



MODELADO DE EXPRESIONES PARA UNA CARA ROBÓTICA

Oscar Déniz Suárez

Licenciado en Informática. Estudiante de doctorado
Universidad de Las Palmas de Gran Canaria.

INTRODUCCIÓN

En el marco del desarrollo de interfaces hombre-máquina cada vez más flexibles y sencillos de utilizar, se ha realizado en los últimos años un esfuerzo considerable en dotar de características y habilidades humanas a dispositivos físicos destinados a estar en íntimo contacto con personas, como por ejemplo los robots móviles. Obviamente, si el aspecto del robot, sus movimientos, y sus capacidades de más alto nivel se asemejan hasta cierto punto a las humanas, la respuesta de las personas que interactuarán con él será más positiva. Éstas se sentirán más atraídas por el dispositivo, se despertará su curiosidad, y la comunicación será más fluida. Escenarios típicos en los que este tipo de robots resultaría muy adecuado serían museos, puestos de atención al público, juguetes, etc.

De todas las partes del cuerpo humano, la que mayor información aporta en el proceso de comunicación persona-persona es sin duda la cara (la cabeza es la única parte del cuerpo donde se encuentran los cinco sentidos). Las posibilidades de comunicación que nos aporta la cara son realmente destacables, y todos somos capaces de comprobarlo en nuestras relaciones diarias con otras personas. En la mayoría de las ocasiones las expresiones que adopta la cara permiten complementar de forma significativa la información que se transmite con el habla. Otras veces, las expresiones «lo dicen todo». No es de extrañar por tanto, que una gran parte del esfuerzo investigador arriba mencionado se haya dedicado al estudio y desarrollo de caras de aspecto humano o animal. En unos casos se ha optado por emplear gráficos de ordenador, con la ventaja principal de no tener prácticamente limitaciones de tipo físico. En otros casos se han diseñado y construido dispositivos físicos, que si bien presentan limitaciones, aportan una componente no despreciable: las personas perciben con mayor interés al dispositivo que es físico, real. El dispositivo físico puede además ser observado desde diferentes ángulos. Por muy complejos que sean los gráficos, el parecido a la cara humana debe completarse con esa existencia física. En adelante nos centraremos en esta última opción.

En este artículo se presenta la descripción de un esquema de modelado de expresiones para una cara robótica. El software que lo implementa constituye uno de los módulos

de un proyecto actualmente en curso denominado CASIMIRO (*Cara Expresiva y Procesamiento Visual Básico para Robots Interactivos*). Por modelado de expresiones se entiende el proceso por el cual un diseñador define, de una forma más o menos sencilla, cuáles serán las expresiones que podrá presentar la cara, las posiciones asociadas de los distintos motores y cómo se realiza el paso de una expresión a otra. El modelado de expresiones debe emplear el mayor nivel de detalle posible, con el fin de dotar de más flexibilidad y posibilidades a la cara. No obstante, es fundamental que el modelado no se convierta en un proceso excesivamente complejo y sobre todo que pueda hacerse de forma interactiva, es decir, comprobando con el robot físico el aspecto de las distintas expresiones, las transiciones entre ellas, etc. En el siguiente apartado se describen brevemente otros trabajos relacionados, para a continuación presentar el marco de modelado. Por último, se describirá el software desarrollado y se discutirán posibles ampliaciones futuras.

TRABAJOS RELACIONADOS

Una cara robot muy simple es la del robot móvil Minerva¹. La cara tiene cuatro grados de libertad, uno para cada una de las cejas y dos para la boca. Las cejas se mueven rotando sobre su centro. La boca está constituida por una banda elástica roja. Cada extremo de la banda se halla enganchado a un brazo del servomotor, y el movimiento está limitado por tres pins. A pesar de la simplicidad del hardware, la cara de Minerva es capaz de producir un efecto significativo sobre el observador. Minerva podía adoptar cuatro expresiones básicas: neutral, sonrisa, tristeza y enfado.

