

INTÉRPRETE DE LENGUAJE DE SIGNOS PARA LA COMUNICACIÓN DE PERSONAS CON DISCAPACIDAD AUDITIVA EMPLEANDO PROCESAMIENTO DE IMÁGENES

INTERPRETER SIGN LANGUAGE FOR COMMUNICATION
HEARING IMPAIRED USING IMAGE PROCESSING

Jaime-Alberto Parra-Plaza¹

¹ Grupo de Investigación GICAT, Institución Universitaria Antonio José Camacho, UNIAJC, Cali, Colombia.
E-mail: jparra@admon.uniajc.edu.co

Recibido: 3 Diciembre

Aceptado: 15 Diciembre

*Correspondencia del autor. Jaime-Alberto Parra-Plaza. Grupo de Investigación GICAT, Institución Universitaria Antonio José Camacho, UNIAJC, Cali, Colombia. E-mail: jparra@admon.uniajc.edu.co

RESUMEN

Se presenta una herramienta computacional basada en procesamiento de imágenes para asistir a las personas que presentan discapacidad auditiva. La herramienta está en capacidad de capturar una secuencia de imágenes que corresponde a los gestos que el usuario hace con sus manos para realizar alguno de los signos empleados para la comunicación no verbal. La herramienta somete esta información a una serie de etapas de procesamiento con miras a identificar apropiadamente de qué gesto se trata y a emitir su equivalente textual. Los resultados obtenidos muestran que se logró identificar correctamente y en diversidad de situaciones los signos correspondientes al lenguaje de signos americano, el más empleado a escala mundial y base del lenguaje colombiano. Más aún, la herramienta permite que el usuario cree sus propios signos, siendo de esta manera una herramienta no sólo de ayuda, sino también de soporte para la ulterior investigación en el área.

Palabras claves: *Comunicación no-verbal, Detección de gestos, Discapacidad auditiva, Lenguaje de signos, Procesamiento de imágenes.*

ABSTRACT

It is presented a computational tool based on image processing conceived to assist people with hearing disability. The tool is able to capture a sequence of images corresponding to the gestures the user makes with his/her hands to perform any of the signs used for nonverbal communication. The tool submits this information to a series of processing steps in order to properly identify which gesture is and to present an equivalent, textual information. The results show that the signs for the American Sign Language, the worldwide most employed language and base for the Colombian language were identified correctly and in a variety of situations. Moreover, the tool allows the user to create their own signs, thus being not only an assistant tool, but also a support environment for further research in the area.

Key Words: *Gesture recognition, hearing impairment, image processing, non-verbal communication, sign language.*

I. INTRODUCCIÓN

La comunicación es un aspecto fundamental en la vida del ser humano, quien como ser social requiere transmitir eficientemente ideas y emociones a sus congéneres. La manera fundamental de comunicación que suelen emplear los seres humanos es la verbal, en donde el mensaje que se desea transmitir se transforma, en el tracto vocal, en una secuencia de movimientos que, a su vez, se expresan como una secuencia de sonidos, cuya composición, al interpretarse por el oído y cerebro del escucha, contiene el mensaje objeto de la comunicación. Infortunadamente en algunas personas se presenta una incapacidad para la generación de estas secuencias sonoras, debido ya sea a una dificultad funcional en el mecanismo vocal o en el sistema auditivo, el cual impide que el individuo desarrolle la capacidad de verbalización.

Sin embargo el ser humano no se rinde ante la adversidad y es así como las personas que presentan esta discapacidad han desarrollado a lo largo del tiempo un lenguaje equivalente que les permita mantener niveles adecuados de expresividad. Este lenguaje es el denominado lenguaje de signos o señas, en el cual, mediante un conjunto de gestos realizados primordialmente con las manos, el emisor transmite su mensaje empleando, en vez del canal vocal-auditivo, un canal gesto-viso-espacial. Cabe anotar que el lenguaje de signos, más que un simple conjunto de códigos para designar objetos o ideas básicas, es en propiedad un lenguaje natural con una estructura gramatical perfectamente definida.

