

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA DE TELECOMUNICACIÓN

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

Proyecto Fin de Carrera

Alumno: Jose Francisco Velasco Cerpa

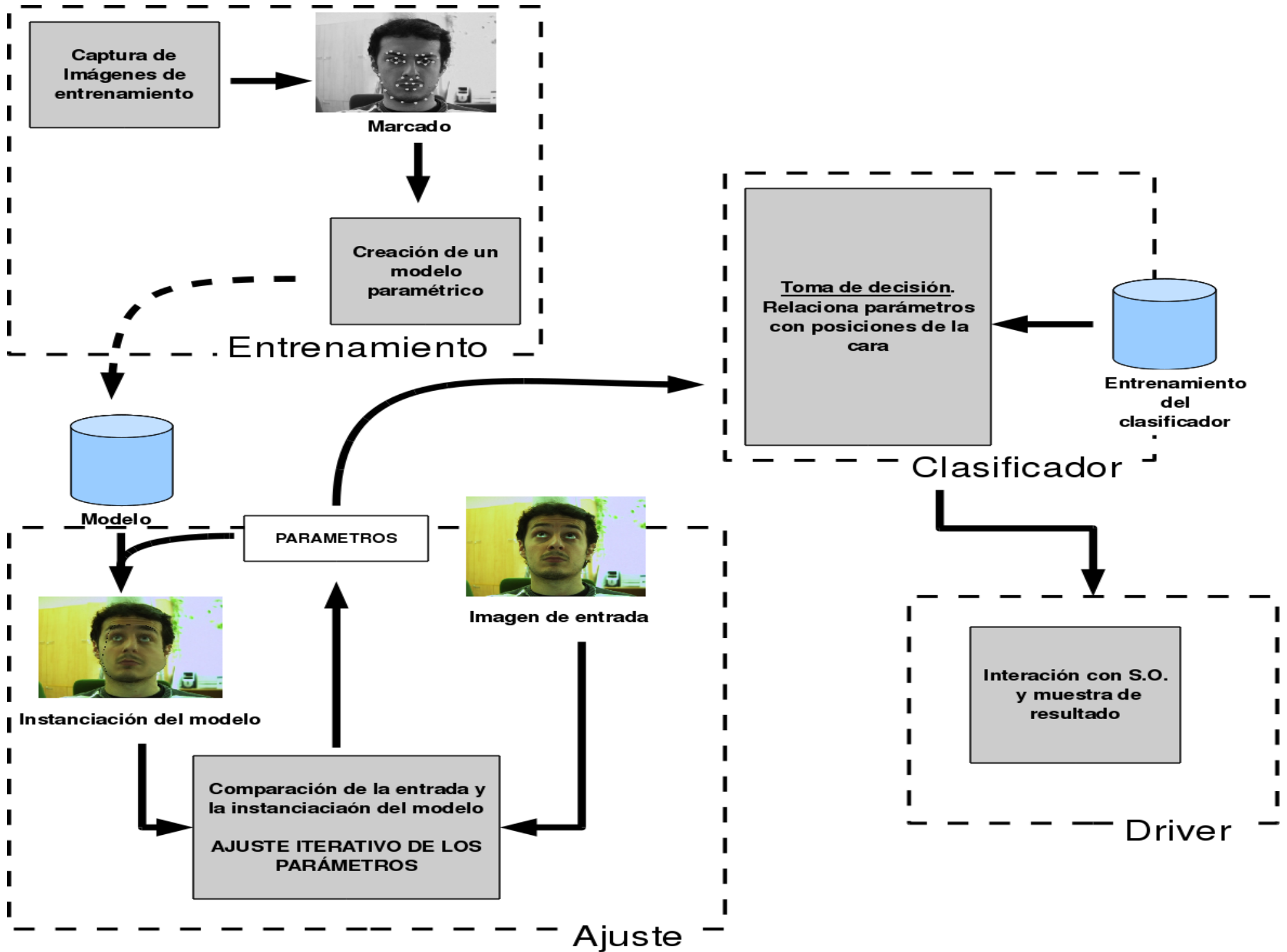
Tutor: Daniel Pizarro Pérez

Introducción

- Proliferación de entornos gráficos interactivos: Necesidad de emplear comandos complejos para su utilización.
- Hay personas que no pueden utilizar los dispositivos tradicionales
- Grandes avances en visión artificial en los últimos tiempos permiten procesar imágenes cada vez más complejas.
- “La cara es el reflejo del alma”: Hay mucha información en nuestra cara (sentimientos, reconocimiento de personas, diagnóstico prematuro...)