El trabajo más relacionado con el proyecto CASIMIRO es Kismet². Kismet es una cara robótica desarrollada en el MIT que consiste de un sistema de visión activa estéreo y de características faciales que dotan al robot de un aspecto parecido al de un animal. Las características faciales que incluye son cejas (cada una con dos grados de libertad), orejas (cada una con dos grados de libertad), párpados (un grado de libertad) y boca (un grado de libertad). En versiones recientes se han incluido nuevos grados de libertad en partes como la boca. El robot es capaz de adoptar expresiones de enfado, fatiga, temor, disgusto, excitación, felicidad, interés, tristeza y sorpresa, todas

fácilmente interpretables para el observador humano. El sistema motor de Kismet se dividió en tres niveles. En el nivel más bajo se encuentran procesos que controlan cada motor. En el nivel siguiente, existen procesos que coordinan el movimiento de las características faciales. Por último, en el tercer nivel existen procesos que se encargan de disponer las características faciales para conformar las distintas expresiones. La ventaja de este esquema es evidente: la descomposición del trabajo de modelado de una forma natural y fácilmente escalable. Es por ello que en el modelado de expresiones de CASIMIRO se ha empleado este mismo esquema básico. En Kismet también se emplea el concepto de intensidad de la expresión. La intensidad no es más que un grado en que se adopta una expresión, con respecto a una pose que se considera neutral (que no parece conformar ninguna expresión). Los niveles superiores del sistema serán los encargados de suministrar la intensidad con la que se quiere adoptar determinada posición.

Las expresiones básicas que puede adoptar son: neutral, felicidad, enfado, sorpresa, tristeza, temor, disgusto, «bebido» y vergüenza.

El robot WE-3RIV (Waseda Eye Nº 3 Refined IV)³ presenta un hardware avanzado constituido por un total de 26 grados de libertad y multitud de sensores. La cara presenta nada menos que 21 grados de libertad: 4 para los globos oculares, 4 para los párpados, 8 para las cejas, 4 para los labios y 1 para la mandíbula. Las expresiones básicas que puede adoptar son: neutral, felicidad, enfado, sorpresa, tristeza, temor, disgusto, «bebido» y vergüenza. Al igual que en Kismet, se utiliza una medida de intensidad (de 50 grados) para las expresiones. La intensidad se emplea para hacer una interpolación de la expresión, con respecto a la expresión neutral.

ESQUEMA DE MODELADO

Para modelar las expresiones en el robot CASIMIRO, se emplea la jerarquía de tres niveles explicada en el apartado anterior. Se definen grupos de motores que constituyan una característica facial concreta. Por ejemplo, se agrupan dos motores determinados para el control de una ceja. Para cada uno de los grupos de motores definidos, se especifican las poses que se desea presente la característica facial. Por ejemplo, ceja derecha neutral, ceja derecha levantada, ceja derecha hacia el interior, etc. Por defecto, se realiza la transición entre esas poses en línea recta (en el espacio de valores de consigna de los motores), pero se debe dar al modelador la oportunidad de modificar de alguna forma la trayectoria descrita en las transiciones, porque algunas pueden resultar no naturales. En nuestro

caso particular, se optó por dar la oportunidad de introducir puntos intermedios en la trayectoria de transición. Adicionalmente, se otorga la posibilidad de especificar una velocidad entre cada dos puntos de la trayectoria. Se analizó la posibilidad de emplear interpolación no lineal (splines), pero se llegó a la conclusión de que no resultaría necesaria para un modelado aceptable. Una de las poses introducidas (la primera) es la pose neutral. Las poses introducidas son las de grado máximo (por ejemplo 100, si se usa el rango 0-100). El grado de la pose (que se especifica solo en ejecución) se utiliza para interpolar linealmente los puntos de la trayectoria, con respecto a la pose neutral. Con respecto al tercer nivel de la jerarquía comentada, las expresiones hacen referencia a poses de los diferentes grupos, cada una con un grado determinado. Por ejemplo, la expresión «Sorpresa» podría representarse simplificada por los siguientes datos:

