Evaluación de técnicas de Machine Learning para el reconocimiento de gestos corporales

Rodrigo Ibañez, Álvaro Soria, Alfredo Teyseyre, Marcelo Campo

ISISTAN Research Institute (CONICET-UNICEN), Campus Universitario, Paraje Arroyo Seco, Tandil, Buenos Aires, Argentina
{rodrigo.ibanez, alvaro.soria, alfredo.teyseyre, marcelo.campo}@isistan.unicen.edu.ar

Abstract. El progreso y la innovación tecnológica alcanzados en los últimos años, en particular en el área de entretenimientos y juegos, han promovido la creación de interfaces más naturales e intuitivas. Por ejemplo, dispositivos de interacción natural como Microsoft Kinect permiten explorar una nueva forma de comunicación hombre-máquina mucho más expresiva mediante el reconocimiento de gestos corporales. En este sentido, han surgido diferentes estrategias que permiten el reconocimiento de gestos utilizando técnicas de Machine Learning. Sin embargo, no se ha hecho un estudio comparativo del comportamiento de estas técnicas. Por lo tanto, este trabajo presenta una evaluación de 4 técnicas de Machine Learning con un dataset de 7 gestos diferentes y 80 muestras para cada uno de ellos. Se evaluó la precisión de las distintas técnicas obteniendo resultados cercanos al 100% de los gestos evaluados en algunas de ellas.

1 Introducción

Los sensores de profundidad se han vuelto cada vez más populares reduciendo no solo su costo sino también su tamaño. El más conocido es sin duda Microsoft Kinect [1]. Este dispositivo permite identificar personas y obtener en tiempo real la posición en el espacio 3D de 20 partes del cuerpo humano. Esta característica ha sido aprovechada por desarrolladores de aplicaciones de interfaz natural, ya que permite generar una representación 3D del esqueleto humano que imite los movimientos de cada persona e incluso interpretar sus movimientos.

En este sentido, con el objetivo de facilitar la interacción entre humano-computadora han surgido diferentes enfoques para el reconocimiento de gestos corporales. Inicialmente, aparecieron enfoques basados en reglas sobre las posiciones de las partes del cuerpo que permiten reconocer posturas estáticas o movimientos simples de alguna parte del cuerpo [2][3][4]. Por ejemplo, permite identificar si la mano derecha está arriba de la cabeza o si la mano izquierda se mueve hacia la derecha en un intervalo de tiempo. La desventaja de estos enfoques es la dificultad para definir reglas de gestos complejos y el esfuerzo requerido para probar el correcto funcionamiento de las mismas. A esto se le suma la poca flexibilidad al momento de reconocer gestos realizados por personas con diferentes destrezas y diferentes contexturas físicas.

Para abordar estos problemas, aparecieron posteriormente enfoques más robustos que permiten reconocer gestos mediante técnicas de Machine Learning [5][6]. Estas técnicas requieren un conjunto etiquetado de gestos de ejemplo para aprender y posteriormente

poder identificar un nuevo gesto como uno de los gestos aprendidos. Por ejemplo, Bhattacharya y otros utilizaron *Support Vector Machines* (SVM) y *Decision Trees* (DT) para el reconocimiento de gestos en una aplicación militar [7]. Otro enfoque exitoso se basó en un algoritmo de *Dynamic Time Warping* (DTW) [8]. Si bien estos trabajos describen la aplicación exitosa de varias técnicas de Machine Learning para el reconocimiento de gestos, no se realiza un análisis comparativo del desempeño de los diversos algoritmos.

En este contexto, presentamos un conjunto de técnicas basadas en la aplicación de Machine Learning a reconocimiento de gestos (Sección 2). Particularmente, se implementaron las técnicas (Sección 2.1): *Dynamic Time Warping (DTW)* [9]**,** *Procrustes Analysis* [10], *Markov Chain* [11] y *Hidden Markov Models (HMM)* [12]. La evaluación de la precisión de estas técnicas alcanzo reconocimientos cercanos al 100% (Sección 3). Finalmente, las conclusiones de este trabajo se resumen en la Sección 4

2 Reconocimiento de gestos utilizando técnicas de Machine Learning

Kinect identifica personas dentro del área de detección y calcula la posición en el espacio 3D de 20 partes del cuerpo humano. Estas posiciones son recalculadas 30 veces por segundos y empaquetados en una estructura llamada ‘stick model’. Cada stick model contiene la posición (X, Y, Z) de las 20 partes del cuerpo en un determinado momento. Al observar una secuencia de stick models sucesivos durante un intervalo de tiempo se obtienen los movimientos de las partes del cuerpo. Analizar estos movimientos permite a los desarrolladores reconocer gestos y crear un mecanismo de interacción natural entre humano-computadora.

