

Identificación de lengua de señas mediante técnicas de procesamiento de imágenes

Manuel Martínez Gutiérrez¹, José Rafael Rojano-Cáceres²,
Ismael Everardo Bárcenas Patiño¹, Fredy Juárez Pérez³

¹ Universidad Veracruzana, Maestría en Sistemas Interactivos Centrados en el Usuario,
Xalapa, México

² Universidad Veracruzana, Facultad de Estadística e Informática,
Xalapa, México

³ Instituto Tecnológico Superior de Álamo Temapache,
Facultad de Ciencias de la Computación,
Tuxpan, México

cheleazar1805@gmail.com, rrojano@uv.mx, iebarcenaspa@conacyt.mx,
fredyjuarezp@gmail.com

Resumen. En este trabajo se aborda el procesamiento de imágenes obtenidas a partir de una cámara digital con el objeto de caracterizarlas empleando técnicas básicas de visión computacional. Se puede definir como técnicas básicas a las transformaciones en el dominio de la frecuencia. La aplicación de diversas técnicas tiene como objeto la identificación de señas que provienen de la Lengua de Señas Mexicana, en particular de señas del tipo estático como es el caso de un subconjunto del alfabeto. Como muestra del enfoque planteado se presenta el desarrollo de un prototipo que sirva de apoyo para que una persona que realiza una seña pueda saber si está correctamente realizada.

Palabras clave: Lengua de señas, procesamiento digital, inclusión social.

Identification of Sign Language Using Image Processing Techniques

Abstract. This work deals with the processing of images obtained from a digital camera in order to characterize them using basic computer vision techniques. The transformations in the frequency domain can be defined as basic techniques. The application of different techniques has as an objective the identification of signs that come from the Mexican Sign Language, in particular signs of the static type as one subset of the alphabet. As a sample of the approach, the development of a prototype is presented in order to help a person in the correct realization of signs.

Keywords: Sign language, digital processing, social inclusion.

1. Introducción

En la vida cotidiana, las computadoras manejan una gran cantidad de datos provenientes de diferentes fuentes y formatos como pueden ser sensores, bases de datos, redes sociales, textos, etc. Aunado a este proceso, nos encontramos ante la necesidad que tienen las personas de utilizar diferentes dispositivos de comunicación que enriquezcan y faciliten la Interacción Humano Computadora (IHC). Como resultado se tiene la necesidad desarrollar técnicas computacionales que permitan la búsqueda de patrones o datos característicos en imágenes, ondas de audio o pulsos eléctricos, entre otros, para llevar a cabo tareas que solo hasta el momento puede realizar mejor el ser humano [1].

De esta forma, con la intención de mejorar tanto la Experiencia de Usuario (EU) así como la facilidad de interacción con las computadoras se vienen proponiendo diversos enfoques de interacción natural [2]–[6] entre los cuales se considera tanto el tratamiento digital de imágenes junto con la adquisición desde diversas fuentes de datos como puede ser un sensor como Kinect.

En este trabajo se plantea una propuesta para la identificación de señas derivadas de la Lengua de Señas Mexicana empleando algoritmos básicos para el procesamiento de la imagen tal como lo son las transformaciones espaciales (transformación en escalas de grises, binarización, el uso de umbrales, etc.). El objetivo propuesto tras emplear diferentes técnicas para el procesamiento de imágenes es el de desarrollar un prototipo a través del cual una persona con o sin entrenamiento en Lengua de Señas (LS) pueda validar si lleva a cabo de forma correcta la seña. A este proceso de acuerdo con Cruz-Aldrete [7] se le denomina “señar” otros términos también son el de “signar”. Aunado a lo anterior es necesario plantear que la Lengua de Señas es específica para cada país, en el caso de México se le conoce como Lengua de Señas Mexicana. Por otro lado la LS se define como aquel lenguaje viso-espacial en donde se genera comunicación empleando las manos y sus posiciones en relación con el cuerpo y la mirada intencional [8].

Así la estructura de este trabajo considera una introducción, descripción sobre las técnicas de procesamiento de imágenes empleadas, el diseño de un prototipo, identificación de la seña, conclusiones y referencias.

2. Técnicas de procesamiento de imágenes

En esta sección se presentan y explican brevemente las diferentes técnicas de procesamiento de imágenes empleadas para el desarrollo del prototipo que reconoce un conjunto de Lengua de Señas estáticas. A continuación se explica cada una de ellas.