Un aspecto implícito en este mecanismo de comunicación es que tiende a ser restringido, en el sentido de que, al requerir un conocimiento especializado tanto para elaborarlo como para entenderlo, se opone al deseo de ser un mecanismo generalizado de comunicación, pues sus usuarios terminan empleándolo entre ellos mismos, aislándose por ende de la sociedad en general, de tal manera que los hablantes vocal-auditivos en términos generales no entienden qué es lo que desean expresar mediante los signos que emplean o, en algunos casos, consideran, por falta de información o por falta de interés, a dicho lenguaje como una especie de curiosidad más que como un mecanismo alternativo, pero igualmente válido de comunicación. Afortunadamente, la tecnología ofrece posibilidades

para intervenir en este tipo de situaciones y lograr restablecer el objetivo primordial de la comunicación.

En el caso del presente proyecto, se desarrolló una plataforma de interacción a través de la cual los gestos que el usuario del lenguaje de signos realiza son captados mediante una cámara y en tiempo real procesados computacionalmente, de modo que el usuario puede expresarse de manera natural, tal como lo hace cuando interactúa con otros usuarios del lenguaje. A través de técnicas de procesamiento de imágenes, junto con algoritmos para detección de contornos y mecanismos de clasificación automática, se logró un sistema que es capaz de detectar de manera apropiada y eficiente el conjunto de gestos del lenguaje de signos americano empleado en Colombia y que tiene la posibilidad de ser expandido y entrenado para poder detectar nuevos gestos arbitrarios, ya sea estándares o creados por el usuario.

A. Procesamiento digital de imágenes

Hace referencia a las diferentes estrategias y métodos que se emplean para analizar, modificar y/o extraer información contenida en estructuras de datos de tipo matricial, cuyo contenido representa en general intensidades de iluminación o de color en un área bi o tridimensional. Las etapas que suelen asociarse con este procesamiento son (1): 1. *Adquisición* de la imagen mediante algún dispositivo o sensor de captura que haga posible su digitalización y en donde la calidad de las imágenes tomadas depende de la precisión de la digitalización de este sensor. 2. *Pre-procesamiento*, en la cual se busca mejorar la calidad de la imagen mediante técnicas como atenuación de ruido o aumento del contraste (2). 3. *Segmentación*, en donde se aplican algoritmos para separar la imagen en sus componentes, buscando con esto discriminar partes que sean de interés para la aplicación particular. 4. *Descripción* o selección de rasgos, que permite aislar información de interés para la identificación de patrones, por ejemplo, en el caso presente, de gestos o signos realizados con las manos. 5. *Reconocimiento*, con miras a interpretar los resultados de las fases anteriores y compararlos con referentes en bases de datos de conocimiento buscando correspondencias, en general basadas en patrones (3). La Figura 1 muestra el flujo de información y las interacciones que se dan entre estas etapas.

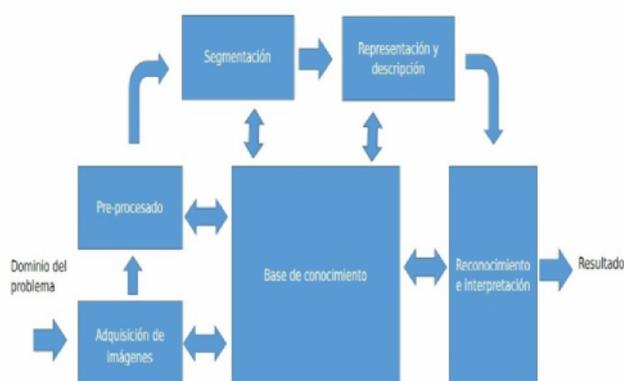


Figura 1. Etapas del procesamiento digital de imágenes.

Dos de las etapas que tienen mayor importancia para este trabajo en particular son la *detección de contornos* y la *clasificación de patrones*. Es por ello que los métodos empleados para ellas, *convexhull* y *k-nearest* respectivamente, se explicarán con mayor detalle a continuación.

