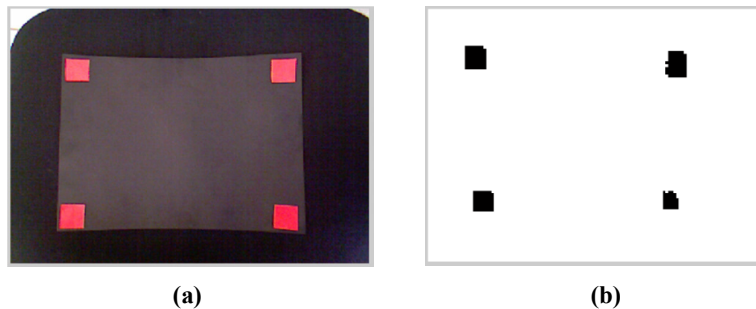


Fig. 6. Prueba de detección de los puntos de anclaje con un filtro de color; (a) Imagen vista por la cámara de color; (b) segmentación de los puntos de referencia para ubicar el teclado.

Después de aplicar el filtro de color para detectar los puntos de anclaje, se etiquetan las áreas segmentadas en la imagen para encontrar el centro de cada una de éstas. Utilizando como referencia un eje cartesiano, se realiza la diferencia de las distancias entre los objetos superiores sobre el eje X (ancho) y otra entre los dos objetos ubicados a la izquierda de la imagen sobre el eje Y (alto). Las distancias obtenidas se dividen entre el número de teclas contenidas en el teclado a lo ancho y a lo alto, obteniendo así las áreas activas para cada tecla (figura 7), las cuales variarán en base a la distancia entre el teclado y la cámara pero sin perder la ubicación de las teclas.

