Reconocimiento de Gestos basado en Modelos Ocultos de Markov utilizando el Kinect

Adrián Pastor López Monroy y Jesús Adrián Leal Meléndrez

Instituto Nacional de Astrofísica, Óptica y Electrónica
Coordinación de Ciencias Computacionales
Luis E. Erro 1 Tonantzintla,
Puebla, México
{pastor,jalm}@ccc.inaoep.mx

Resumen En este documento presentamos una alternativa para representar y aprender gestos utilizando Modelos Ocultos de Markov HMM y el Kinect. Nos enfocamos en sacar la máxima ventaja del kinect para evitar que la extracción de datos sea afectada por factores como la iluminación. Implementamos algoritmos de seguimiento de objetos y guardamos los gestos (ordenes) de el usuario para ser aprendidos. Procesamos los datos de los movimientos y los aproximamos con HMM. Estos modelos son evaluados en un esquema de clasificación con 7 gestos diferentes. Hemos encontrado que la representación que utilizamos y los HMM en conjunto con tecnologías nuevas como el kinect dan resultados interesantes con apenas la aplicación fundamental de la teoría de los Modelos Gráficos Probabilísticos.

1. Introducción

La extracción de datos mediante una cámara convencional, puede ser utilizada para el aprendizaje de gestos. Muchos experimentos en esta dirección han demostrado resultados satisfactorios, la mayoría de éstos aplicando Modelos Gráficos Probabilísticos (MGP); tales como Modelos Ocultos de Markov (HMM) y Clasificadores Bayesianos Dinámicos Simples (DNBCs). Sin embargo nuevas tecnologías en el área de visión han emergido, facilitando y haciendo más exacta la extracción y manipulación de datos. Una de estas tecnologías es el dispositivo Kinect de Microsoft. Éste es un dispositivo para captura de datos especializado en seguimiento y reconocimiento de gestos; pensado inicialmente para una interacción libre de contacto físico entre el jugador y la consola de videojuegos Xbox 360 también de Microsoft.

Desde el lanzamiento al mercado del Kinect, se ha estado hablando de sus implicaciones más allá del negocio de los videojuegos, aprovechando las principales tecnologías que ofrece: sensor de profundidad, cámara RGB y micrófono, todo ésto integrado en un mismo dispositivo de forma óptima para su uso en conjunto.

En este trabajo presentamos una combinación Kinect-HMMs para el aprendizaje de gestos dinámicos. En secciones posteriores explicaremos con más detalle

los HMMs. Por ahora es suficiente mencionar que son un modelo de Markov en el que los estados no son directamente observables y los utilizamos para obtener representaciones robustas de cada gesto. El interes particular por utilizar esta representación basada en HMMs es debido a la amplia literatura al respecto y los buenos resultados que se han conseguido con ellos.

Esta combinación Kinect-HMMs no ha sido utilizada ampliamente debido a la recién aparición del Kinect. Ésto nos da un incentivo extra para la investigación, además del potencial que ofrece la nueva tecnología aunado con los excelentes resultados que ha arrojado el uso de HMMs.

La idea principal de nuestro trabajo es utilizar el Kinect para extraer características de movimiento de la mano (coordenadas de la posición de la mano) y usar los puntos $p(x, y, z)$ para construir representaciones de HMMs que sean eficaces en aprendizaje automático. Con ésto tendremos un modelo que representa un gesto particular hecho por el usuario. Este procedimiento se realizará para modelar cada uno de los ademanes que nos interesa aprender. Después entrenaremos nuestro modelo para que ajuste los parámetros y nos devuelva uno más exacto. Posteriormente podremos evaluar un nuevo gesto midiendo la semejanza con cada uno nuestros modelos.

El resto del documento está organizado como sigue. En la sección 2 presentamos algunos trabajos relacionados con el reconocimiento de gestos y el uso de MGP. En la sección 3 presentamos la metodología y el desarrollo de nuestro trabajo. En la sección 4 mostramos algunos experimentos realizados y los resultados obtenidos. Finalmente en la sección 5 mostramos las conclusiones y trabajo futuro de nuestra investigación.

2. Trabajo Relacionado

En trabajos recientes se aborda el reconocimiento de gestos desde diferentes enfoques. A continuación presentamos una breve recapitulación de algunos trabajos sobre esta misma línea de investigación.

En [5] se aborda el tema de reconocimiento de gestos aplicando HMM y DNBCs. Los DNBCs extienden a los HMM incorporando suposiciones de independencia entre los atributos dado el estado del modelo. Esta combinación ofrece porcentajes de clasificación y dispersión de error competitivos, un menor número de parámetros para el modelo y una mejora considerable del tiempo de entrenamiento. Además, para describir los gestos se propone un conjunto de atributos simples de postura y movimiento que incrementan el porcentaje de reconocimiento en comparación a modelos que sólo utilizan información de movimiento.