Objetivos

- Objetivo: Utilizar Visión por Computador para mejorar la interacción humano-máquina.
 - 1.- *Modelado paramétrico del rostro.*
 - 2.- *Implementación de un algoritmo de ajuste.*
 - 3.- *Inicialización automática.*
 - 4.- *Método de clasificación de gestos.*
 - 5.- *Implementación práctica.*
- Solución de bajo coste: Aplicación Software (Hardware estandarizado IEEE1394)



Elección del Algoritmo

- Diversas alternativas: Active Shape Models, Active Appearance Models, Active Blobs, Morphable Models, etc...
- Elección: Active Appearance Models
- Ventajas:
 - Preciso, Permite identificar a la persona, rápido...
- Inconvenientes:
 - Necesario entrenamiento, poco robusto ante cambios de iluminación, sombras, etc.

MODELOS ACTIVOS DE APARIENCIA

ACTIVE APPEARENCE MODELS
(AAM's)

Modelos Activos de Apariencia: Introducción

- Propuesto inicialmente en **1998** por Cootes.
- Basado en modelos generadores (Posibilidad de sintetizar caras) paramétricos y no lineales.
- **Necesario proceso de entrenamiento**
- Parámetros de los AAM's:
 - Forma.
 - Apariencia.
- Dos tipos de AAM's: Independientes y combinados

AAM's Independientes

Tratamiento por separado de forma y apariencia

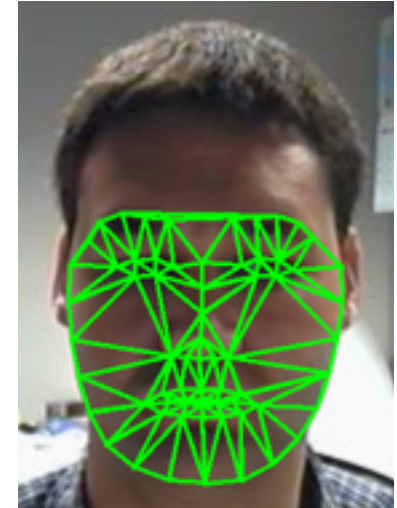
- **Forma:**

- Definida por los vértices de una malla

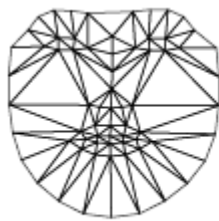
$$s = (x_0, y_0, x_1, y_1, \dots, x_v, y_v)^T$$

- Variaciones lineales

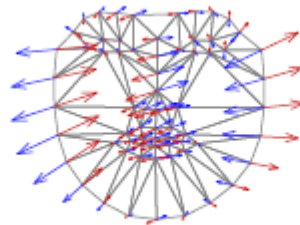
$$s = s_0 + \sum_{i=1}^n p_i \cdot s_i$$



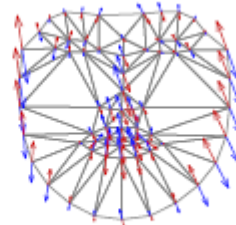
(s_0 : base de forma; s_i : vectores de forma; p_i : parámetros de forma)



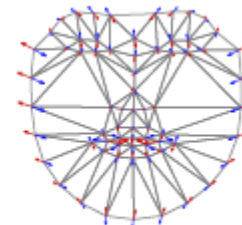
s_0



s_1



s_2



s_3

AAM's Independientes

Apariencia:

- Imagen $A(x)$ definida dentro de la malla base \mathbf{s}_0
- Variación lineal: $A(x) = A_0(x) + \sum_{i=1}^m \lambda_i \cdot A_i(x) \quad \forall x \in S_0$

(\mathbf{A}_0 : base de apariencia; \mathbf{A}_i : vectores de apariencia;

λ_i : parámetros de apariencia)



$A_0(\mathbf{x})$



$A_1(\mathbf{x})$



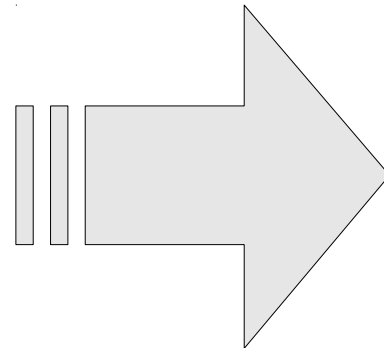
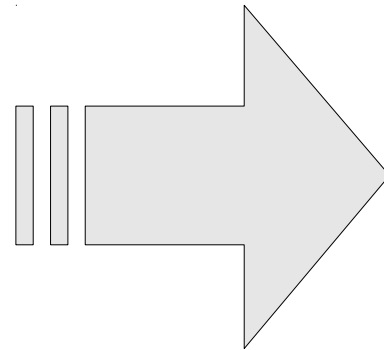
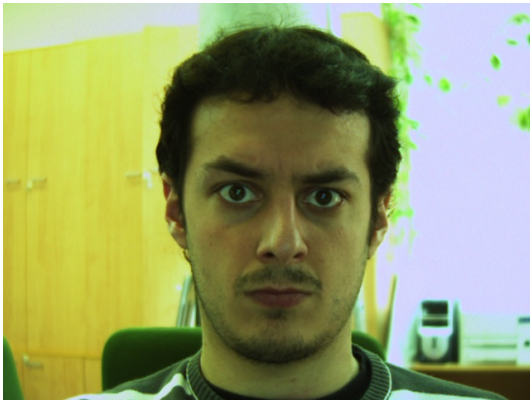
$A_2(\mathbf{x})$