2.1. Escala de grises

La primera técnica a explicar consiste en la aplicación de un filtrado de transformación en escala de grises. Para transformar una imagen a escala de grises mientras que esta proviene del modelo RGB, se puede realizar de diversas maneras, en este caso se realiza un promedio de los valores RGB de entrada y se realiza una asignación de dicho valor promediado igual en cada uno de los píxeles de la imagen.

La función que representa esta transformación se muestra en (1). Donde los valores RGB transformados pueden ser ya sea tomando el promedio de los valores R, o G, B. En este caso la transformación en escala de grises sirve como etapa previa de la binarización, ya que al estar todos los píxeles en dicha escala facilita la siguiente etapa del proceso:

$$(r', g', b') = \begin{cases} (r, r, r), & o \\ (g, g, g), & o \\ (b, b, b). \end{cases} \quad (1)$$

2.2. Negativo

Otro filtro básico es el de la transformación a valores negativos. El filtro negativo consiste en realizar una resta del valor máximo de la imagen (255) menos el valor RGB del pixel, esto por cada pixel de la imagen. La transformación en negativo se define a partir de un rango de valores de intensidad comprendido entre $[0, L - 1]$ y mediante la aplicación de la ecuación (2) el resultado es la inversión de los valores dado el máximo del rango de entrada $L - 1$ menos el pixel de entrada r :

$$s = L - 1 - r. \quad (2)$$

2.3. Threshold

El filtro threshold o mejor conocido como filtro binario, permite transformar una imagen a binario o blanco y negro. Este filtro necesita que el usuario ingrese un valor mayor a 0 y menor a 255, con este valor le permite al filtro catalogar los colores para saber si cambiarlo por el color blanco o el color negro. Se utiliza en este caso para remarcar la mano con el color negro y el fondo quede de color blanco:

$$s = \begin{cases} 0 & \text{si } r \leq c, \\ L - 1 & \text{en otro caso.} \end{cases} \quad (3)$$

2.4. Recorte binario

Se definió una función denominada recorte binario cuyo propósito es la extracción de la mano, misma que se encuentra en color negro. De esta forma el recorte binario recorre la imagen buscando los puntos en color negro estableciendo los puntos mínimos y máximos que conforman la imagen para descartar el resto del fondo.

2.5. Zoom

Uno de los filtros más importantes de la aplicación consiste en la aplicación de un aumento digital, así como su correspondiente decremento digital. El aumento digital consiste en incrementar el tamaño de la imagen dependiendo de un valor que introduce el usuario, a partir de ahí se debe realizar una ecuación para separar los píxeles y rellenarlos con valores de algún vecino. El decremento digital es parecido al aumento,

pero en su función inversa. Se utiliza para proporcionar una altura a la imagen que el usuario ingresa, en este caso se maneja el tamaño de la imagen a 150 píxeles.

2.6. Rotación

La función para rotación se emplea como un mecanismo de prueba de tolerancia al identificar una imagen. En este caso se consideró una tolerancia de ± 5 grados. La función de rotación es tal y como se describe en la ecuación (4) en donde R se define como en (5):

$$x' = Rx + t, \quad (4)$$

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (5)$$

2.7. Closing y opening

El filtro opening permite agrandar espacios pequeños en las imágenes, esto lo logra aplicando el filtro erosión y luego dilatación. El filtro closing a diferencia del opening se centra en cerrar los pequeños agujeros que se pueden considerar como ruido en una imagen, esto lo logra aplicando el filtro de dilatar y luego el de erosionar. Así la aplicación de estos filtros permite eliminar el ruido de la imagen, así como el ruido que se genera al aplicar diversos filtros para el tratamiento.

3. Prototipo desarrollado

El desarrollo del prototipo se realizó en el lenguaje Java, con una interfaz gráfica que permitiera ver la funcionalidad del sistema y que mediante su empleo el usuario pudiera conocer si al realizar una determinada seña se realizaba de forma correcta, veáse Fig. 1.

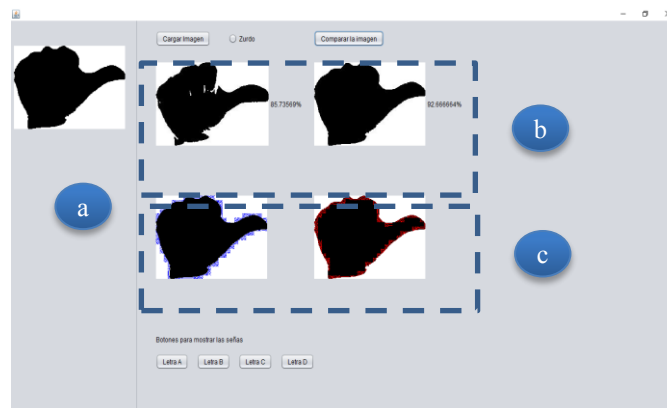


Fig. 1. Pantalla del prototipo. a) imagen de usuario, b) imágenes identificadas como correctas, c) errores de comparación.