Los HMMs describen propiedades estadísticas de gestos dinámicos, a través de los algoritmos de estimación de probabilidad bien estudiados para aprendizaje y reconocimiento [1].

Los Modelos Ocultos de Markov Parametrizados (PHMMs) representan gestos que involucran variaciones espaciales en su ejecución i.e, This length ó Go there En PHMMs las variables de observación están condicionadas a la variable estado

y uno o más parámetros que cuentan para tales variaciones. Los valores de los parámetros son conocidos y constantes durante el entrenamiento [4].

3. Metodología y Desarrollo

3.1. Configuración del Kinect en Linux y Requerimientos preliminares

El primer paso hacia el reconocimiento de gestos dinámicos es la correcta configuración del kinect. Para ello hemos considerado una instalación sobre un sistema operativo basado en Linux con kernel 2.6.35-28. En específico utilizamos Ubuntu 10.10 de 32 bits. Es importante escoger bien el sistema operativo debido a la compatibilidad de éste y las otras librerías que serán necesarias. Un análisis superficial del sistema operativo y su compatibilidad con otras herramientas puede conllevar a un bajo rendimiento tanto del software como del hardware. Por ejemplo, el sistema operativo Windows en general no es buena opción para este proyecto en particular ya que es considerablemente más lento con todo el conjunto de librerías que utilizaremos, lo que provoca un desempeño pobre de los algoritmos de seguimiento de los que hacemos uso.
Algunas de las herramientas adicionales en las que nos hemos apoyado para la manipulación e implementación de algoritmos son:

- OpenCV: Para la visualización de la cámara RGB del Kinect.
- OpenGL: Utilizado para la creación de ventanas y gráficos de prueba.
- OpenNi: Librería con implementación de algoritmos de segmentación de persona con profundidad.
- NitePrimeSense: Controladores para el kinect e implementación de algoritmos de seguimiento.
- jahmm para el modelado y manipulación de los HMMs.

3.2. Establecimiento del conjunto de gestos

Al igual que en muchas otras tareas del Aprendizaje Automático, la siguiente etapa consiste en la recolección del conjunto de datos. Éste lo utilizamos tanto para construir el modelo de cada gesto, así como también para la fase de pruebas. Para nuestro trabajo consideramos el aprendizaje de 7 gestos básicos. Los gestos son realizados a 2 metros de distancia del kinect y en posición frontal a través de movimientos específicos del brazo y la mano. Los gestos que hemos tomado en cuenta para el presente trabajo son:

- Izquierda (A1).
- Derecha (A2).
- Abajo (A3).
- Arriba (A4).
- Ven (A5).
- Alto (A6).

- Saludo (A7).

La figura 2 muestra la forma en que son ejecutados cada uno de estos gestos por la persona que esta dando la orden, el gesto saludo es solo para introducir variedad en las distintas formas de los gestos y probar el poder de los HMMs para este tipo de curva oscilante.

Figura 1: Gestos ejecutados por el usuario.

3.3. Extracción de características

Para la extracción de los datos de cada gesto, le hemos implementado al kinect un algoritmo de seguimiento enfocado en la palma de la mano. El algoritmo se programó utilizando algunas funciones básicas de la librería OpenNi. La idea principal de este algoritmo es seguir continuamente la mano y al mismo almacenar en una estructura de datos los puntos $p(x, y, z)$ de su posición. Estas coordenadas estarán en el rango dentro de la resolución proporcionada por los sensores del Kinect (640x480 pixeles) y se almacenan en promedio 30 puntos por segundo. Para la extracción de estas características se asume que la persona siempre está de frente y a una distancia constante del kinect (2 metros).

Al extraer la información de los gestos explotamos la capacidad del sensor de profundidad. Así pues los datos que se extraen no se ven afectados por el nivel de iluminación. La razón de ello es que en los algoritmos de seguimiento solamente se está haciendo uso del sensor de profundidad y no de la cámara RGB.

3.4. Representación de los gestos

Para la duración de captura de datos de cada gesto consideramos en promedio una duración de 2 segundos por gesto. De esta forma, en promedio se obtienen 60 puntos por movimiento. Sin embargo, realmente hay una acumulación de puntos al iniciar cada gesto y también al terminarlo (no se identifica bien el inicio y el fin del gesto). Además, algunos gestos duraron menos que otros y por lo tanto se tienen menos puntos con información del movimiento. Es por esto que hemos procesado estos datos de la siguiente manera: obtenemos cada secuencia de puntos de los gestos y la dividimos para ir tomando solo cada cuatro puntos. De esta manera en las observaciones consideraremos los cambios de información cada cuatro puntos. De esta manera, si el kinect guarda aproximadamente 30 puntos por segundo y cada gesto dura 2 segundos, entonces tendremos aproximadamente unas 20 observaciones por gesto.

