Interfaz hombre-máquina basado en el análisis de gestos faciales mediante visión artificial

Jose F. Velasco, Daniel Pizarro, Javier Macias y Cristina Losada
Departamento de Electrónica
Universidad de Alcalá
Email: {jose.velasco, pizarro, macias y losada}@depeca.uah.es

***Resumen*—En el presente documento se propone una aplicación, orientada a la ayuda a la discapacidad, en la que mediante el análisis por visión artificial de los gestos faciales realizados por el usuario, se provee una interfaz alternativa de comunicación entre el usuario y una máquina. Para este fin se propone el uso de un Modelo de Apariencia Activa, al cual se le han añadido parámetros adicionales para modelar los cambios en las condiciones de iluminación.**

I. Introducción

En la actualidad, el uso de herramientas tales como los ordenadores, en las que son necesarios comandos complejos para su uso y en las que se desarrollan entornos gráficos cada vez mas interactivos e intuitivos, se ha convertido en habitual en nuestra sociedad.

Para la mayoría de las personas resulta sencillo manejar este tipo de herramientas mediante diversos dispositivos hardware estandarizados. Sin embargo, existen personas que, por diversas razones, tienen dificultades para utilizar este tipo de dispositivos.

Como alternativa, en el presente trabajo se propone el uso de la visión artificial para la detección de determinados gestos faciales que, mediante su posterior procesado, permitan al usuario realizar comandos específicos a sus necesidades.

El uso de la visión artificial tiene como gran ventaja su carácter no intrusivo, es decir, no requiere la colocación de dispositivos en el rostro del usuario. Además, el uso de cámaras se ha generalizado, pudiendo encontrar dispositivos de bajo coste integrados en la mayoría de ordenadores portátiles.

En este trabajo se ha decidido utilizar el rostro humano dado que presenta un amplio abanico de movimientos y posturas posibles (lo cual puede facilitar la comunicación entre el humano y la máquina). Además el rostro humano tiene la virtud de conservar la movilidad en ciertas circunstancias en las que se pierde esa capacidad en las manos, como por ejemplo en tetraplejias o amputaciones.

La técnica en la que se basa este trabajo utiliza los denominados Modelos de Apariencia Activa (Active Appearance Models, en inglés), a partir de ahora AAM. Propuesta originalmente por [1], esta técnica ha sido objeto de diversas y profundas revisiones en los últimos años; entre ellas destaca [2].

No obstante, esta técnica, al igual que otras también basadas en el registro de imágenes, presentan ciertos problemas cuando la escena en la que se encuentra el objeto al que se pretende ajustar el modelo, ha sido sometida a cambios notables de iluminación. Para solventar estos problemas, [3] propone construir el modelo de un AAM, y realizar su ajuste en un espacio invariante a la iluminación basado en [4].

En este trabajo se añadirá al enfoque clásico de los AAM, el espacio invariante sugerido por [3] además de parámetros fotométricos de ganancia y offset, propuesto por [5] para las técnicas de registro de superficies rígidas en imágenes.

En este artículo se demuestra la aplicabilidad real de un sistema AAM ante cambios de iluminación severos en la escena y mediante el uso de cámaras de bajo coste.

II. Trabajo Previo

El contenido de esta sección se divide en tres partes. En la primera, se revisarán los modelos de apariencia activa y su aplicación a la detección de caras. En la segunda parte se describirán los fundamentos de cómo se forma el color en una imagen, necesarios para explicar los espacios invariantes a la iluminación utilizados en este trabajo, que serán descritos en último lugar.

II-A. Modelos de Apariencia Activa

Los Modelos de Apariencia Activa (AAM) permiten reproducir de forma sintética imágenes de superficies que incluyen deformaciones no rígidas y cambios de apariencia. Están basados en la obtención, mediante una fase de entrenamiento, de un modelo estadístico de la **forma** y la **apariencia** del objeto de interés [1].

Parámetros de Forma: En un AAM la forma es descrita mediante un conjunto de N puntos característicos, que determinan una malla similar a la representada en la figura 1 y que es expresada por el siguiente vector:

$$s = (u_1, v_1, u_2, v_2, \cdots, u_N, v_N) \tag{1}$$

donde u_i, v_i son las coordenadas del vértice i.