<i>Expresión: "Sorpresa"</i>		
<i>Grupo</i>	<i>Pose</i>	<i>Grado</i>
Boca	Abierta	90
Ceja derecha	Levantada	90
Ceja izquierda	Levantada	90
Oreja derecha	Levantada	100
Oreja izquierda	Levantada	100
Párpado derecho	Levantado	80
Párpado izquierdo	Levantado	80

A su vez, la expresión a adoptar puede especificarse en ejecución acompañada de un grado. Este grado permite, mediante multiplicación, obtener el grado concreto a aplicar a las poses de los distintos grupos. Para obtener un mayor control, se permite especificar un tiempo de comienzo de cada grupo. De esta forma, podría por ejemplo conseguirse que al adoptar la expresión de sorpresa, primero se levantarán las cejas y después se abra la boca. En este punto es fácil ver que el empleo de la jerarquía de tres niveles comentada permite realizar movimientos de partes individuales, como por ejemplo guiñar un ojo, parpadear, mover la boca para hablar, etc. Con respecto a este último caso, puede además combinarse el movimiento de la boca con otras poses de otras características faciales, dando lugar a combinaciones como hablar con expresión de sorpresa, de enfado, etc.

TRANSICIONES ENTRE EXPRESIONES

En el apartado anterior se indicó que en tiempo de ejecución las poses de los grupos de motores podían adoptarse en cierto grado, mientras que las poses que se introducían correspondían al grado máximo. Queda pues por resolver la cuestión de obtener la trayectoria de transición de una pose A con un grado cualquiera G_i a una pose B con un grado cualquiera G_f . En otras palabras, en un momento determinado el grupo se encuentra en la pose A con grado

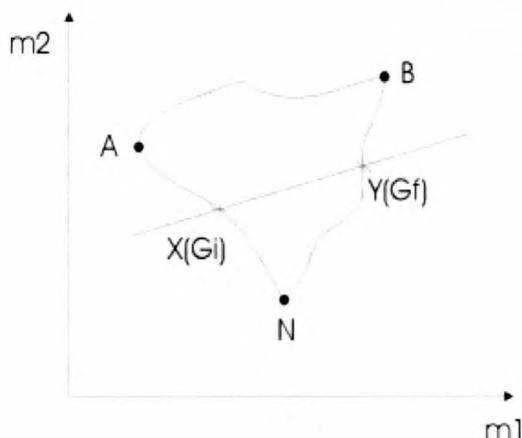
G_i , y se desea adoptar la pose B con grado G_f . Para un grupo de dos motores, el escenario sería el representado en la figura siguiente, donde N representa la pose neutral, X el punto correspondiente al grado inicial e Y el punto correspondiente al grado final. Para obtener la expresión que nos dará la trayectoria buscada, fijamos las siguientes condiciones extremo:

- Si $G_i=0 \rightarrow T=XNY$
- Si $G_f=0 \rightarrow T=XNY$
- Si $G_i=1 \rightarrow T=(1-G_f) \cdot XNY + G_f \cdot XABY$
- Si $G_f=1 \rightarrow T=(1-G_i) \cdot XNY + G_i \cdot XABY$

La ecuación que cumple estas restricciones es:

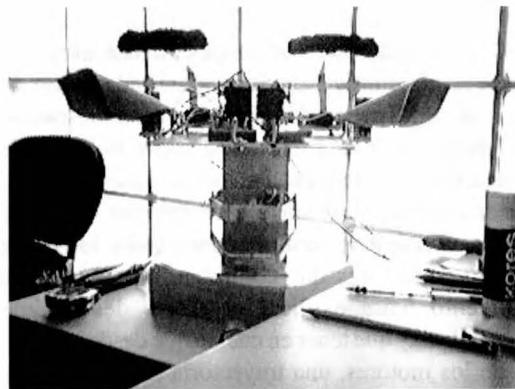
$$T = [1 - ((G_i \geq G_f) ? G_f : G_i)] \cdot XNY + ((G_i \geq G_f) ? G_f : G_i) \cdot XABY$$

Donde los símbolos ? y : corresponden a la instrucción IF-THEN-ELSE. Puede observarse además que la ecuación de trayectoria es continua en los valores de G_i y G_f . La misma relación se utiliza para hallar las velocidades de la trayectoria.