Para esta versión del prototipo las imágenes a verificar pueden ser tomadas a partir de una cámara del celular o de una cámara web y posteriormente se salvan a un archivo jpg o png que deberá ser introducido al sistema para su validación. La aplicación consta únicamente de una sola ventana que se encuentra dividida en cuatro funcionalidades descritas en detalle en la siguiente sección.

3.1. Casos de uso

Las funcionalidades que puede realizar el usuario en el sistema, se limitan a cuatro actividades que se denotan en el diagrama de caso de usos mostrado en la Fig. 2. El usuario puede realizar tres de esas actividades sin ningún requisito, pero para comparar la imagen es necesario que primero cargue una imagen. A continuación se explica brevemente cada caso de uso.

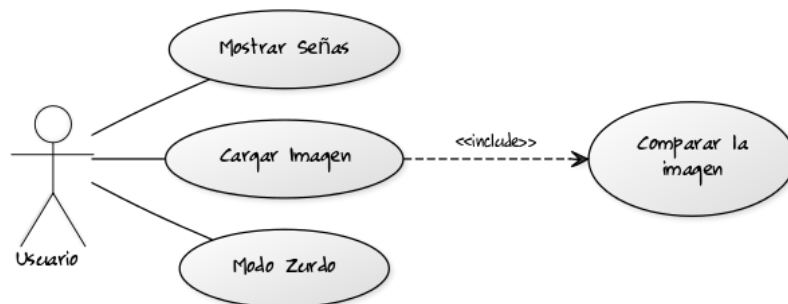


Fig. 2. Diagrama de casos de uso para el prototipo desarrollado.

Mostrar señas. El propósito de esta función es mostrar las letras que corresponden a una determinada seña. En este caso para prueba el conjunto de señas es reducido y se limita a cuatro letras del alfabeto dactilológico. Así cuando el usuario desea conocer cómo se seña la letra “a” entonces el sistema mostrará la letra “a” en el área b de Fig. 1. Así mismo, dentro del área b, se muestran resultados asociados al porcentaje de reconocimiento de la seña.

Cargar imagen. La funcionalidad de cargar imagen, como su nombre lo indica, permite al usuario subir al sistema una imagen de su mano realizando una seña. Para su correcto funcionamiento, se espera que la mano se vea claramente y el fondo sea oscuro. Al realizar el proceso de carga también se realizará la llamada a los diferentes pasos del procesamiento de imágenes descritos en la sección anterior. En concreto al realizar la carga se efectúa la binarización de la imagen: escala de grises, negativo y threshold. Una vez que tiene la imagen en binario se aplican los filtros morfológicos opening y closing para disminuir el ruido de la imagen. Al resultado del paso anterior se le aplica una función que recorta la imagen para obtener la imagen con la mano únicamente. El último paso en esta función es redimensionar la imagen a un tamaño esperado por la aplicación, en este caso es una altura de 150cm. para esto se utiliza el filtro zoom in y zoom out.

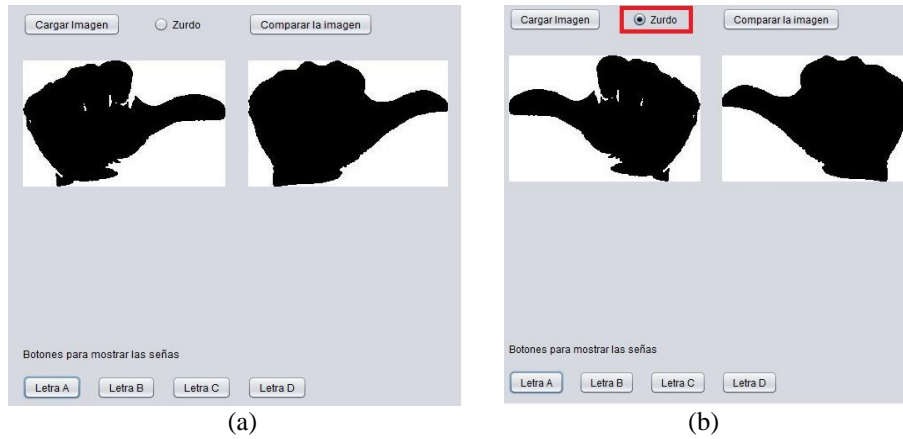


Fig. 3. Ventana de la aplicación (a) mostrando la letra A, (b) mostrando la letra A en zurdo.