Mediante el análisis de componentes principales (PCA) sobre las mallas de entrenamiento se obtiene una malla media s_0 y un subespacio $B_S = [s_1, \cdots, s_n]$ formado por n componentes principales, con una dimensionalidad menor que la del conjunto de entrenamiento.

Cualquier instancia de la forma del modelo se obtiene a partir de una combinación lineal de los vectores de la base de

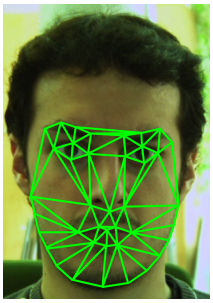


Figura 1. Ejemplo de Malla en AAM

forma, B_S mediante la siguiente expresión:

$$s(p) = s_0 + \sum_{i=1}^{n} p_i s_i \tag{2}$$

La inclusión de los parámetros de forma en el modelo se realiza mediante una transformación afín definida a trozos denominada **función warp** $W(x;p)$. Esta función se encarga de transformar los puntos interiores de una malla concreta (normalmente se elige s_0), en donde se define la apariencia, a cualquier malla $s(p)$ generada a partir de (2). Es decir:

$$x' = W(x;p) \tag{3}$$

donde x son puntos en el interior de s_0 y x' está definido en el interior de $s(p)$.

Parámetros de Apariencia: La apariencia se describe a partir del mapa de bits definido en el interior de los diversos triángulos que forman los puntos de la malla s_0.

Mediante (3) se transforman las imágenes de entrenamiento, con el fin de normalizarlas en forma. De la misma manera que con los parámetros de forma, mediante PCA se obtienen tanto la apariencia media A_0, como la base de un subespacio $B_A = [A_1(x), A_2(x), \cdots, A_m(x)]$, de dimensión menor al conjunto de entrenamiento, que está formada por las m componentes principales del entrenamiento.

A partir de estos elementos, se obtiene un modelo de apariencia lineal, que es capaz de generar una instancia de apariencia a partir de una combinación lineal de la media y las componentes de la base, ponderadas por un conjunto de parámetros $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)$:

$$A(x;\lambda) = A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) \tag{4}$$

Ajuste del modelo: El ajuste del modelo trata de, a partir de una imagen de entrada $I(\hat{x})$, encontrar el conjunto de parámetros p y λ que minimicen el error cuadrático entre la instancia del modelo generado a partir de esos parámetros y la imagen de entrada:

$$\sum_{x} \left[A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(W(x;p)) \right]^2 \tag{5}$$

En [2] se describen diversos métodos para minimizar (5) entre los que destacan por su precisión el algoritmo de **Lucas-Kanade**, el cual , en lineas generales, es un método de minimización iterativo basado en el algoritmo de Gauss-Newton.

II-B. Fundamentos en la Formación del color en una Imagen

A continuación se presenta el modelo físico usado para describir el proceso de formación de una imagen. La teoría del espacio invariante de iluminación, tomada de [5] y descrita brevemente a continuación, está basada sobre las siguientes suposiciones: se considerará que todas las superficies son lambertianas, que la iluminación sigue el modelo de Plank, y que el sensor de la cámara tiene un ancho de banda estrecho. Para obtener información más detallada sobre este modelo se recomienda la lectura de [4].

Con todo lo anterior, el color RGB obtenido en cada píxel, bajo las suposiciones anteriores, responde al siguiente modelo físico:

$$\rho_k = \sigma S(\lambda_k) E(\lambda_k, T) Q_k \delta(\lambda - \lambda_k) \tag{6}$$

donde $\sigma S(\lambda_k)$ representa la reflectancia espectral de la superficie ponderada por el factor lambertiano σ. El término $Q_k \delta(\lambda - \lambda_k)$ representa la respuesta espectral del sensor en función del color de cada canal k centrado en la longitud de onda λ_k. $E(\lambda_k, T)$ es la distribución espectral de potencia de la luz en el modelo de Plank, en la aproximación de Wien, que esta modelada por la siguiente expresión:

$$E(\lambda, T) = I c_1 \lambda^{-5} e^{(\frac{-c_2}{T\lambda})} \tag{7}$$

El anterior modelo es válido para un amplio abanico de temperaturas $T \in [2500°, 10000°]$. El término I es la intensidad global de luz y las constantes c_1 y c_2 son fijas. De acuerdo con este modelo, el valor obtenido en cualquier píxel ρ_k de la cámara es obtenido directamente por la siguiente expresión:

$$\rho_k = \sigma I c_1 \lambda^{-5} e^{(\frac{-c_2}{T\lambda})} S(\lambda_k) Q_k \delta(\lambda - \lambda_k) \tag{8}$$