El algoritmo de *convexhull* (4) se basa en el concepto matemático que define la envolvente convexa o envoltura convexa de un conjunto de puntos X de dimensión n como la intersección de todos los conjuntos convexos que contienen a X . Dados k puntos, su envolvente convexa C se calcula mediante la expresión dada en la Ecuación 1.

$$C(X) = \left\{ \bigcap_{i=1}^k \{x_i\} \mid x_i \in X, \alpha_i \in \mathbb{R}, \sum_{i=1}^k \alpha_i = 1 \right\} \quad (1)$$

En el caso particular de puntos en un plano, si no todos los puntos están alineados, entonces su envolvente convexa corresponde a un polígono convexo cuyos vértices son algunos de los puntos del conjunto inicial de puntos. Una forma intuitiva de ver la envolvente convexa de un conjunto de puntos en el plano, es imaginar una banda elástica estirada que los encierra a todos. Cuando se libere la banda elástica tomará la forma de la envolvente convexa.

El algoritmo de *k-nearest* (5) es un método no paramétrico ampliamente usado para clasificación de patrones. El método recibe un conjunto de ejemplos y, en la medida en que se va agregando un nuevo ejemplo, lo clasifica en relación al ejemplo o grupo de ejemplos previos que sea más cercano a él. La cercanía se da en términos de las variables, expresadas en dimensionalidad de características, que identifican las muestras. A manera de ejemplo para el caso de imágenes, considere un conjunto de muestras que repre-

sentan escrituras ya sea del número cero o del número cinco, y considere como características el número de segmentos y la excentricidad para cada imagen. En la medida en que el algoritmo procesa las muestras, se presentará una clara distinción entre dos centros de atracción, uno para cada uno de los dígitos (ver Figura 2).

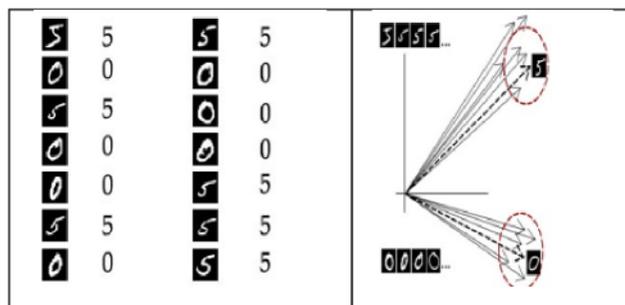


Figura 2. K-nearest. Izq: Muestras; Der: Clasificación.

B. Lenguaje de signos o de señas

Es el mecanismo de comunicación usado por las personas con discapacidad oral o auditiva el cual les permite, usando expresiones corporales, signos y gestos, comunicarse con otras personas y entre ellos. El lenguaje más difundido a este respecto es el lenguaje americano de signos o ASL (6). La lengua de señas colombiana, basada en la americana, es la lengua utilizada por la comunidad sorda de nuestro país y fue reconocida oficialmente en el año 1996. Los trabajos realizados en Colombia consolidan un total de 1200 señas para el año 2000, aunque las letras y los números constituyen los signos más básicos y de mayor empleo tanto por sordos como por personas que desean comunicarse con ellos (7). El lenguaje se basa en la generación de gestos, usualmente con las manos, para hacer referencias a signos individuales o a conceptos. La Figura 3 ilustra estos dos casos.

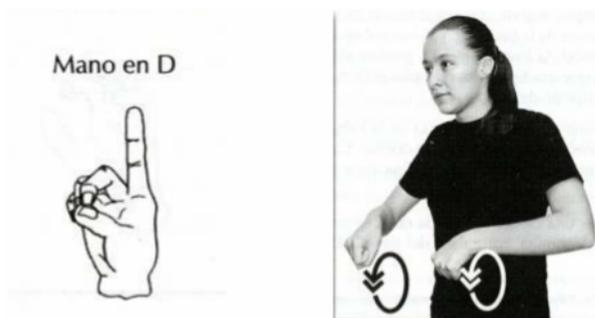


Figura 3. Gestos. Izq: Letra D; Der: Bicicleta.