Modo zurdo. El modo zurdo como su nombre lo indica es una función que al ser activada aplica una reflexión horizontal a las imágenes. La activación del modo zurdo tiene como propósito reconocer aquellas imágenes que hayan sido grabadas por personas zurdas.

Comparar la imagen. Finalmente, la funcionalidad de comparar la imagen realiza un cálculo que consiste en determinar el porcentaje de similitud entre un conjunto de señas previamente registradas en el sistema con relación a la imagen que carga el usuario.

En este caso al realizar la comparación de las imágenes se mostrará en la aplicación un contorno en tono rojo para las zonas que debería ocupar la mano y azul para las zonas que no debería ocupar la mano. Véase Fig. 4.

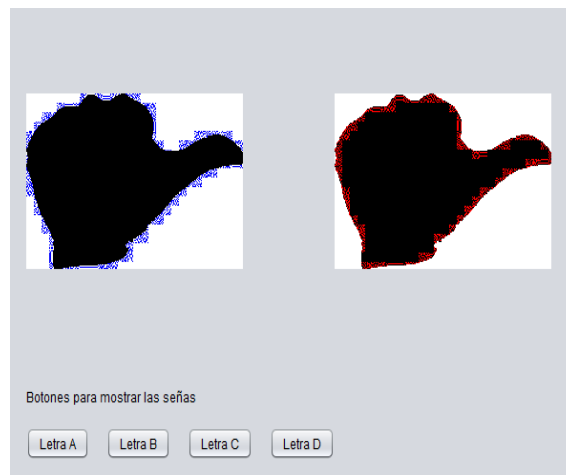


Fig. 4. Área de la aplicación que muestra los resultados de la comparación de la imagen.

4. Identificación de la seña

El proceso de identificación de la seña se basa en la comparación de dos imágenes, la primera se adquiere a través de la cámara por parte del usuario y la segunda se encuentra previamente almacenada en el sistema. A partir de las dos imágenes, el sistema recorre pixel por pixel comparando el valor de tonalidad de estos. En el caso de que coinciden los valores binarizados el sistema aumentará un contador, en caso contrario el sistema identificará cual era el resultado esperado tomando como base la imagen propia del sistema. Así, dependiendo del valor esperado el sistema guarda un registro del exceso o la falta de la coincidencia. De forma que se indicará con un color rojo aquellos pixeles que representan lo que debería haber ocupado o bien en azul representa lo que no debe ocupar, es decir ambos son un tipo de error natural pues es muy complejo que la mano siempre ocupe la misma posición aun cuando lo haga la misma persona. Una vez que el sistema termina de recorrer los pixeles, realiza la media de aciertos entre el número de pixeles recorridos para determinar el porcentaje de precisión. El proceso anterior se debe repetir para comparar la imagen del usuario con todas las imágenes que tiene el sistema y a partir de la media más alta se deduce la seña que realizó el usuario. De esta forma, la imagen de entrada se compara contra todas las imágenes que se encuentran en el sistema dado que no existe conocimiento de cuál podría ser la letra a identificar. Así mismo en el sistema existen variaciones de la misma letra considerando aspectos como el de rotación o bien la flexión de los dedos que como se expresó son naturalmente cambios de la postura al grabar la seña. De esta forma el algoritmo comparará contra todas las variaciones con el propósito de tolerar cierta imprecisión derivada de la generación de la seña por parte del usuario.

```
Función compara (imagen A, imagen B)
  doble porcentaje=0
  entero contador=0
  entero numeroPixel=0
  Desde entero I=0, hasta I == ancho(A)
    Desde entero J=0, hasta J == alto(A)
      entero tonalidadA = tonalidad(I,J) de A
      entero tonalidadB = tonalidad(I,J) de B
      Si tonalidadA == tonalidadB
        Contador = contador + 1
        numeroPixel= numeroPixel + 1
    Fin desde
  Fin desde
  Porcentaje = contador/numeroPixel
  Porcentaje = porcentaje*100
  regresa porcentaje
Fin función
```

Fig. 5. Pseudo-código de la función compara imagen.

5. Pruebas

Para corroborar que el algoritmo planteado en la Fig. 5 era funcional para la identificación de la seña se procedió a realizar dos tipos de pruebas, la primera toma como parámetros la imagen de la cual ya se sabe que letra es, la segunda prueba hace una evaluación considerando el resto de letras.