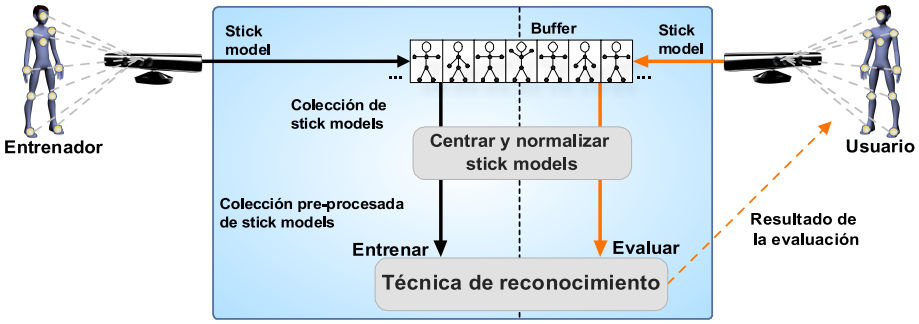

Fig. 1. Enfoque para el reconocimiento de gestos utilizando técnicas de Machine Learning

Los enfoques que utilizan técnicas de Machine Learning para reconocimiento de gestos en general siguen el flujo que muestra la Figura 1. Este flujo consta de dos fases: entrenamiento y reconocimiento. La fase de entrenamiento consiste en entrenar la(s) técnica(s) de reconocimiento a partir de sucesivas repeticiones del gesto a reconocer. Esta fase comienza cuando la persona que cumple el rol de entrenador se posiciona frente al dispositivo Kinect, realiza el movimiento y el *Software Development Kit* (SDK) de Kinect codifica estos movimientos en una secuencia de stick models.

Contando con esta secuencia de stick models, el siguiente paso consiste en centrar y normalizar cada stick model con el objetivo de atenuar las variaciones entre gestos. Estas

variaciones están provocadas por las diferentes ubicaciones del entrenador dentro del campo de detección al momento de realizar el gesto como así también por las diferentes contexturas físicas de los entrenadores y usuarios. Para esto, se aplican dos transformaciones a la secuencia de stick models: centrado y normalizado. El centrado de los stick models consiste en trasladar la secuencia al centro de coordenadas (0, 0, 0). Básicamente, se calcula el centroide de la trayectoria descripta por la posición 3D del torso y se resta este valor a todas las posiciones de las partes del cuerpo de la secuencia de stick models. Una vez centrada la secuencia, cada stick model es normalizado para atenuar diferencia físicas de los usuarios. Este normalizado consiste de multiplicar todas las posiciones de las partes del cuerpo por un factor de escala de acuerdo a la distancia entre el cuello y el toso que es relativa a cada persona [13].

Contando con la secuencia pre-procesada de stick models, el siguiente paso es el entrenamiento de las técnicas. Para esto, cada técnica de manipula convenientemente el conjunto de entrenamiento para ajustar el umbral de aceptación para el gesto.

Finalizada la fase de entrenamiento, se comienza la fase de reconocimiento. Para esto la persona que cumple con el rol de usuario realiza alguno de los gestos entrenados frente al dispositivo Kinect. Al igual que en la fase de entrenamiento, la secuencia de stick models generada es pre-procesada para atenuar la variación entre gestos. En este punto, a diferencia de la fase de entrenamiento, los stick models que corresponden al movimiento son evaluados por la técnica que compara la secuencia con el conjunto de secuencias entrenadas y presenta el resultado de la evaluación al usuario.

Es importante notar que este enfoque ha sido implementado en un prototipo que facilita el agregado de técnicas de Machine Learning contando con capacidades de pre-procesado de stick models y comparación de gestos. En la siguiente subsección, se describen las técnicas soportadas actualmente por el prototipo y que serán utilizadas para evaluar la factibilidad, flexibilidad y potencial uso de cada tipo de técnica.