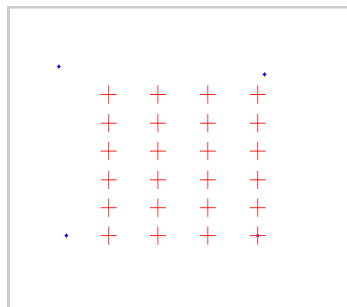


Fig. 7. Áreas activas de las teclas

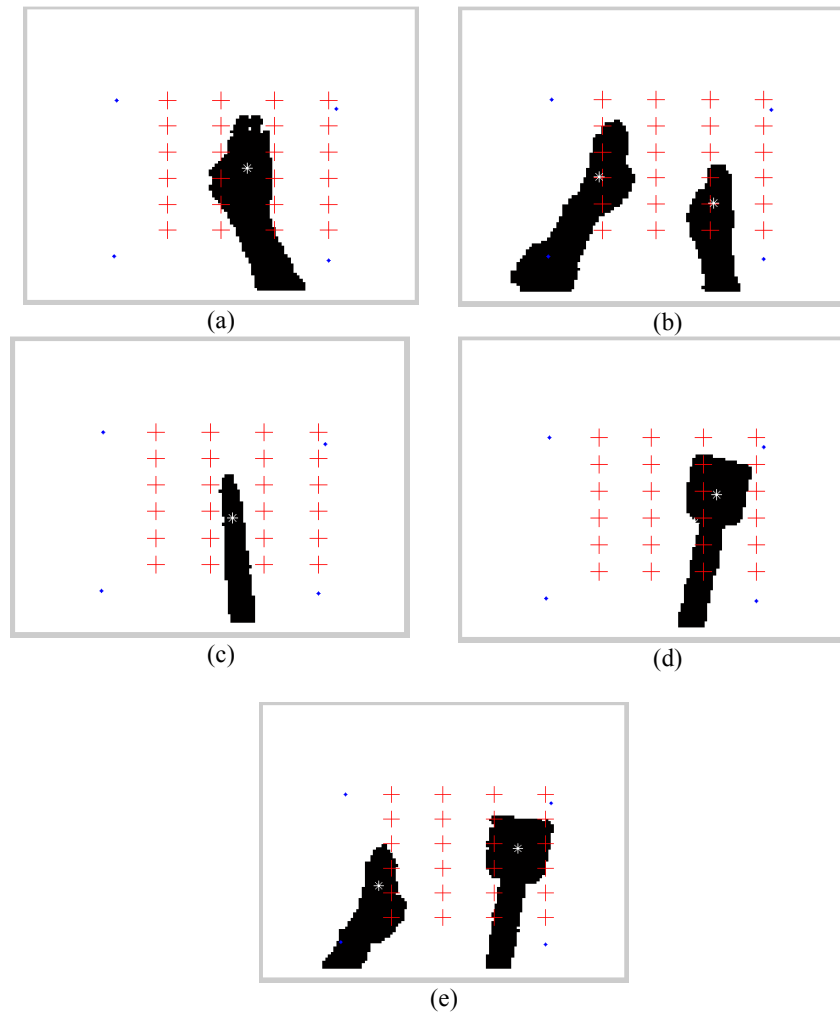


Fig. 8. Resultados del seguimiento con las manos y diversos objetos: (a) Una sola mano; (b) dos manos; (c) Un objeto cilíndrico delgado; (d) Una pala y (e) Una pala y una mano.

3. Resultados preliminares

Para las pruebas se utilizó un patrón rectangular de 100 x 70 cm., con 4 cuadrados rojos en las esquinas de 2x2 cm., un dispositivo Kinect situado a 70 cm. de distancia del patrón de prueba y diversos objetos.

Al aplicar ambos algoritmos, tanto el de seguimiento de objetos como el de detección del teclado e imprimir los objetos de interés en una sola imagen obtenemos los resultados observados en la figura 8. En la figura 8a se muestra la presencia de

una sola mano en negro; se indican los centros de cada tecla con las cruces rojas, y el centroide de la mano con una estrella blanca dentro de la mano. En la figura 8b se muestran dos manos con las mismas características. En la figura 8c se utilizó un utensilio (un objeto cilíndrico delgado) en el cual también se muestran las mismas características que en las figuras anteriores. En la figura 8d se muestra la presencia de una pala y por último en la figura 8e se muestra una mano y la pala.

Como se ve en la serie de imágenes de la figura 8 el sistema detecta cualquier utensilio utilizado para hacer interacción con el teclado virtual y no habrá algún inconveniente por la forma que tengan estas herramientas.

Hasta ahora sólo se ha realizado la detección de las manos o utensilios utilizados para interactuar con el teclado; sin embargo, aún falta realizar el proceso de activación de las señales o comandos necesarios para emular el teclado de la computadora. Para detectar la interacción de la mano con la tecla se utilizará la imagen de profundidad proporcionada por el sensor y se tratará de detectar el movimiento vertical de la mano. Este es un proceso un poco complejo que se está resolviendo.

4. Conclusiones y trabajos futuros

El presente trabajo presentó los avances de la realización de un teclado virtual utilizando una cámara de profundidad y una cámara de color, incluidos en el sensor Kinect de Microsoft. El sistema realizado hasta ahora presenta buen desempeño tanto en el seguimiento de las manos del usuario o utensilios utilizados para interactuar con él, así como en la detección del teclado físico, asignando satisfactoriamente las áreas activas para cada tecla. La segmentación de los objetos se realizó con la cámara de profundidad (IR) y la identificación del teclado con la cámara de color (RGB). El sistema se probó con movimientos en tiempo real sin tener problemas de ralentización y fue implementado utilizando la herramienta Matlab.

En trabajos posteriores se procederá a detectar el movimiento vertical para detectar la interacción física de la mano o el utensilio con el teclado, además de realizar la interacción con el Sistema Operativo de la computadora y realizar las pruebas correspondientes de desempeño.

Referencias

1. DePriest, D.: The Fifth Dimension: How Augmented Reality is Launching Worlds within, TCC Proceedings 2012, pp. 6-13 (2012).
2. Justina Hsu, H.: The potential of Kinect in education. International Journal of Information and Education Technology, pp. 356-370 (2011)
3. Da Gama, A., Chaves, T., Figueiredo, L., and Teichrieb, V.: Poster: Improving motor rehabilitation process through a natural interaction based system using Kinect sensor. 3D User Interfaces (3DUI), 2012 IEEE Symposium, pp. 145-146 (2012)

4. Nooan, Philip J., Hallet, William A.: The Design and Initial Calibration of an Optical Tracking System Using the Microsoft Kinect, 2011 IEEE Nuclear Science Symposium Conference Record, pp. 3614-3617 (2011)
5. Park, H.: A Method for Controlling Mouse Movement using a Real-Time Camera.
6. Barilovits, D., DePriest, K.: LIVE: Xbox Kinect's Virtual Realities to Learning Games, TCC 2011 Proceedings, pp. 48-54 (2011)