3.5. Implementación del método de aprendizaje basado en HMMs

Para la representación del movimiento de la mano decidimos utilizar HMMs. Una características importante que tienen los HMMs es que pueden modelar bastante bien los estados con respecto al tiempo. Nuestro principal objetivo es obtener representaciones robustas que saquen la mayor ventaja de los sensores con los que cuenta el kinet. Para esto proponemos extraer características de movimiento. Por lo tanto, aprovechamos la precisión para el seguimiento de objetos. Se implementó un algoritmo de la librería OpenNi que hace un seguimiento preciso de la palma de la mano del usuario. Posteriormente obtenemos puntos $p_i(x_i, y_i, z_i)$ que representan la posición de la mano y preprocesamos los datos para después ejecutar la clasificación de los objetos midiendo la exactitud de clasificación.

Atributos de los Gestos Como atributos de los Gestos hemos considerado las coordenadas de la palma de la mano del usuario. De esta forma, nuestro interés se enfoca principalmente en los cambios que se presenten cada 4 puntos observados. Las características son específicamente del movimiento de la mano. Así que, tomamos como atributos los incrementos $(\triangle x, \triangle y, \triangle z)$ entre los puntos $p_i(x_i, y_i, z_i)$ y $p_i(x_{i+4}, y_{i+4}, z_{i+4})$. Por medio del uso de estos tres valores en conjunto y la tecnología de kinect para seguimiento de objetos nos es posible estimar la posición del usuario en todo momento. También es importante mencionar que el valor que pueden tomar cada uno de estos atributos solo son $\{+, -, 0\}$. Donde el signo + representa que hubo un incremento positivo en los valores de los puntos con respecto a x (Similarmente se aplica a y, z). Así pues, tenemos 3 atributos donde cada uno puede tomar 3 valores diferentes. Esto nos deja con un total de 27 posibles observaciones.

Gestos y Clasificación Una buena selección general de las características para representar los gestos es algo que se ha perseguido durante bastante tiempo

[2] [3] [**?**]. Algunos de los gestos fueron modelados con distintos números de estados. Todo dependiendo de la complejidad del gesto. Por ejemplo para el gesto "alto"solamente se utilizo un estado, para los gestos "derecha", "izquierda", "ven", "arriba"y "abajo"se han utilizado 2 estados y por último para el gesto "saludo"se han utilizado 3 estados para representar el movimiento. El criterio principal para el escoger el número de estados entre cada gesto consistió de un análisis general y algunas pruebas empíricas.

Para nuestros experimentos hemos construido una estructura de clasificador bastante simple apoyándonos en la librería jahmm. Éste clasificador guarda una referencia a cada uno de los HMMs de los gestos. Cada uno de los modelos de gestos son entrenados individualmente con 100 iteraciones del algoritmo de Baum-Welch. Además hemos adicionado a nuestro clasificador con un par de funciones que facilitan el análisis de los resultados. El primero es la fácil obtención de una matriz de confusión y el segundo es una función que devuelve una estructura que nos proporciona la probabilidad de que la observación pertenezca a cada modelo. Posteriormente, el procesos de clasificación se desarrolla haciendo uso de esta función. Luego de la evaluación solo queda escoger aquel que presente una mayor probabilidad de presentarse en cierto modelo y seleccionarlo.

4. Experimentos y Resultados

Primeramente extrajimos secuencias de puntos con el kinect que representan los gestos. Este proceso lo repetimos para para dos personas extrayendo para la primer persona 50 secuencias de observaciones para cada gesto aproximadamente (20 para entrenamiento y 30 para prueba), y 25 secuencias por gesto para otra persona diferente con la que se entrenó el modelo. Una vez obtenidos los datos realizamos un pre-procesamiento, para reducir nuestras observaciones y transformar las coordenadas *kinect* (x, y, z) a una representación más adecuada(utilizando solo $+, -, 0$). La idea de tener dos corpus de secuencias es probar que tan bien funcionan los modelos que construimos cuando las ordenes son de diferentes personas.

Arriba	Abajo	Derecha	Izquierda	Saludo	Ven	Alto	
32	0	1	0	0	0	0	Arriba
0	26	1	0	4	0	1	Abajo
0	0	21	0	11	0	0	Derecha
0	0	0	34	0	0	0	Izquierda
0	0	0	0	31	0	2	Saludo
4	0	0	0	0	34	1	Ven
0	0	0	2	3	0	34	Alto

Tabla 1: Experimento 1. Matriz de confusión experimento 50 instancias por gesto (20 para entrenamiento y 30 para prueba) con 87,6 % de exactitud

En este experimento se observa que cuando la misma persona que entrena es la que da la orden la exactitud en la clasificación es hasta de 87,6 %. Sin embargo, no hay que pasar por alto que el gesto Derecha y Saludo, son altamente confunsos entre sí para poderse aprender. Mientras que gestos como Alto y Ven son altamente distinguibles debido a que se explota la coordenada de profundidad z.