$A_3(\mathbf{x})$

Relación entre forma y Apariencia: Función de Warp

- $W(x;p)$: Transformación afín definida a trozos.
- “Transporta” los pixeles desde la base de apariencia s_0 , hasta la malla del modelo s (o viceversa)

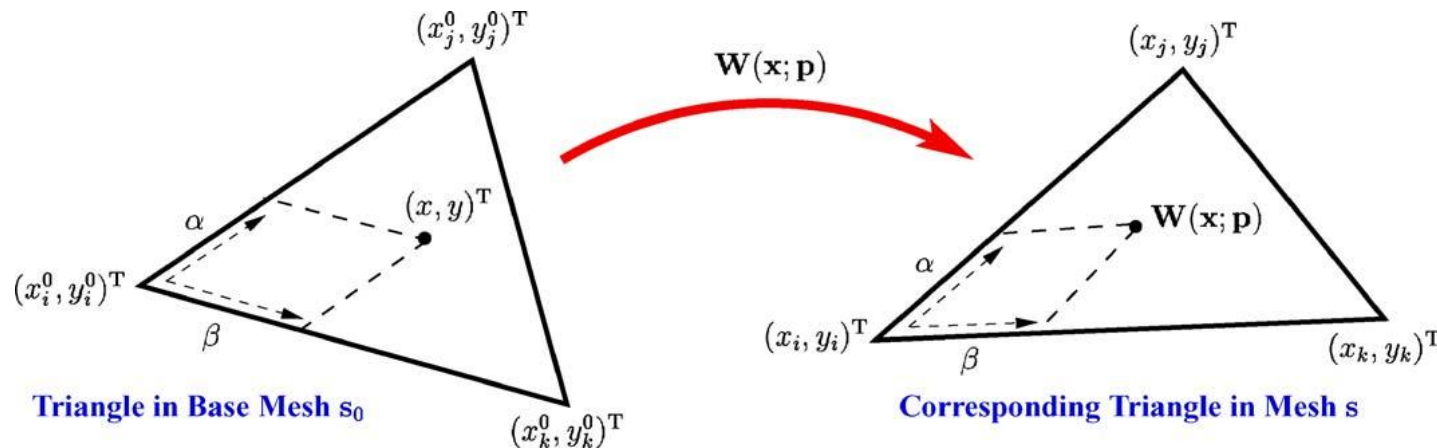


Función de Warp: Implementación

$$\mathbf{x} = (x, y) = (x_i^0, y_i^0) + \alpha \left[(x_j^0 + y_j^0) - (x_i^0 + y_i^0) \right] + \beta \left[(x_k^0 + y_k^0) - (x_i^0 + y_i^0) \right]$$

$$\alpha = \frac{(x - x_i^0)(y_k^0 - y_i^0) - (y - y_i^0)(x_k^0 - x_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (y_j^0 - y_i^0)(x_k^0 - x_i^0)}$$

$$\beta = \frac{(y - y_i^0)(x_j^0 - x_i^0) - (x - x_i^0)(y_j^0 - y_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (y_j^0 - y_i^0)(x_k^0 - x_i^0)}$$

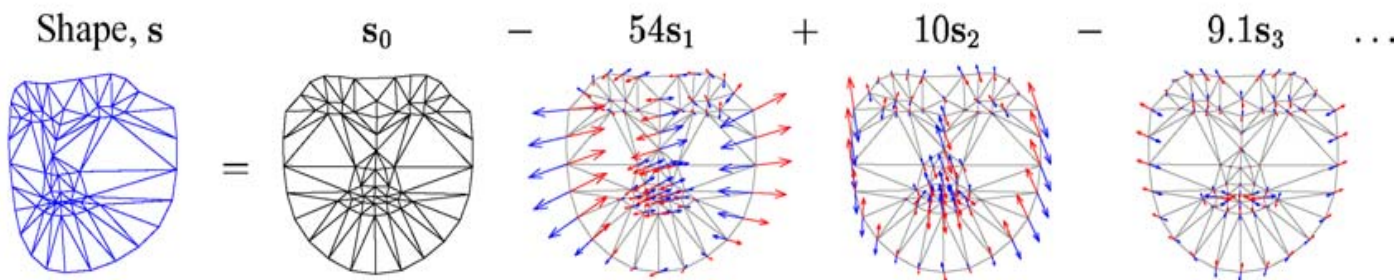
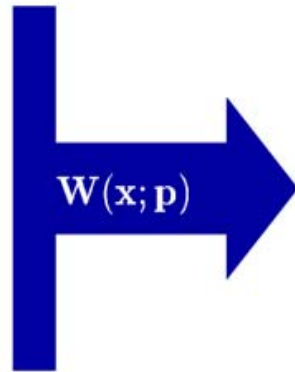