II-C. Espacio Invariante de Iluminación

La transformación que permite realizar una imagen invariante a la iluminación está basada en el trabajo original de [4], en donde se explica un método de obtención de una imagen invariante a la iluminación intrínseca a la imagen en color de entrada. Dicho método esta basado en el anteriormente mencionado modelo de formación de la luz, basado en la asunción de superficies lambertianas, sensores de banda estrecha e iluminación plankiana.

Dadas las tres componentes de color $\rho = (\rho_1, \rho_2, \rho_3)$ descritas en (8), se forman los logaritmo de las relaciones de cromaticidad.

$$\begin{aligned} \chi_1 &= \log(\tfrac{\rho_1}{\rho_3}) = \log(s_1/s_3) + (e_1 - e_3)/T \\ \chi_2 &= \log(\tfrac{\rho_2}{\rho_3}) = \log(s_2/s_3) + (e_2 - e_3)/T \end{aligned} \tag{9}$$

Donde $e_k = -c_2/\lambda_k$ solo depende de la respuesta espectral de la cámara y no de la superficie y $s_k = c_1 \lambda_k^{(-5)} S(\lambda_k) Q_k$ no depende de la temperatura de color T.

Para cualquier iluminación (cambio en la temperatura de color T) de una misma superficie con un determinado color, el par de valores χ_1 y χ_2 caen sobre una linea con la dirección del vector $\bar{e} = (e_1 - e_3, e_2 - e_3)$. En consecuencia, el vector $\chi = (\chi_1, \chi_2)$ se mueve a lo largo de esa linea en función de la temperatura de las diferentes iluminaciones T.

Puede encontrarse un valor invariante a la iluminación proyectando el vector χ sobre la dirección ortogonal a $\bar{e}$ definida por $\bar{e}^{\perp} = (cos(\theta), sin(\theta))$. De esta manera, dos píxeles de la misma superficie vistos con distinta iluminación serían proyectados en el mismo lugar.

Para reducir la arbitrariedad de la elección de las relaciones de cromatricidad en [4], se propone un método que usa la media geométrica $(\rho_1\rho_2\rho_3)^{1/3})$ de los valores de los tres canales como denominador. De este modo, se obtiene un vector de tres coordenadas (una por cada canal RGB) linealmente dependiente.

Mediante la elección de una descomposición adecuada se obtiene un vector bidimensional equivalente a χ que conserva la propiedades esenciales de la ecuación (9). La transformación $\mathcal{L}$ se obtiene de manera sencilla proyectando el vector χ sobre linea invariante parametrizada por su pendiente θ:

$$\mathcal{L}(\rho, \theta) = \chi_1(\rho)\cos(\theta) + \chi_2(\rho)\sin(\theta) \tag{10}$$

Esta transformación, como se ha señalado previamente, representa la relación entre el color de la imagen y su correspondiente representación invariante a la iluminación. La transformación es global por lo que no depende de la posición del píxel $q \in \mathbb{R}^2$ sino de únicamente su valor de color. El ángulo de pendiente θ de la recta invariante solo depende de las propiedades espectrales de la cámara.

En [4] se presenta un método para obtener la pendiente θ a partir de un paso de calibración usando o bien un patrón de color o un conjunto de imágenes registradas previamente con la cámara en cuestión bajo cambios en la iluminación.

En [6] se presenta un método de autocalibración para obtener dicha pendiente mediante la búsqueda de la pendiente que hace que la entropía de la representación invariante sea mínima. Este último método es capaz de encontrar la pendiente adecuada a partir de una única imagen. Sin embargo, requiere que las imágenes utilizadas tengan áreas donde existan sombras notables. En el caso de que el cambio de iluminación sea global y no haya sombras, este método no es capaz de encontrar la pendiente correcta.