En este documento se explican los pasos seguidos para desarrollar una herramienta computacional que permita detectar, analizar e identificar signos generados por movimientos corporales, primordialmente

con las manos y que corresponden a los signos del lenguaje de signos utilizados por personas con dificultades para emitir y/o captar señales verbales. Posteriormente se presentan y analizan las pruebas realizadas para validar la herramienta y se concluye con las conclusiones emanadas del trabajo.

II. MATERIALES Y MÉTODOS

El resultado de este proyecto es una herramienta computacional que realiza todas las etapas necesarias para ir desde la captura de las imágenes asociadas con un gesto hasta la interpretación del mismo y su posterior presentación en forma textual. El sistema procesa video en tiempo real y, a través de los módulos que lo componen, es capaz de reconocer, analizar e interpretar los signos del lenguaje de señas americano, base del lenguaje empleado en Colombia. El sistema cuenta con cinco grandes módulos: Adquisición y pre-procesamiento, Segmentación, Entrenamiento, Interpretación y Reconocimiento.

El sistema fue desarrollado en lenguaje C++ y se basa en una interacción entre tres componentes: código propio implementado, bibliotecas de acceso libre para procesamiento de imágenes y bibliotecas para implementación de plataformas. La aplicación se ejecuta

en un computador bajo el sistema operativo Windows y emplea como elemento para captación una cámara RGB o un *Kinect* (ver Figura 4).

El diseño parte de analizar los casos de uso, diagramas de componentes y diagramas de clases (ver Figura 5). La clase *CaptureDevice* es la base de herencia para otras dos clases, *Camera* y *Kinect*, para la interacción del Sensor y el sistema con métodos como *start*, *stop* y *getframe* que da la imagen tomada. Para el mejoramiento de la imagen en aspectos como control de brillo, contraste e intensidad, se tiene la clase *ImageProcessor* y su derivada *ColorProcessor*. Para la ecualización y para resaltar los colores se tiene la clase *HistogrammImage3D*. Para dividir o segmentar la imagen se tienen las clases *HSVBackgroundProcessor*, *YCbCrBackgroundProcessor* y *LearningModeBackgroundProcessor*, que heredan de *BackgroundProcessor*. Ésta es muy importante, pues su método *classify* es el encargado de usar el algoritmo *k-nearest neighbor* para la clasificación. Para la selección de rasgos o descripción se extraen rasgos con información cuantitativa de interés que sean fundamentales para la clasificación de las señas como cantidad de dedos visibles, defectos en el contorno de la mano, agujeros, etc., con ayuda del algoritmo *convexhull* (8).