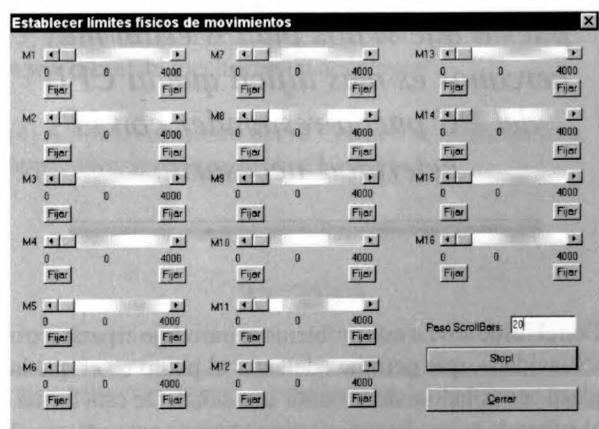


SOFTWARE IMPLEMENTADO

El esquema de modelado descrito se implementó en un programa de ordenador que se denominó «editor de poses». El cometido del editor de poses es proporcionar al modelador una forma sencilla e interactiva de definir las poses, transiciones, etc. La cara robot con la que se probó el software desarrollado es una versión inicial de CASIMIRO. El aspecto de la cara aparece en la figura siguiente. La cara robot está dotada de 9 motores: uno para la boca, 2 para la ceja izquierda, 2 para la ceja derecha, uno para la oreja derecha, uno para la oreja izquierda y dos para los párpados. Todos los motores empleados son servomotores del mismo modelo, conectados a una placa controladora ASC16 de Medonis⁴, que es a su vez gobernada desde un PC mediante una conexión serie.



En una primera fase del desarrollo se elaboró una librería de bajo nivel con el fin de proporcionar una forma de enviar comandos a la placa controladora de motores. El editor de poses se basa en esta librería. La filosofía que sigue el editor de poses es la de dar la oportunidad al modelador de especificar una serie de datos sobre la cara y sus movimientos, y grabar todos estos datos en un fichero de poses. Este fichero será posteriormente lo único que se necesite para reproducir (y generar) los movimientos. El editor de poses está pues pensado para funcionar tanto en tiempo de modelado como en tiempo de ejecución, y puede controlarse por otras aplicaciones. Los pasos que seguiría el modelador para empezar el diseño serían: conectar con la placa ASC16, establecer los límites físicos de movimiento y especificar poses y transiciones. Los límites físicos de movimiento son los valores máximos y mínimos que pueden tomar los motores. Estos valores se introducen por el modelador y quedan guardados en el fichero de poses. La siguiente figura muestra la ventana de especificación de límites de movimiento.