Arriba	Abajo	Derecha	Izquierda	Saludo	Ven	Alto	
26	0	0	0	0	0	0	Arriba
0	19	1	0	0	0	2	Abajo
0	2	5	0	10	0	4	Derecha
0	0	0	7	0	0	14	Izquierda
0	0	0	0	10	0	12	Saludo
6	0	0	0	0	8	9	Ven
0	0	0	1	3	0	19	Alto

Tabla 2: Experimento 2. Matriz de confusión experimento 25 instancias de prueba con los modelos entrenados en el experimento anterior con 59,4 % de exactitud

En el segundo experimento el panorama cambia debido a que la persona que dio los gestos es considerablemente más alta y delgada. Sin embargo, el clasificador aun es capaz de distinguir la mayoría de las ordenes dadas. Creemos que esto se debe sobre todo al factor de profundidad del que hacemos uso en todos los gestos y que nos proporciona una noción más clara de la posición y seguimiento de la mano.

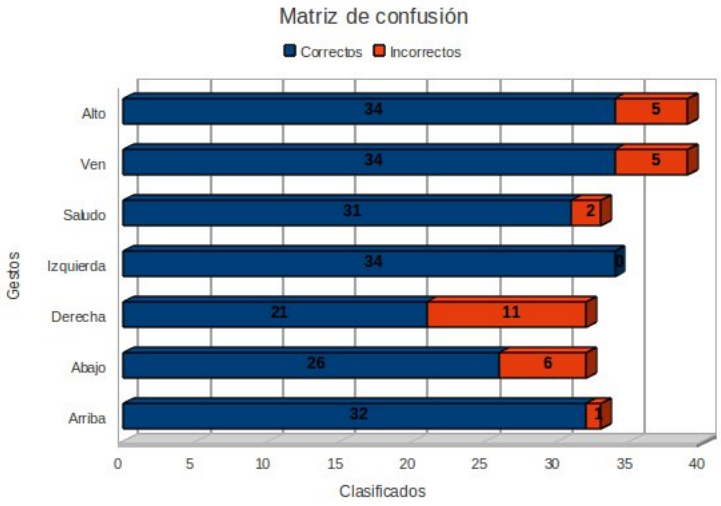

Figura 2: Barras de confusión experimento No 1.

5. Conclusiones y trabajo futuro

Hemos encontrado que la representación con HMMs para el aprendizaje de gestos puede dar resultados satisfactorios si los conceptos matemáticos se aplican correctamente. Aúnque aún falta probar movimientos a diferente distancia y rotación, queda claro que al menos con las pruebas realizadas hasta el momento se comprueba que los HMMs en conjunto con el kinect representan una alternativa factible y sencilla para aprender gestos. Ésta resulta ser además muy rápida una vez que ya se tiene el modelo de cada gesto.

Como trabajo futuro proponemos hacer un sistema de clasificación automatizado. Actualmente tenemos todo el proyecto separado en distintos programas con funcionalidades específicas. Creemos que sería útil unir todo este proyecto para facilitar el mantenimiento y acelerar el desarrollo de nuevas pruebas. Además proponemos procesar los datos de manera más fina para obtener más puntos significativos que puedan ayudar a obtener mejores HMMs. También queda como trabajo futuro las pruebas del clasificador con instancias de movimientos capturadas por un número mayor de personas diferentes.

Referencias

1. Alex Waibel, K.F.L.E.M.K.: A tutorial on hidden Markov models and selected applications in speech recognition pp. 267–296 (1990)
2. D., R.: Specifying Gesture by Example. Computer Graphics 25(4), 329–337 (July 1991)
3. L.E., M.J..S.: Feature Selection for Visual Gesture Recognition Using Hidden Markov Models. Proc. Fifth Mexican International Conference in Computer Science pp. 1–8 (2004)
4. Mardia K.V., G.N.H.T.H.M..S.N.: Techniques for online gesture recognition on workstations. Image and Vision Computing 15(1), 283–294 (1993)
5. Pedrycz, W.: Using Hidden Markov Models to Model and Recognize Gesture Under Variation. International Journal on Pattern Recognition and Artificial Intelligence, Special Issue on Hidden Markov Models in Computer Vision 15(1), 123–160 (2000)
6. Sucar-Succar, L., Mendoza-Durán, C., Pineda-Cortés, L., H.H. Avilés-Arriaga: A Comparison of Dynamic Naive Bayesian Classifiers and Hidden Markov Models for Gesture Recognition