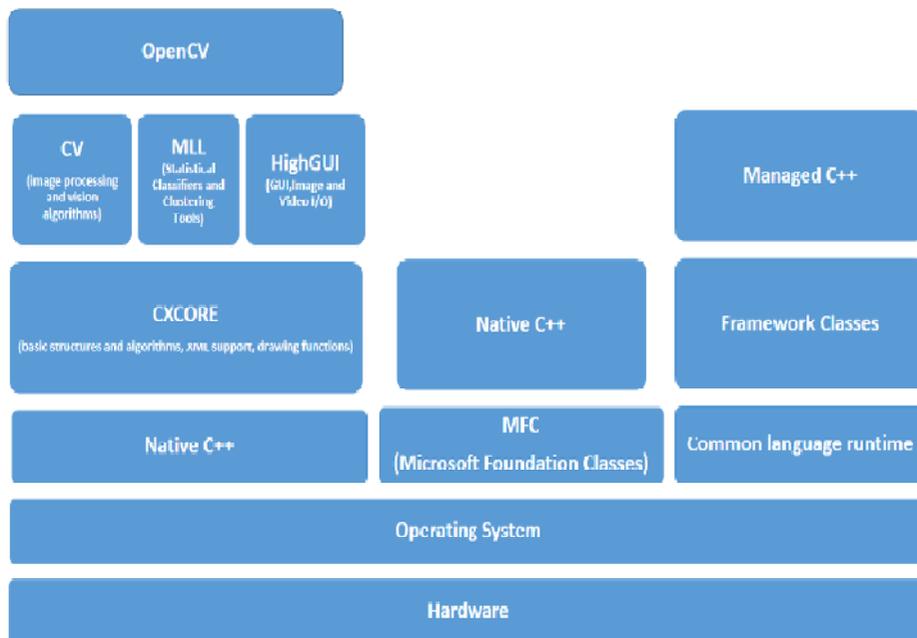


Figura 4. Pila de tecnologías de la aplicación.

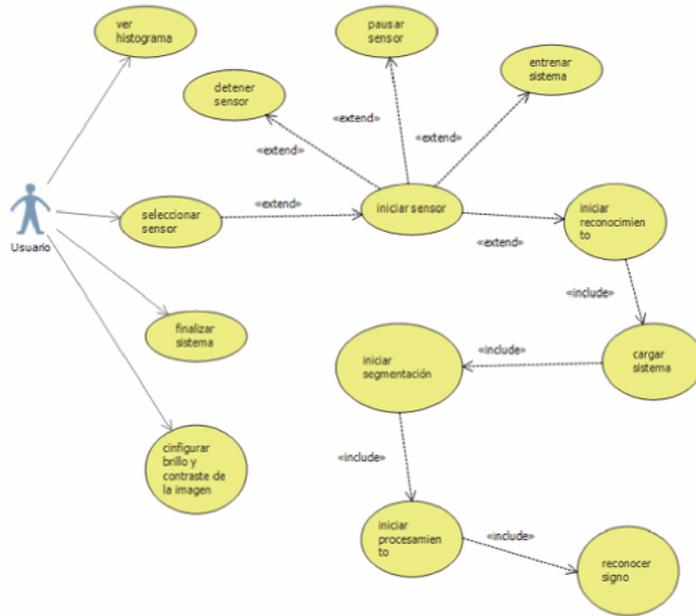


Figura 5. Diagrama de casos de uso.

III. RESULTADOS

Dado que el producto busca ser una plataforma para posteriores desarrollos, se puede usar en dos modalidades: como aplicación convencional en su modo de interpretador o como entorno de procesamiento de imágenes gestuales. En este último caso, la herramienta no sólo entrega el resultado textual sino un conjunto variado de informaciones útiles para el investigador. La Figura 6 muestra la pantalla de interacción con el usuario. En ella se identifican siete regiones: 1. Imagen original, 2. Imagen procesada, 3. Salida, 4. Herramientas de procesamiento, 5. Parametrización, 6. Ajustes y 7. Accesos.

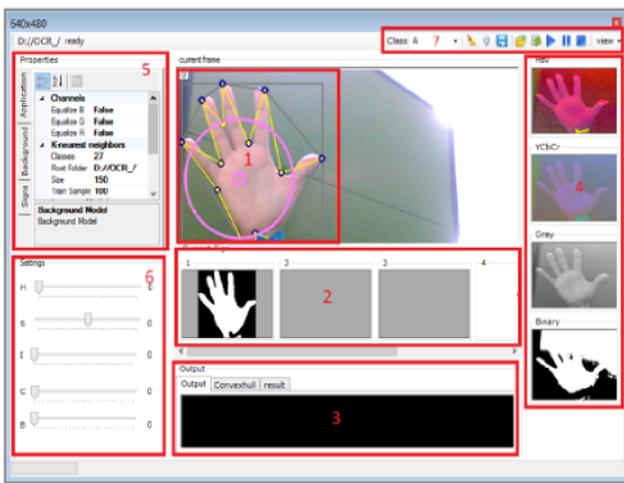


Figura 6. Interfaz del sistema.