III. Añadiendo el Espacio Invariante de Iluminación a un AAM

En el modelo que se ha llevado a cabo para la aplicación, se ha realizado un modelo de apariencia activa, en el que se ha incluido el espacio invariante de iluminación como en [3] y además, del mismo modo que en [5], se han añadido parámetros que modelan la ganancia y el offset de la iluminación para compensar los efectos debidos a las no linealidades de las cámaras de bajo coste. Con todo esto se ha obtenido el siguiente error que se pretende minimizar:

$$\mathcal{E}(\tau) = \sum_x \left[\mathcal{L}'(A(x;\lambda), \theta_1) - \mathcal{L}'(I(W(x;p)), \theta_2)\right]^2 \tag{11}$$

donde $A(x;\lambda)$ está definida en (4), $I(\hat{x})$ es la imagen captada por la cámaray $W(x;p)$ es la función Warp de (3). $\mathcal{L}'(\rho, \theta)$ es una pequeña modificación de (10):

$$\mathcal{L}'(\rho, \theta) = \log\left(\frac{\rho_1 + b_1}{\rho_3 + b_3}\right)\cos(\theta) + \log\left(\frac{\rho_2 + b_2}{\rho_3 + b_3}\right)\sin(\theta) + d\mathcal{L} \tag{12}$$

en la que se han incluido los parámetros $b = [b_1, b_2, b_3]$, que modelan el posible offset producido en los canales de color. El parámetro $d\mathcal{L}$ trata de reproducir las diferencias entre la relación de ganancia de los distintos canales de color que suelen aparecer debido a las no linealidades de la cámara.

Por lo tanto, la ecuación de error (11) dependerá finalmente del conjunto de parámetros $\tau = [p, \lambda, \theta, b, d\mathcal{L}]$. El uso de los parámetros $\theta = [\theta_1, \theta_2]$ está motivado por la posibilidad de utilizar cámaras distintas en el entrenamiento que en el uso cotidiano de la aplicación.

IV. Aplicación

En esta sección se presentará una aplicación informática que, basada en los conceptos anteriormente abordados, pretende manejar un puntero, a partir de gestos faciales. Para ello, esta sección se divide en dos partes bien diferenciadas dentro del desarrollo: Entrenamiento y Programa-Interfaz.

Los Modelos de Apariencia Activa, explicados anteriormente, se ajustan bien a los requerimientos de esta aplicación debido a que por una parte, estos modelos se han demostrado lo suficientemente precisos. Por otra parte, debido a que modelan la apariencia, son capaces de hacer distinción entre varias personas, además de, al ser un modelo generativo, pueden reproducir gestos y caras que no han sido entrenadas.

IV-A. Entrenamiento

La aplicación destinada al entrenamiento tiene como fin el crear la información necesaria para que el Programa-Interfaz pueda funcionar correctamente. Por lo tanto este proceso debe ser ejecutado con anterioridad al Programa-Interfaz.

Entre los datos necesarios para que la aplicación del Programa-Interfaz pueda ejecutarse están las bases de forma y apariencia, definidas para los AAM. La creación de estas base, parte del marcado previo de unas imágenes de entrenamiento, y se basa fundamentalmente en el análisis de componentes principales.

El proceso comienza con una etapa previa en la que el usuario final u otra persona etiqueta en cada una de las imágenes de entrenamiento la posición correcta de los puntos de la malla mostrada en la figura 1. A partir de aquí, esta información es normalizada y analizada mediante PCA. Como resultado se obtiene tanto para la forma como para la apariencia, la media de las imágenes entrenadas y un conjunto de vectores que conforman la base.

Esta base obtenida, posee una dimensionalidad menor que el conjunto de datos del entrenamiento. Esto es consecuencia

directa de el algoritmo PCA que 'filtra' aquellas componentes que representan las variaciones menos características.

Además de las bases de forma u apariencia el entrenamiento debería generar cualquier otra información útil que pueda ser precalculada, como por ejemplo tablas de búsquedas o elementos precalculables del algoritmo de optimización.

IV-B. Programa-Interfaz

El Programa-Interfaz se refiere a todo aquello que hace posible la comunicación entre el usuario y el ordenador en *tiempo real*[1].