En la primera prueba, que funcionó como experimento de control, se ingresó al sistema la imagen previamente identificada esperando que al compararla con su correspondiente se tuviese una precisión del 100%. Sin embargo, se encontró que el algoritmo solo daba precisiones superiores al 90% sin llegar al cien por cien. Tras un análisis se observó que la función implementada en Java generaba un pequeño ruido, pero este se trató mediante la aplicación de un filtro de threshold con lo cual dichos errores desaparecieron produciendo una certidumbre del 100%.

Para el segundo tipo de prueba, se usaron diferentes imágenes considerando una ejecución normal del sistema. Es decir, se capturaron nuevas señas y se procedió a ingresarlas al sistema para su identificación contra las que tenía registradas. Para la verificación se capturaron diversas imágenes aplicando rotación, misma que fue considerable al ser de aproximadamente de entre 10 y 40 grados. De esta forma se ingresaron grupos de cinco imágenes por cada letra, mismas que fueron realizadas por dos personas, con lo cual nos daba como punto de comparación y reconocimiento un total de veinte imágenes de prueba.

Como resultado de la identificación de las señas se encontró que el sistema reconoció correctamente dieciséis de las veinte imágenes. Entre las imágenes que no identificó se encuentran la letra a y la letra d que correspondían a la grabación de la seña realizada por una persona no registrada en el sistema, mientras que para la misma persona aun cuando se aplicaron las rotaciones y las variaciones naturales generó una precisión del 100%. Cabe hacer algunas precisiones, la variación de la rotación se debe el valor mayor a que la persona que ingresó la imagen tenía escasos conocimientos sobre Lengua de Señas por lo cual la emisión de la seña tenía un gran error, mientras que la persona con la cual tuvo mejor precisión era un usuario que conoce la LS y por tanto producía menores variaciones al momento de ingresar la seña.

6. Conclusión

En esta investigación se abordó el procesamiento de imágenes mediante la aplicación de técnicas esenciales como las descritas en la sección 2, así mismo se implementó un procedimiento que permitiera identificar la seña realizada teniendo como fuente la imagen procesada. En este trabajo se optó por emplear técnicas derivadas de la visión por computadora para demostrar su potencial en el reconocimiento de señas provenientes de la Lengua de Señas Mexicana, en particular de señas estáticas, es decir aquellas que no requieren movimiento continuo para su ejecución. De acuerdo a las pruebas realizadas se encontró que el porcentaje de reconocimiento fue superior al 75%. De esta forma se considera que el prototipo cumple 1) con la función de procesar imágenes para su tratamiento digital, y 2) identificar las imágenes mediante un algoritmo sencillo como el descrito en la Fig. 5, al final del artículo.

Finalmente, como trabajo futuro se considera ampliar la base de imágenes empleadas para prueba, es decir en primera instancia proveer todo el conjunto de letras del abecedario y posteriormente realizar una prueba con diversas señas estáticas de la lengua de señas. De esta forma se provee un medio para que personas puedan por un lado practicar el uso de la Lengua de Señas, y por otro lado se fomenta la inclusión de personas con discapacidad auditiva al generar una actitud sensibilizadora en las personas oyentes a través del aprendizaje de dicha lengua.

Referencias

1. Rich, E., Knight, K.: Artificial intelligence. McGraw-Hill, New (1991)
2. Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P.: American sign language recognition with the kinect. In: Proceedings of the 13th international conference on multimodal interfaces, pp. 279–286 (2011)
3. Liang, R. H., Ouhyoung, M.: A real-time continuous gesture recognition system for sign language. In: Automatic Face and Gesture Recognition, Third IEEE International Conference on, pp. 558–567 (1998)
4. Sturman, D. J., Zeltzer, D.: A survey of glove-based input. IEEE Comput. Graph. Appl., Vol. 14, No. 1, pp. 30–39 (1994)
5. Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. IEEE Trans. Pattern Anal. Mach. Intell., Vol. 20, No. 12, pp. 1371–1375 (1998)
6. Gavrilu, D. M.: The visual analysis of human movement: A survey. Comput. Vis. image Underst., Vol. 73, No. 1, pp. 82–98 (1999)
7. Cruz, M.: Gramática de la lengua de señas mexicana. Universidad del Valle, Colombia (2008)
8. D. O. de la Federación: Ley General para la Inclusión de las Personas con Discapacidad. D. Of. la Fed. (2011)