$$W(x; p) = (x_i, y_i) + \alpha \left[(x_j + y_j) - (x_i + y_i) \right] + \beta \left[(x_k + y_k) - (x_i + y_i) \right]$$

$$W(x; p) = (a_1 + a_2 \cdot x + a_3 \cdot y, a_4 + a_5 \cdot x + a_6 \cdot y)$$

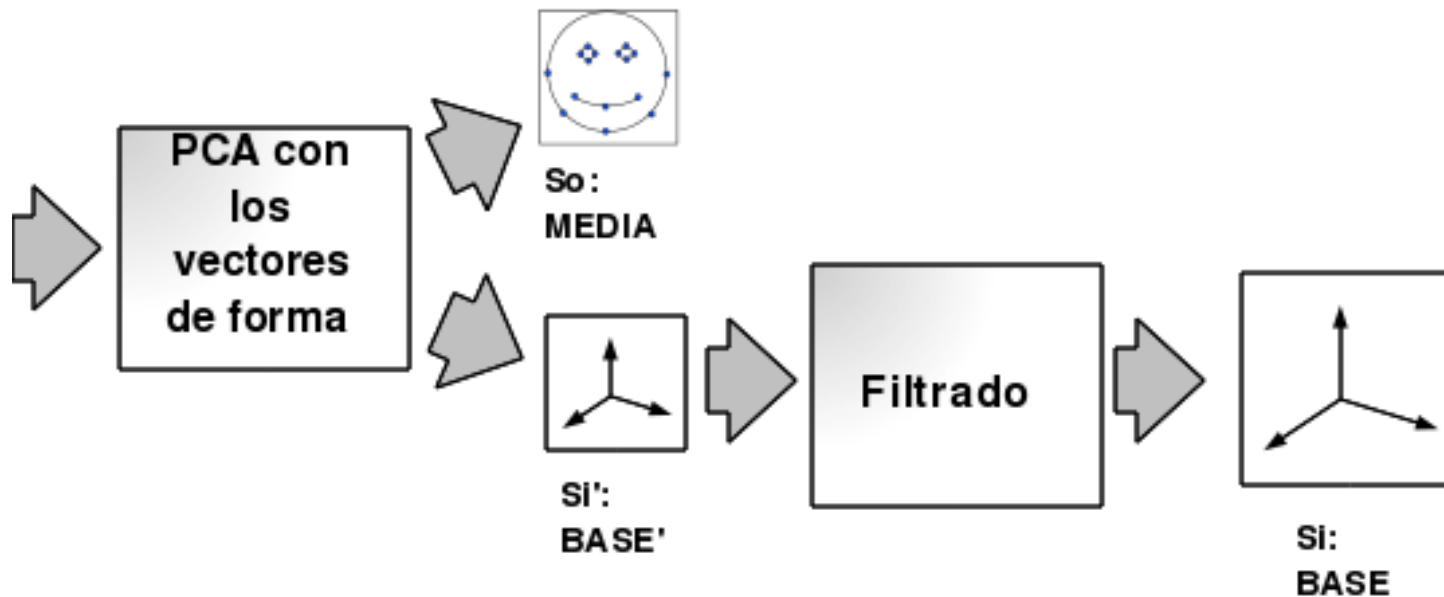
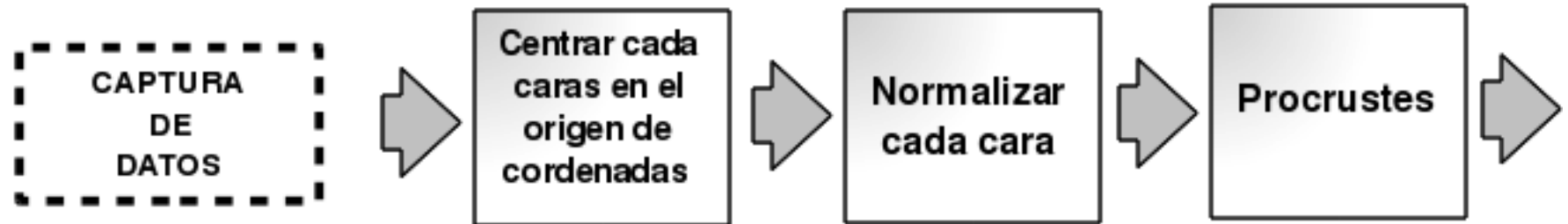
AAM's Independientes: Instanciación del modelo

- Dada una particularización de \mathbf{p} y $\boldsymbol{\lambda}$ ¿Como creamos la instancia del modelo, $M(\mathbf{x})$?