En la aplicación que se presenta en el presente trabajo, existen tres módulos independientes bastante bien diferenciados que se reflejan en la figura 2:

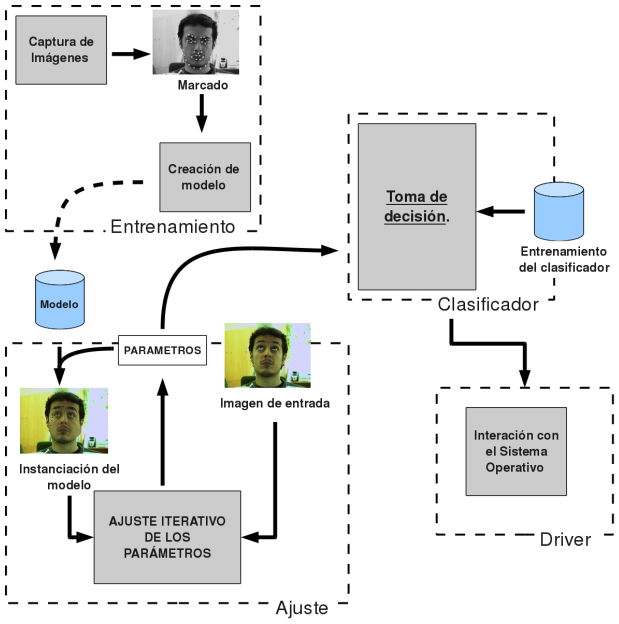


Figura 2. División e interacción entre los módulos de la aplicación.

Ajuste: Esta parte del Programa-Interfaz trata de encontrar los parámetros τ que minimizan la función de error $\mathcal{E}(\tau)$. Para ello, y ya que se desconoce la forma de la superficie que forma la función de error, se ha implementado un algoritmo de optimización iterativo ampliamente utilizado en visión artificial: el algoritmo de Lucas-Kanade [2], al cual se le ha añadido como modificación la aproximación de Levenberg-Marquardt [7] para mejorar la convergencia.

Este algoritmo de minimización local basado en el algoritmo de Gauss-Newton, tiene como ventajas una rápida convergencia y una elevada precisión. Sin embargo, a la posibilidad de converger en un mínimo local, inherente a los métodos locales, se le suma como inconvenientes a este algoritmo que resulta computacionalmente costoso debido a que en cada iteración se deben recalcular las matrices del Jacobiano y el Hessiano de la función de error.

[1]Es condición imprescindible para que la aplicación sea en tiempo real que el tiempo de procesado de la imágen y respuesta, sea menor que el tiempo de adquisición entre imágenes.

Para evitar converger en un mínimo local que resulte en una parametrización errónea de la cara, se deben de asegurar unas condiciones de inicio lo sufientemente cercanas al resultado final deseado. Para ello, se han realizado las siguientes modificaciones en el modelo explicado hasta ahora:

- Se ha añadido a la función de Warp hasta ahora explicada, una transformación global $N(x';q)$ que, mediante los parámetros $q = [q_1, q_2, q_3, q_4]$ modelará los cambios de tamaño, rotaciones y translaciones de la cara en la imagen. Esto es idéntico a definir una nueva función de warp $W'(x;p,q)$ resultado de componer la función de warp que se tenía y la transformación global explicada:

$$W'(x;p,q) = N(W(x;p);q) \tag{13}$$

  La nueva función de warp se puede sustituir en el modelo sin ningún cambio salvo que se deben añadir los nuevos parámetros q al conjunto de parametros τ de los que depende el error.
- Se inicializarán los nuevos parámetros globales q, hallando el tamaño, posición y orientación de la cara en la imagen usando el algoritmo propuesto por [8]. Se ha comprobado empíricamente que esta inicialización es suficientemente buena para la mayoría de los casos.

Además, también con el fin de tratar de evitar el converger sobre un mínimo local erróneo, y bajo la premisa que el tiempo entre las distintas capturas es lo suficientemente pequeño para asegura que una imagen y la siguiente son lo suficientemente parecidas, las sucesivas imágenes serán inicializadas con la parametrización del estado anterior.

Clasificador: La función del clasificador consiste en, a partir de los parámetros obtenidos mediante el ajuste, identificar el gesto o la posición de la cara con una etiqueta, como por ejemplo,"boca abierta", "guiño",etc.

En el caso de la aplicación presentada, esta clasificación se ha realizado mediante un sencillo clasificador basado en la distancia euclídea de los gestos en las bases del modelo. No obstante, si lo que se pretende es distinguir un mayor número de gestos o hacer una clasificación más eficaz de estos, en [9] se explica la realización de clasificadores a partir de una determinada parametrización.