2.1 Técnicas de Machine Learning para el reconocimiento de gestos.

Las técnicas realizan el reconocimiento de gestos comparando las trayectorias que describen las partes del cuerpo. Las trayectorias que corresponden a un mismo gesto de entrenamiento son utilizadas para ajustar el valor de umbral de aceptación para el gesto. Este umbral se utiliza para detectar si un nuevo gesto se parece a alguno de los gestos entrenado y su cálculo varía de acuerdo a la técnica de Machine Learning elegida. En particular, en este trabajo hemos dividido a las técnicas en dos grupos: técnicas que utilizan directamente las trayectorias y técnicas que necesitan de una transformación previa de las trayectorias para poder ser utilizada.

Para simplificar la explicación, suponga que se quiere reconocer el gesto que consiste en realizar un círculo con la mano izquierda en sentido anti horario. Suponga además que las técnicas son capaces de reconocerlo analizando la trayectoria descripta por esa única parte del cuerpo. Por lo tanto, las técnicas reciben un conjunto de gestos de entrenamiento como entrada. Cada uno de los cuales está representado por una secuencia de stick models. En este punto las técnicas extraen de cada secuencia la trayectoria de la mano izquierda y la utilizan para ajustar el valor de umbral.

2.1.1 Técnicas que se aplican directamente sobre las trayectorias de las partes del cuerpo.

Estas técnicas utilizan directamente las posiciones de las articulaciones para el entrenamiento y posterior reconocimiento de gesto. La primer técnica soporta es *Dynamic Time Warping (DTW)* [9]. DTW mide la similitud entre dos secuencias temporales. En el contexto del reconocimiento de gestos, esas secuencias están formadas por los puntos de las trayectorias que describen las partes del cuerpo. El algoritmo alinea dos trayectorias estirando y encogiendo el eje temporal iterativamente hasta lograr una mínima distancia entre cada par de puntos de las trayectorias. Como resultado se obtiene un valor de distancia que indica cuánto se asemejan las dos trayectorias.

Para entrenar DTW con el gesto círculo se toman las trayectorias de la mano izquierda de todos los gestos de entrenamiento. Luego de aplicar DTW entre cada par de trayectorias, se obtiene una trayectoria modelo que representa al conjunto de entrenamiento y un valor de referencia que marca el límite superior del umbral de aceptación para el gesto. La trayectoria modelo es la trayectoria que más se parece a las demás, es decir, la que tiene menos distancia a cada una de las otras trayectorias de entrenamiento. El valor de referencia es la mayor distancia obtenida al comparar las trayectorias. Cuando el usuario realiza un gesto, la nueva trayectoria es comparada con la trayectoria modelo y si el valor de aplicar DTW es menor al valor de referencia el gesto será aceptado como válido.

La segunda técnica soportada es *Procrustes Analysis* [10]. *Procrustes Analysis* encuentra la alineación óptima entre dos figuras aplicando una serie de transformaciones matemáticas. En el contexto del reconocimiento de gestos, esas figuras están representadas por las trayectorias que describen las partes del cuerpo, es decir, se ve a las trayectorias como figuras estáticas. El algoritmo consta de tres transformaciones que se aplican a las trayectorias consecutivamente: centrar, normalizar y rotar. Las primeras dos son las mismas aplicadas al pre-procesado de los stick models. La última trasformación consiste en rotar las trayectorias hasta obtener una mínima distancia entre ellas. El criterio de mínima distancia utilizado es la diferencia de cuadrados entre los puntos que definen las trayectorias. El resultado de aplicar el análisis de Procrustes sobre dos trayectorias es un valor de distancia que mide cuánto se asemejan las trayectorias y se utiliza de la misma manera que en DTW.