Entrenamiento(I)

- **ENTRENAMIENTO DE FORMA:**



Entrenamiento(II)

- **ENTRENAMIENTO DE APARIENCIA**

1) Transformación
(mediante función warp)
de todas las imágenes a una
forma comun (s_0)

2) *Análisis de componentes
principales de la apariencia
(Se crea la base de apariencia)*

3) *Filtrado por energía*



Aplicación con AAM's

- **'FITTING'**

- Objetivo: Encontrar λ y p que hagan que el modelo se ajuste a la imagen de entrada.

- La solución debe minimizar la siguiente función de coste:

$$\sum_{x \in s_0} E(x; p, \lambda)^2 = \sum_{x \in s_0} \left[A_0(x) + \sum_{i=1}^m \lambda_i \cdot A_i(x) - I(W(x; p)) \right]^2$$

- Inicialización con Viola&Jones.
- Algoritmos empleados:
 - Lucas-Kanade
 - Composicional Inverso

Algoritmo Lucas-Kanade

- Características

- Método Iterativo. Suponiendo $(\Delta p, \Delta \lambda) \rightarrow 0$

- Incremento Aditivo y sobre $I(x)$:

$$\sum_{x \in s_0} E(x; p + \Delta p, \lambda + \Delta \lambda)^2 = \sum_{x \in s_0} [A_0(x) + \sum_{i=1}^m (\lambda_i + \Delta \lambda_i) \cdot A_i(x) - I(W(w; p + \Delta p))]^2$$

$$(\Delta p, \Delta \lambda) = -H^{-1}(p, \lambda) \cdot \sum_{x \in s_0} [SD^T(p, \lambda) \cdot E(x)]$$

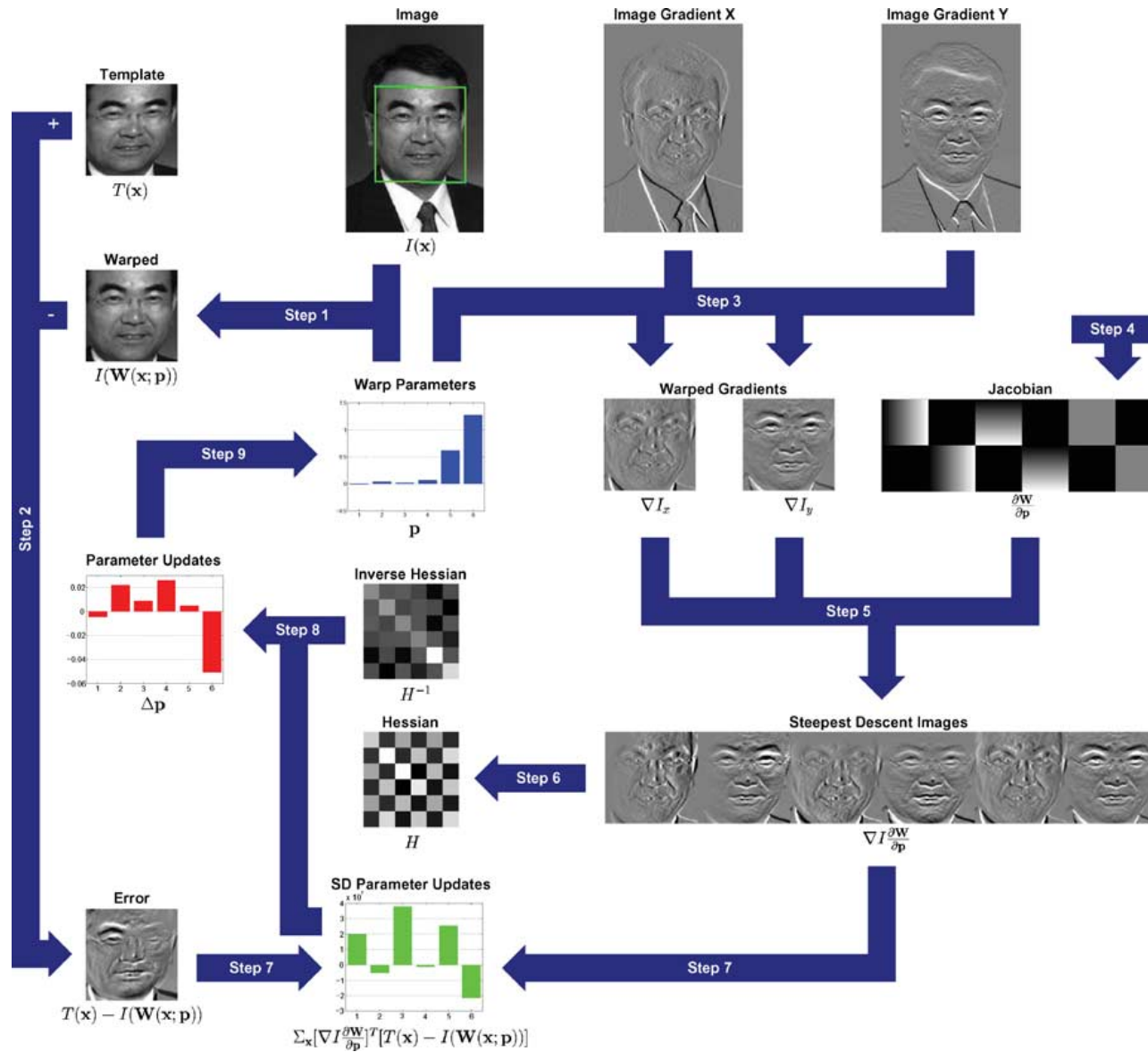
- Ventajas:

- Preciso

- Inconvenientes:

- Es necesario recalcular las matrices H y SD:
(No se cumplen requisitos de tiempo real)

Algoritmo Lucas-Kanade.



Algoritmo Composicional Inverso.

- Trata de mejorar el rendimiento de Lucas-Kanade
- Características
 - Nueva forma de Actualizar los parámetros de forma

$$W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}$$

- Aproximación: Se asume que el error en el espacio ortogonal solo depende de los parámetros de forma.

$$\sum_{x \in s_0} E(x; p + \Delta p, \lambda + \Delta \lambda)^2 = \sum_{x \in s_0} [A(W(x, \Delta p)) - I(W(x; p))]^2$$

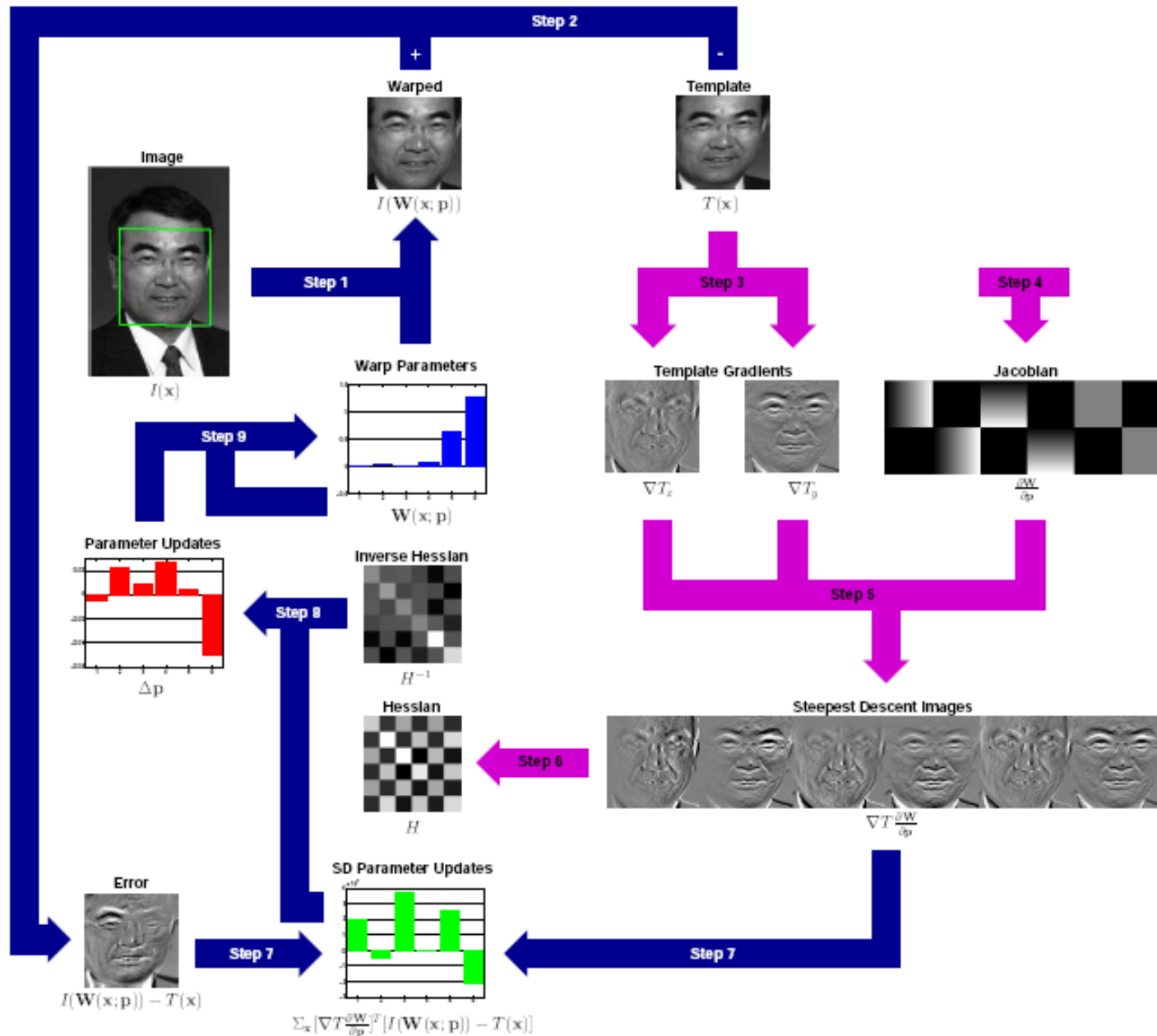
$$\Delta p = -H^{-1}(p=0) \cdot \sum_{x \in s_0} [SD_{po}^T(p=0) \cdot E(x)]$$