El clasificador necesita de cierta información previa sobre el modelo, que debe obtenerse mediante el proceso de entrenamiento .

Driver

Por último, el denominado driver, se encargará de traducir las mencionadas etiquetas referidas a gestos y/o posiciones de la cara del usuario a una acción específica proporcionada por el Sistema Operativo. Como, por ejemplo mover en una determinada dirección un puntero.

V. PRUEBAS Y RESULTADOS

V-A. Descripción del experimento

En el experimento realizado se pretende comparar el algoritmo propuesto, que utiliza el espacio invariante junto a un

conjunto de parámetros que modelan la ganancia y el offset, con el algoritmo “Forward Additive”en RGB explicado en [2] y con el algoritmo que usa el espacio invariante sin los parámetros de ganancia y offset propuesto por [3].

Se ha partido de las imágenes mostradas en figura 3 en las que aparece la cara del mismo individuo en una escena en la que las condiciones de iluminación cambia de una imagen a otra. Como se puede apreciar, las diferencias en iluminación son notables.

(a) Imagen original con la que se ha realizado el entrenamiento (b) Imagen con cambios en la iluminación

Figura 3. Imágenes utilizadas en las pruebas

La totalidad del entrenamiento se realizó, en todos los casos, con las mismas imágenes: Un conjunto de 36 imágenes, obtenidas de una secuencia realizada con la iluminación de la figura 3(a).

En la figura 4 se muestra, como ejemplo, cuatro de las imágenes utilizadas en el entrenamiento en el que se incluye la figura 3(a) utilizada también en la prueba.

(a) (b) (c) (d)

Figura 4. Imágenes utilizadas en el entrenamiento

Para ofrecer igual trato a todos los algoritmos, la inicialización de la forma es la misma para todos: partiendo de un marcado “manual” se ha introducido un ruido gaussiano con cierta varianza. El resultado de la suma del marcado original y el ruido aleatorio ha sido almacenado y utilizado para ambas imágenes.

V-B. Resultados

Los siguientes resultados muestran el comportamiento de los algoritmos ante las imágenes mostradas en la figura 3 partiendo de una perturbación gaussiana, sobre el marcado, de desviación típica 5, 10 y 15 respectivamente. En el estudio se contempló los siguientes algoritmos.

- *RGB_AAM*: Algoritmo original propuesto por [1] sobre RGB.
- *LI_AAM*: Algoritmo utilizado en [3], usando únicamente el espacio invariante de iluminación.
- *LI_GB_AAM*: Algoritmo propuesto en el que se añade al espacio invariante parámetros para regular el offset y la ganancia de cada canal RGB.

En la figura 5(a), en la que la iluminación es igual a la utilizada en el entrenamiento, el algoritmo propuesto obtiene unos resultados finales similares al algoritmo sobre RGB y algo mejores que el algoritmo propuesto por [3]. No obstante nuestro algoritmo se muestra mas lento en la convergencia que los otros.

Es en la figura 5(b), donde se aprecian las ventajas de el algoritmo propuesto en este trabajo sobre los demás. Se puede contemplar claramente como el algoritmo que se propone es el que más reduce el error de malla cuando la imagen de entrada tiene cambios severos en la iluminación.

Por último, se puede apreciar en la figura 6 el comportamiento de los distintos algoritmos ante seis escenas reales con distinta iluminación. En la primera fila se muestra las seis imágenes de entrada. Posteriormente, en la segunda fila, se aprecia el resultado del algoritmo original en RGB donde se ve claramente como los puntos, que representan los vértices de la malla, no han convergido correctamente en la mayoría de la imágenes. Finalmente, en la última fila, se muestra el resultado de la convergencia de nuestro algoritmo, en el que los vértices de la malla han convergido correctamente para todas las imágenes.

V-C. Resultado Aplicación

Para finalizar, en la figura 7 se muestra como un usuario es capaz de mover un puntero usando la aplicación que se propone.

En la secuencia mostrada, el usuario mueve el puntero hacia los lados mediante el movimiento de su cabeza y, posteriormente, hace ‘click’ abriendo la boca (el puntero cambia de color).

Figura 7. Resultado del uso del algoritmo propuesto en una aplicación.