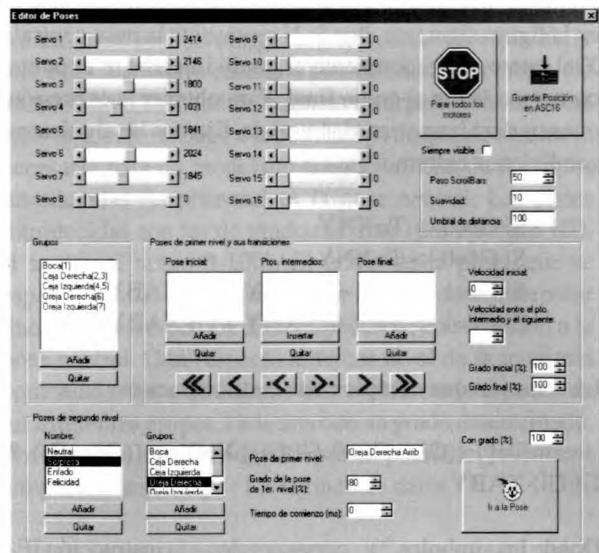


La parte fundamental del programa es la que permite definir los grupos de motores, poses y transiciones. La ventana de esta parte del editor de poses se muestra a continuación. En la parte superior izquierda se encuentran las barras de desplazamiento que permiten mover los

motores. En la parte central izquierda se definen los grupos de motores. En la parte central derecha se definen las poses del grupo seleccionado, y las transiciones entre ellas. Para modelar las transiciones pueden introducirse puntos intermedios y velocidades en cada punto de la trayectoria. En la parte inferior de la ventana se especifican las expresiones. Todos los datos especificados por el modelador se graban en el fichero de poses. En la parte superior derecha de la ventana se dispone de un botón STOP que permite parar al momento todos los motores, así como parámetros que controlan la continuidad del movimiento. Para explicar el efecto del parámetro «Suavidad» hay que tener en cuenta que desde el punto de vista de los motores, una trayectoria es un conjunto de puntos en el espacio motor que los motores siguen en línea recta. Cada tramo de la trayectoria puede tener una velocidad distinta, y además las velocidades definitivas de los motores se modifican para que todos acaben en el mismo instante, aunque el recorrido sea distinto (las velocidades de los servos empleados no son lineales, con lo que hubo de crearse una tabla de conversión). Cada grupo de motores es controlado por una línea de ejecución separada (*thread*), y todos los accesos al puerto serie han de sincronizarse. La placa controladora de motores emite una señal cuando un motor ha alcanzado un punto de la trayectoria. En ese momento, el editor de poses debe enviar a la placa el comando para que el motor se desplace al punto siguiente, con la velocidad asociada al tramo. Desde que se recibe la señal de la placa controladora hasta que se envía el comando de desplazamiento al punto siguiente pasa un tiempo (que dependerá de la carga de la CPU del PC) en el que el motor estará parado, lo cual hace que el movimiento no sea continuo.

El objetivo es también evitar discontinuidades en el movimiento, puesto que si dos puntos están muy cercanos es más difícil que la CPU del PC pueda responder con la celeridad necesaria.

Con el fin de aliviar este problema se introdujo el parámetro «Suavidad», que permite adelantar el punto en el que la placa controladora debe emitir una señal. De esta forma, el editor de poses dispone de cierto tiempo antes de que el motor llegue realmente al punto de parada. Por otro lado, el parámetro «Umbral de distancia» sirve para eliminar de las trayectorias los puntos consecutivos que estén muy cercanos. El objetivo es también evitar discontinuidades en el movimiento, puesto que si dos puntos están muy cercanos es más difícil que la CPU del PC pueda responder con la celeridad necesaria.



Para que el editor de poses pueda funcionar adecuadamente en tiempo de ejecución se incluyó un buffer de peticiones, que se encarga de mantener momentáneamente las peticiones de «Ir a pose X en grado Y» que vengan de otras aplicaciones. El buffer de peticiones es necesario porque las peticiones tardan un cierto tiempo en completarse. El buffer de peticiones comprueba en todo momento las peticiones en espera. Aquellas que sean compatibles con la que actualmente se está ejecutando (no tienen ningún motor en común) son lanzadas a ejecución, pero siempre respetando el orden en que se hizo las peticiones.

El parpadeo se implementaría con el esquema descrito mediante dos comandos consecutivos: uno para mover el párpado izquierdo y otro para mover el derecho.