2.1.2 Técnicas que necesitan pre-procesar las trayectorias antes de ser aplicadas.

En el segundo grupo de técnicas se encuentran dos variantes de modelos de Markov: *Markov Chain* [11] y *Hidden Markov Models (HMM)* [12]. Ambas variantes requieren que las trayectorias de las partes del cuerpo sean transformadas en secuencias finitas de estados. Para esto, se aplicó el algoritmo k-means sobre todas las trayectorias de entrenamiento del círculo. Este algoritmo agrupa los puntos de la trayectoria, usando el criterio de cercanía, en un conjunto de clusteres numerados. De esta forma, cada trayectoria de entrenamiento del círculo se transforma en una secuencia numérica donde cada número representa un estado por los que pasó la mano izquierda durante la ejecución del círculo

Una vez que las trayectorias de entrenamiento son expresadas como secuencias finitas de estados, son entregadas como entrada a las variantes de Markov. La primera variante *Markov Chain* se utiliza para representar ciertos procesos estocásticos que modelan el

comportamiento de una o más variables en función del tiempo con la particularidad de que el valor de las variables es independiente del valor histórico de las mismas. Visualmente un *Markov Chain* es una maquina finita de estados donde las transiciones entre estados son probabilísticas. Para utilizar *Markov Chain* en el contexto del reconocimiento de gestos, las probabilidades entre los estados son calculadas a partir de las secuencias de entrenamiento. Cuando el usuario realiza un gesto, el objetivo es determinar si la secuencia de estados correspondiente a la trayectoria de la mano izquierda es una transición de estados válida en el modelo.

La segunda variante *Hidden Markov Models* al igual que las *Markov Chain* se utiliza para representar ciertos procesos estocásticos pero con parámetros desconocidos. Pueden ser vistos como una máquina finita de estados, compuesta de estados ocultos y estados observables donde las transiciones entre estados son probabilísticas. Estos modelos son utilizados para resolver tres problemas canónicos: (1) dado un modelo entrenado, con las probabilidades de transición entre estados, averiguar la probabilidad de que el modelo haya generado una determinada secuencia de salida. (2) dado un modelo entrenado, averiguar la secuencia de estados ocultos que pueden haber generado una determinada secuencia de salida. (3) dado un conjunto de secuencias de salida, averiguar las probabilidades de transición entre estados. Para utilizar HMM en el contexto del reconocimiento de gestos, inicialmente se toman las trayectorias de entrenamiento expresadas como secuencias de estados y se aplica la solución (3) para entrenar el modelo. Cada vez que el usuario realiza un gesto, la secuencia de estados correspondiente a la trayectoria de la mano izquierda es evaluada con (1).

Una vez finalizado el entrenamiento de las técnicas, se obtienen los valores de referencia que marcan el umbral de aceptación para el círculo. Si la técnica entrenada es DTW o *Procrustes Analysis* este valor de referencia indica una distancia mientras que si es *Markov Chain* o HMM indica una probabilidad. En el primer caso, cuando el usuario realiza un gesto, la evaluación de la trayectoria que describe la mano izquierda debe generar una distancia menor al valor de referencia para ser reconocido mientras que en el segundo caso debe generar una probabilidad mayor.

3 Resultados experimentales

En esta sección se describen las pruebas realizadas y el análisis de la precisión de las diferentes técnicas de Machine Learning. Para la evaluación, se utilizó un dataset de 7 gestos distintos que involucran movimientos de las partes superiores e inferiores del cuerpo. Los gestos realizados fueron: Circulo (C), Estiramiento (E), Nadar (N), Smash (S), Punch (P), Swipe a Izquierda (SI) y Swipe a Derecha (SD). Cada gesto fue realizado 20 veces por 4 personas con diferentes contexturas físicas y en diferentes posiciones dentro del campo de detección de Kinect.

3.1 Precisión de las técnicas de reconocimiento

Para evaluar la precisión de las diferentes técnicas utilizamos cross-validation con 10 iteraciones. Las muestras se dividieron aleatoriamente en 10 grupos, 9 de los cuales se utilizaron para entrenar la técnica y el grupo restante se utilizó para evaluarla. Este proceso se repitió 10 veces variando el grupo utilizado para la prueba y de esta forma garan-

tizamos la independencia de los gestos de entrenamiento y prueba en los resultados obtenidos. Como resultado se construyó la matriz de confusión de la Fig. 2que muestra la clasificación obtenida para cada tipo de gesto de entrada, habiendo ejecutado un método de cross-validation de 10 iteraciones y con 80 muestras. Esto significa que cada técnica fue entrenada con 72 muestras y probada con las 8 restante.