Al momento de realizar la captura, el sistema evalúa la relación entre el objeto de interés y el entorno, con miras a determinar si el contraste entre ambos es ade-

cuado para que las siguientes etapas produzcan resultados apropiados. La emisión del juicio se da en una de tres clasificaciones (ver Figura 7).

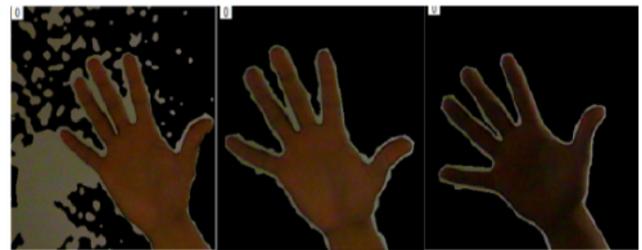


Figura 7. Relación objeto/fondo:
Izq: Deficiente, Centro: Aceptable, Der: Ideal

Las pruebas base se hacen con el signo estándar para la letra A (el puño cerrado). La separación del objeto de interés se realiza utilizando una combinación de segmentadores por modelos de color, en particular los modelos Hsv y YCbCr (ver Figura 8).

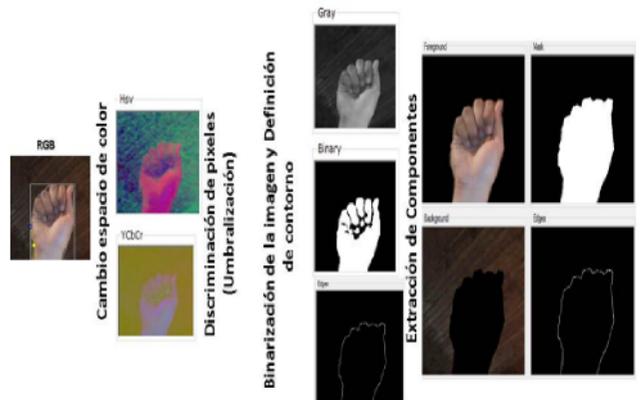


Figura 8. Segmentación basada en color.

La evaluación del contorno se hace con el algoritmo *convexhull*. La Figura 9 ilustra el caso para la letra A.

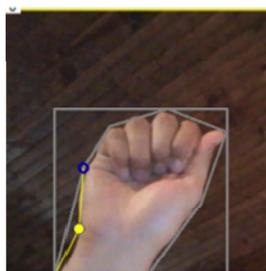


Figura 9. Evaluación de contornos mediante convexhull.

El seguimiento del gesto, para el caso de gestos dinámicos, a través de sucesivas imágenes, se hace mediante la extracción de componentes por máquinas de aprendizaje (Figura 10).

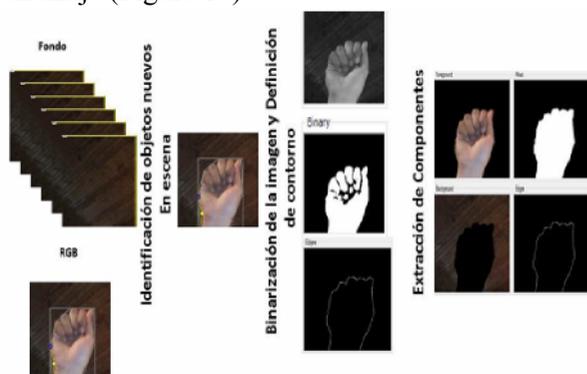


Figura 10. Seguimiento de gestos dinámicos.

Los gestos en el lenguaje de signos para las 26 letras se ilustran en la Figura 11. En la misma, *L indica el signo visto desde la izquierda y *R visto desde la derecha. Observe como algunas letras (J y Z) emplean un gesto dinámico en vez de estático como puede ser el caso para las otras letras.