$$\lambda_i = \sum_{x \in s_0} A_i(x) \cdot [A_0(x) - I(W(x; p))]$$

Algoritmo Composicional Inverso.

- Ventajas
 - Como H y SD son evaluadas siempre en $p=0$ se puede precalcular. Ahorrando tiempo en cada iteración.
- Inconvenientes
 - Se pierde precisión.
 - No se conocen parámetros de apariencia hasta que se termina de iterar.
 - Peor convergencia

Algoritmo Composicional Inverso.



Resultados experimentales

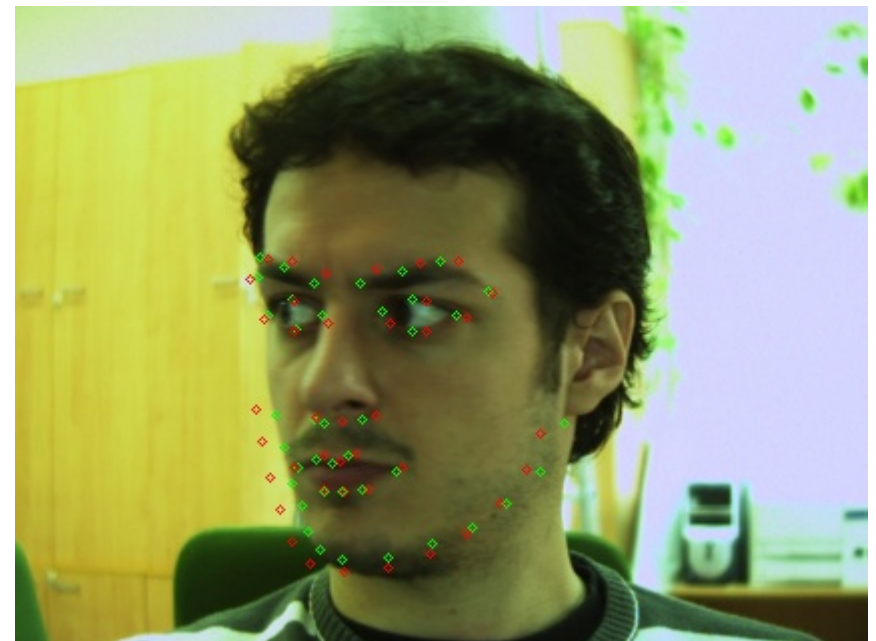
- **Error de Ajuste**

	Converge		No converge	
	FA	CI	FA	CI
Entrendas	34	28	2	8
No Entrendas	13	10	1	4
TOTAL	47	38	3	12

Lucas Kanade



Composicional Inverso



Resultados experimentales

- Respuesta al ruido
 - Experimento realizado con varianzas iniciales de 0, 4, 8, 12, 16 y 20 píxeles al cuadrado.
- Análisis de tiempo