Dynamic Time Warping (precisión: 0.991)

↵	C	E	N	S	P	SI	SD
C	76	0	0	0	0	0	0
E	0	80	0	0	0	0	0
N	0	0	80	0	0	0	0
S	0	0	0	80	0	0	0
P	4	0	0	0	80	0	1
SI	0	0	0	0	0	80	0
SD	0	0	0	0	0	0	79

Procrustes Analysis (precisión: 0.8125)

↵	C	E	N	S	P	SI	SD
C	80	0	0	0	0	1	1
E	0	79	0	0	0	0	3
N	0	0	4	0	0	0	0
S	0	0	0	67	3	0	0
P	0	1	17	12	77	0	5
SI	0	0	47	0	0	78	1
SD	0	0	12	1	0	1	70

Markov Chain (precisión: 0.9696)

↵	C	E	N	S	P	SI	SD
C	80	0	1	0	3	7	4
E	0	78	0	0	0	0	0
N	0	2	79	0	0	0	0
S	0	0	0	80	0	0	0
P	0	0	0	0	77	0	0
SI	0	0	0	0	0	73	0
SD	0	0	0	0	0	0	76

Hidden Markov Models (precisión: 0.9892)

↵	C	E	N	S	P	SI	SD
C	80	0	0	0	0	0	0
E	0	80	0	0	0	0	0
N	0	0	79	0	0	0	0
S	0	0	1	79	4	0	0
P	0	0	0	1	76	0	0
SI	0	0	0	0	0	80	0
SD	0	0	0	0	0	0	80

Fig. 2. Matrices de confusión de cada técnica utilizando cross-validation con 80 muestras de cada gesto

Las columnas de cada matriz indican el gesto evaluado mientras que las filas indican la clasificación obtenida para cada gesto. La diagonal principal contiene los gestos correctamente clasificados mientras que fuera de la diagonal se encuentran los gestos clasificados erróneamente.

Para resumir los resultados obtenidos calculamos la precisión de cada técnica sumando los gestos correctamente clasificados y dividiendo por el total de muestras. De esta forma observamos que DTW es la de mayor precisión seguida por HMM reconociendo 99,1% y 98,9% de los gestos respectivamente. Además, de las matrices se puede observar que la técnica con más confusión es *Procrustes Analysis* obteniendo una precisión de 81.2% seguida de *Markov Chain* con una precisión de 96,9%.

3.2 Precisión de las técnicas al variar la cantidad de muestras utilizadas para entrenamiento

Con el fin de determinar la influencia de la cantidad de muestras utilizadas para entrenamiento en la precisión de las técnicas, aplicamos cross-validation de 10 iteraciones variando el tamaño del dataset de prueba. Los tamaños de dataset utilizados fueron de

20, 40, 60 y 80 muestras de cada gesto. El objetivo de esta prueba es determinar la cantidad mínima de muestras necesarias para obtener una precisión aceptable, y conocer hasta qué punto las técnicas incrementan su precisión al incrementar la cantidad de muestras utilizadas para entrenamiento.

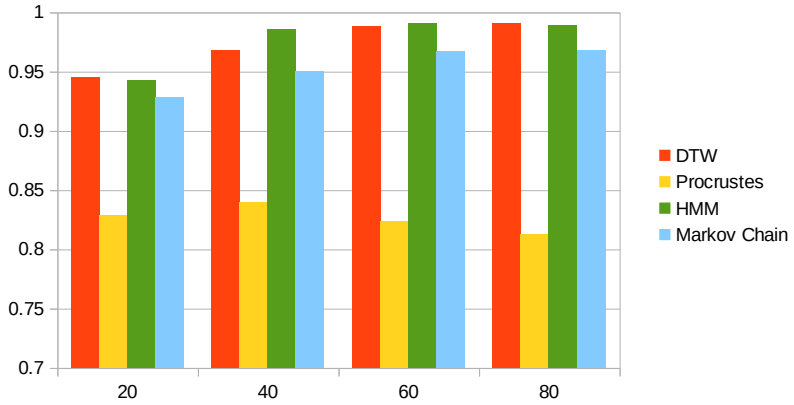

Fig. 3. Grafico que compara la cantidad de gestos reconocidos al variar la cantidad de muestras usadas para el entrenamiento

La Figura 3 muestra los resultados obtenidos. Los tamaños de dataset utilizados se listan en el eje horizontal mientras que la precisión alcanzada se lista en el eje vertical. Del gráfico se puede observar que incrementar la cantidad de muestras utilizadas para el entrenamiento mejora la precisión de DTW un 4.1% cuando se incrementa de 20 a 80 el tamaño del dataset de prueba. Si bien este aumento es notorio, se ve desacelerado al llegar a un dataset de 40 muestras por gesto. Por ejemplo, HMM mejora la precisión en 0.35% cuando se incrementa de 40 a 80 el tamaño del dataset.