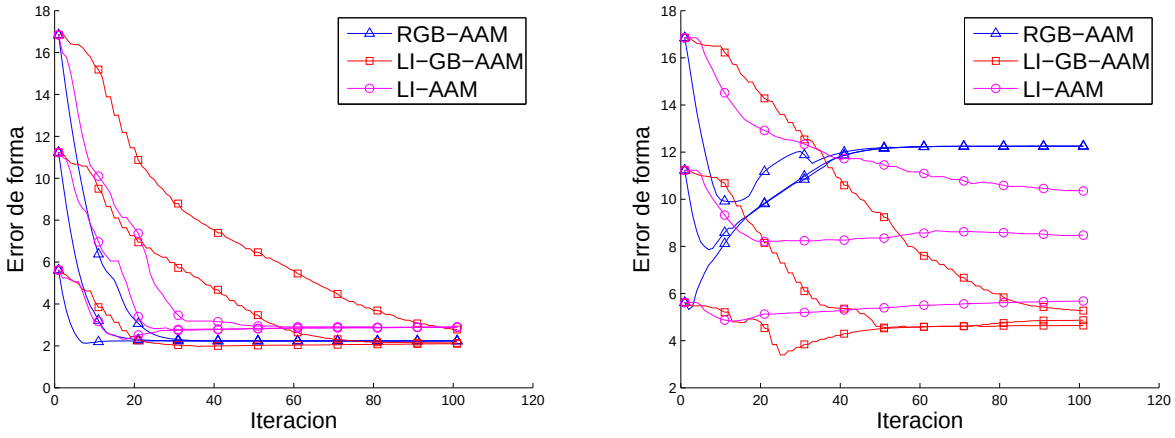
VI. Conclusiones

En este trabajo se ha abordado la realización de un interfaz hombre-máquina por medio de la visión por computador. Para ello se ha contemplado el uso de los gestos faciales, debido a que estos han sido escasamente utilizados en la mayoría de los interfaces humano-máquina realizados hasta ahora.

El uso de los Modelos de Apariencia Activa (AAM) aportan ventajas significativas, como son la precisión de este método o la posibilidad de identificar al usuario. Sin embargo, los AAM presentan otras deficiencias, como es la vulnerabilidad antes sombras y cambios de iluminación no pre-entrenados.

La novedad del método presentado es el uso de un modelo activo de apariencia junto a un espacio invariante a la iluminación, al cual además se le han añadido ciertos parámetros que modelan la ganancia y el offset de cada canal.

Este modelo, se ha mostrado superior a los otros algoritmos en cuanto a estabilidad ante cambios de iluminación aunque



(a) Resultado con iluminación original

(b) Resultados con cambios en la iluminación

Figura 5. Resultados obtenidos

Figura 6. Resultado del algoritmo de ajuste en imágenes con distinta iluminación y usando diferentes algoritmos.

algo más lento en su convergencia. Además, los parámetros añadidos permiten una mejor adaptación ante el posible cambio entre la cámara de entrenamiento y la utilizada en la aplicación.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente respaldado por el Ministerio de Ciencia e Innovación Español bajo los proyectos VISNU (ref. TIN2009-08984) y SDTEAM-UAH (ref. TIN2008-06856-C05-05) y por la Comunidad de Madrid y la Universidad de Alcala bajo el proyecto FUVA (CCG10-UAH/TIC-5988).

REFERENCIAS

[1] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *Computer Vision-ECCV'98*, p. 484, 1998.

[2] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[3] D. Pizarro, J. Peyras, and A. Bartoli, "Light-invariant fitting of active appearance models," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–6.

[4] G. Finlayson, S. Hordley, C. Lu, and M. Drew, "On the removal of shadows from images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 59–68, 2006.

[5] D. Pizarro and A. Bartoli, "Shadow resistant direct image registration," *Image Analysis*, pp. 928–937, 2007.

[6] G. Finlayson, M. Drew, and C. Lu, "Intrinsic images by entropy minimization," *Computer Vision-ECCV 2004*, pp. 582–595, 2004.

[7] J. More, "The Levenberg-Marquardt algorithm: implementation and theory," *Numerical analysis*, pp. 105–116, 1978.

[8] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[9] C. Bishop, *Pattern recognition and machine learning*, ser. Information science and statistics. Springer, 2006.