	Fps
Lucas-Kanade	0,5
Composicional Inverso	15

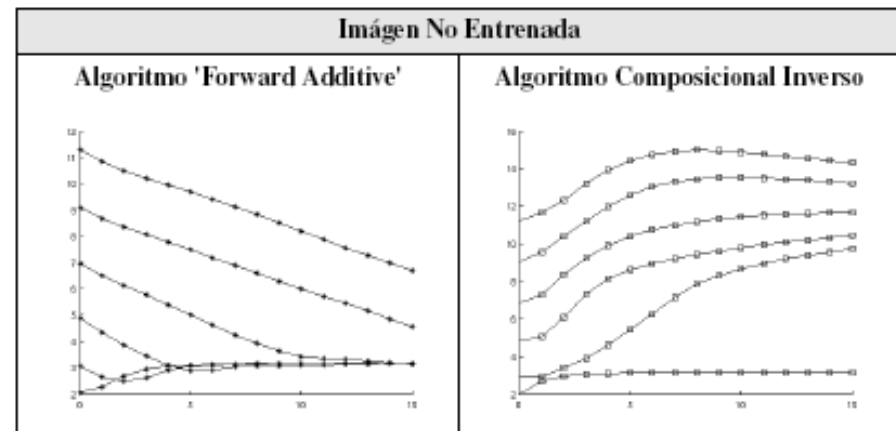
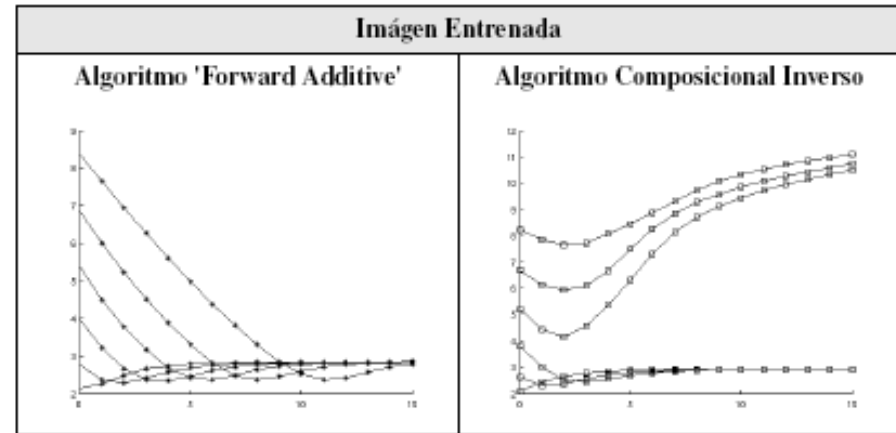


Ilustración 35: Variación del error de malla en función del número de iteraciones para Imágenes entrenadas y no entrenadas, usando distinto tipos de algoritmo y de varianzas en el error inicial

Aplicación con AAM's

- **Reconocimiento de gestos**
 - Relacionar parámetros λ y p con el gesto realizado.
 - Realizado de forma sencilla mediante distancia euclídea en el espacio vectorial de los parámetros de forma. (Cada componente se pondera por el autovalor que tuvo asociado en el entrenamiento PCA)

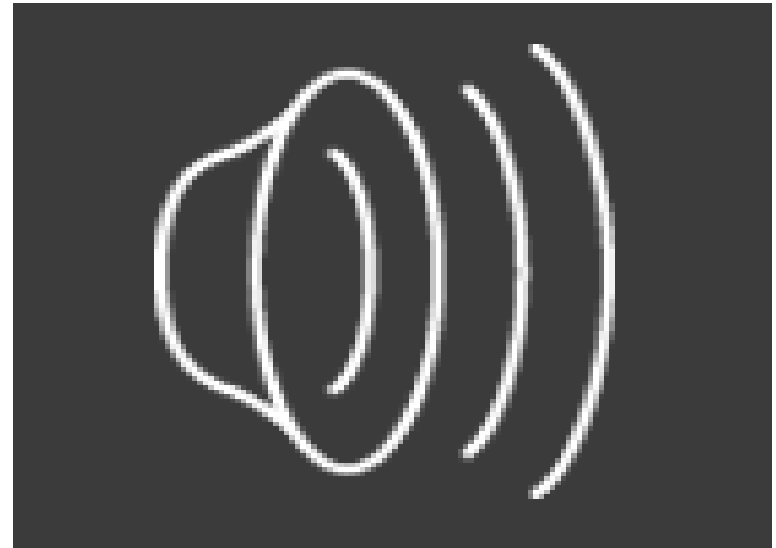
Condiciones de nuestra aplicación

- Precisión.
- Tiempo Real.
- Versatilidad: Capaz de incorporar nuevos gestos y/o adaptarse a las características de cada usuario.
- Económico: Uso de dispositivos de entrada estándar (idealmente una web-cam).

Aplicación

- Interfaz hombre-máquina basado en el análisis de gestos faciales mediante visión artificial.
- Orientado hacia la ayuda a la discapacidad.
- Tratamos de que un usuario sea capaz de dar diversas ordenes como mover un puntero, hacer 'click' o incluso otras más complejas utilizando únicamente su cara
- Necesaria condición de tiempo real.

Demostración



Conclusiones

- Es posible conseguir tiempo real (Resolución: 640x480)
- Los AAM's son poco robustos ante cambios de iluminación.
- Problemas ante oclusiones. (Estimadores estadísticos)
- Posible mejorar el entrenamiento (PCA Missing Data)