4 Conclusiones

En este trabajo hemos evaluado las técnicas *Dynamic Time Warping*, *Procrustes Analysis*, *Markov Chain* y *Hidden Markov Models* en el reconocimiento de gestos corporales, las cuales utilizan las posiciones de las partes del cuerpo provistas por Kinect. Los experimentos mostraron que *Dynamic Time Warping* y *Hidden Markov Models* son las más precisas reconociendo 99,1% y 98,9% de los gestos respectivamente. La técnica con menor precisión fue *Procrustes Analysis* reconociendo el 81,2% de los gestos. Además, como era de esperarse, el incremento de la cantidad de muestras utilizadas para entrenamiento incrementa la precisión de las técnicas. Sin embargo, este incremento no es tan significativo al superar las 40 muestras por gesto.

Como trabajo futuro sería importante medir el tiempo requerido por cada técnica tanto para ser entrenada como para evaluar un nuevo gesto. Además, ver cuánto influye en este tiempo la cantidad de gestos a reconocer en simultáneo. De esta forma podríamos saber cuál se ajusta mejor para ser utilizada en tiempo real.

Por último, sería interesante realizar nuevas pruebas con un conjunto más variado de gestos, incorporando otras técnicas de inteligencia artificial como *Support Vector Machines* y *Decision Trees*.

Agradecimientos

Agradecemos el apoyo financiero brindado por la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) a través del proyecto PICT 2011-0080.

Bibliografía

[1] J. Boehm, "Natural user interface sensors for human body measurement," *Int Arch Photogramm Remote Sens Spat. Inf Sci*, vol. 39, p. B3, 2012.

[2] E. A. Suma, B. Lange, A. Rizzo, D. M. Krum, and M. Bolas, "FAAST: The Flexible Action and Articulated Skeleton Toolkit," in *Virtual Reality Conference (VR), 2011 IEEE*, 2011, pp. 247 –248.

[3] F. Kistler, B. Endrass, I. Damian, C. Dang, and E. André, "Natural interaction with culturally adaptive virtual characters," *J. Multimodal User Interfaces*, vol. 6, no. 1–2, pp. 39–47, 2012.

[4] V. Thiruvarudchelvan and T. Bossomaier, "Towards realtime stance classification by spiking neural network," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 2012, pp. 1–8.

[5] A. Bleiweiss, D. Eshar, G. Kutliroff, A. Lerner, Y. Oshrat, and Y. Yanai, "Enhanced interactive gaming by blending full-body tracking and gesture animation," in *ACM SIGGRAPH ASIA 2010 Sketches*, 2010, pp. 34:1–34:2.

[6] X. Yang and Y. Tian, "EigenJoints-based action recognition using Naïve-Bayes-Nearest-Neighbor," in *CVPR Workshops*, 2012, pp. 14–19.

[7] S. Bhattacharya, B. Czejdo, and N. Perez, "Gesture classification with machine learning using kinect sensor data," in *Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on*, 2012, pp. 348–351.

[8] C. Waithayanon and C. Aporntewan, "A Motion Classifier for Microsoft Kinect," in *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, 2011, pp. 727–731.

[9] S. Salvadore and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," presented at the 3rd Workshop on Mining Temporal and Sequential Data, 2004.

[10] A. Ross, "Procrustes analysis," *Course Rep. Dep. Comput. Sci. Eng. Univ. S. C.*, 2004.

[11] K. Lange, "Finite-State Markov Chains," in *Numerical Analysis for Statisticians*, Springer, 2010, pp. 503–526.

[12] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[13] K. Lai, J. Konrad, and P. Ishwar, "A gesture-driven computer interface using Kinect," in *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, 2012, pp. 185 –188.