Un caso especial a mencionar es el del parpadeo. El parpadeo se implementaría con el esquema descrito mediante dos comandos consecutivos: uno para mover el párpado izquierdo y otro para mover el derecho. No obstante, el parpadeo requiere de un tratamiento particular: los párpados han de volver a su posición original, sea cual sea. Además, el movimiento ha de hacerse siempre a la máxima velocidad posible. Por consiguiente, de forma automática los dos comandos de movimiento se convierten en cuatro, y se emplea la máxima velocidad. En el caso de solo querer guiñar un ojo, un comando de movimiento se

convierte en dos. Por otro lado, debido a que en la implementación desarrollada el movimiento de los párpados se ejecutará por dos *threads* distintos y a que el desplazamiento es muy corto y a gran velocidad (lo cual como hemos visto puede provocar que el PC no responda con la celeridad deseable), se impone la siguiente restricción: solo se puede parpadear cuando no hay ningún otro motor de la cara funcionando. De esta forma podemos asegurar la sincronía en el movimiento de los dos párpados. Esta restricción se considera razonable pues los movimientos de la cara para cambiar de expresión duran relativamente poco. La restricción implicaría que la cara no podría hablar (mover la boca) y parpadear al mismo tiempo, pero igualmente se considera aceptable en tanto en cuanto CASIMIRO no está destinado a realizar monólogos, y en cualquier caso, siempre hay pausas en el discurso que permitirían parpadear. El parpadeo no puede iniciarse desde el editor de poses, ha de hacerse con otra aplicación.

La restricción implicaría que la cara no podría hablar (mover la boca) y parpadear al mismo tiempo, pero igualmente se considera aceptable en tanto en cuanto CASIMIRO no está destinado a realizar monólogos, y en cualquier caso, siempre hay pausas en el discurso que permitirían parpadear.

CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha descrito un programa de modelado de expresiones para una cara robótica. Un software que permita modelar de forma sencilla e interactiva los movimientos de una cara electromecánica constituye una herramienta muy útil sobre todo en las primeras fases del desarrollo de la misma, cuando se añaden o eliminan motores o piezas. El repertorio de poses es fácilmente escalable con el número de motores que se empleen. El software siempre puede dotarse de una mayor flexibilidad, pero hay que tener en cuenta que el objetivo no debe ser otro que el de conseguir que la cara se mueva lo mejor posible, desde el punto de vista de un observador humano. Quiere esto decir que no debe faltar flexibilidad pero tampoco debe hacerse el programa excesivamente complejo.

Aunque el esquema de modelado presentado es en sí mismo muy flexible, existen posibilidades que la versión

actual aún no contempla. Es particular, la «mezcla» de expresiones sería un aspecto a tener en cuenta. Por mezcla de expresiones se entiende la posibilidad de que la cara adopte simultáneamente dos o más expresiones, dando lugar a casos como por ejemplo presentar un aspecto de sorpresa a la vez que de felicidad. Dadas las posibilidades actuales del hardware disponible de la cara de CASIMIRO, se consideró que la expresividad de la misma no era suficiente para justificar el desarrollo de la mezcla de expresiones, aunque en un futuro debería ser tenida en cuenta.

Dadas las posibilidades actuales del hardware disponible de la cara de CASIMIRO, se consideró que la expresividad de la misma no era suficiente para justificar el desarrollo de la mezcla de expresiones, aunque en un futuro debería ser tenida en cuenta.

BIBLIOGRAFÍA

- [1] Minerva: Carnegie Mellon's Robotic Tourguide Project. <http://www-2.cs.cmu.edu/~minerva/>
- [2] Kismet: <http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html>
- [3] An Anthropomorphic Head Robot WE-3RIV. <http://www.takanishi.mech.waseda.ac.jp/eyes/>
- [4] ASC16: Advanced Servo Controller, Medonis Eng. <http://www.medonis.com/asc16.html>

AUTOR



Oscar Déniz Suárez: Licenciado en Informática. Estudiante de doctorado y becario de investigación adscrito al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria. Sus intereses investigadores se centran en robótica, interfaces percepto-efectores, y reconocimiento de caras.