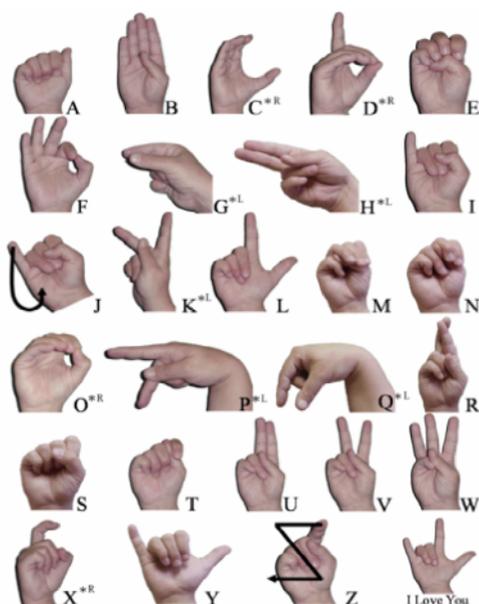


Figura 11. Gestos para las letras.

Para las pruebas de reconocimiento se emplearon los gestos para las primeras seis letras del alfabeto. En un primer conjunto de pruebas se emplearon 50 imágenes de entrenamiento, pero se observó cómo se generaba ambigüedad en la identificación de varios de los signos. Se incrementó la cantidad de imágenes en pasos de diez y se observó cómo a partir de 100 imágenes la identificación ya no presenta ambigüedades. En todas las pruebas se usaron seis segundos como parámetro para el tiempo de separación entre un gesto y el siguiente. Los resultados para las primeras seis letras del alfabeto se consignan en la Tabla 1.

Tabla 1. Prueba de entrenamiento/reconocimiento.

Signo/Clase	Imágenes por signo	
	50	100
A	A, E	A
B	B, R	B
C	C	C
D	D, V	D
E	A, E	E
F	F	F

Finalmente, el sistema se prueba para determinar su competencia en la detección de secuencias de gestos, respetando la normativa de una ventana de seis segundos entre un gesto y el siguiente. La detección es totalmente exitosa (Figura 12).



Figura 12. Seguimiento de secuencias de gestos.

DISCUSIÓN

- Es posible implementar un detector de gestos que trabaje en tiempo real.

- Utilizando una combinación apropiada de código propio con bibliotecas de soporte, la implementación puede realizarse con recursos estándares, tanto para la cámara como para el computador.
 - El algoritmo *convexhull* puede emplearse para detectar exitosamente el contorno de la mano y servir así como insumo para la detección del gesto asociado.
 - El algoritmo *k-nearest* puede emplearse, en combinación con un conjunto apropiado de entrenamiento, para clasificar apropiadamente un gesto.
 - Soluciones como la planteada ayudan no sólo a la comunidad de usuarios con discapacidad, como una herramienta que mejore sus formas de comunicación, sino también para la comunidad de investigadores, como una herramienta para la exploración de técnicas y métodos para interpretación de gestos.
-

BIBLIOGRAFÍA

1. González R, Woods, R. Digital Image Processing.; 2009.
2. Rodriguez, R. Procesamiento y análisis digital de imágenes. Mexico; 2012.
3. Balsero, N.O., Botero, D.A., Zuluaga, J.P. Reconocimiento e Interpretación de Gestos Manuales por Medio de Video (Tesis). Bogotá; 2005.
4. Chan TM. A Dynamic Data Structure for 3-D Convex Hulls and 2-D nearest neighbor queries. J. ACM 01/2010. 2010; p. 57.
5. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician 46 (3). 1992; p. 175–185.
6. Neidle, C. The Syntax of American Sign Language: Functional Categories and Hierarchical Structures Cambridge, MA: The MIT Press; 2000.
7. Rozo, N. La lengua de señas colombiana. Recuperado el 19/06/2014, de www.lenguasdecolombia.gov.co/content/lengua-de-señas-colombiana.
8. Trigo, T, Pellegrino, S.R.M. An Analysis of Features for Hand-Gesture Classification. En 17th International Conference on Systems, Signals and Image Processing (IWSSIP 2010), Brasil: Electronic Engineering and Computer Department, 2010. pp 412-415.