

Evaluación de desempeño del algoritmo de seguimiento de características faciales basado en modelos ASM usando Kinect¹

Performance evaluation of a face tracking algorithm based on ASM models using the Kinect Sensor.

Diego Alejandro Rodríguez Ardila², Jahir Joel García Mendoza³, Carlos Humberto Esparza Franco⁴, Edwin Alberto Silva Cruz⁵

¹Unidades Tecnológicas de Santander – UTS, Bucaramanga, Colombia.

Artículo recibido en mes **XX de año**; artículo aceptado en mes **XX de año**

Citación del artículo: Rodríguez, D. & García, J. & Esparza, C. & Silva, E. (2016). Evaluación de desempeño del algoritmo de seguimiento de características faciales basado en modelos ASM usando KINECT. *I+D Revista de Investigaciones*, 1(2), pp-pp.

¹ Artículo con enfoque cuantitativo resultado de un trabajo de grado en la modalidad de proyecto de investigación en el área de procesamiento digital de imágenes, desarrollado en el grupo de investigación en control avanzado – GICAV, y financiado por las Unidades Tecnológicas de Santander en la ciudad de Bucaramanga. Calle de los estudiantes No. 9-82. PBX:6917700. Fecha de inicio: enero de 2015, fecha de terminación: mayo de 2016.

² Ingeniero Electrónico, Unidades Tecnológicas de Santander. Estudiante de trabajo de grado en modalidad de proyecto de Investigación. Correo electrónico: diegorodriguez_978@hotmail.com

³ Ingeniero Electrónico, Unidades Tecnológicas de Santander. Estudiante de trabajo de grado en modalidad de proyecto de Investigación. Correo electrónico: jahirg@yahoo.com.

⁴ Ingeniero Electrónico, Universidad Industrial de Santander UIS. Candidato a Magister en Diseño, Gestión y Dirección de Proyectos, Centro Panamericano de Estudios Superiores CEPES. Docente- investigador del grupo: GICAV. Unidades Tecnológicas de Santander UTS de la ciudad de Bucaramanga (Colombia): Dirección Calle de los estudiantes No. 9-82, PBX: 6917700. Correo electrónico institucional: gicav@correo.uts.edu.co

⁵ Ingeniero Electrónico, Magíster en Ingeniería, Magíster en Procesamiento de imágenes, audio, señales y telecomunicaciones y PhD en Ingeniería Electrónica. Docente investigador en el grupo GPS de la Universitaria de Investigación y Desarrollo. esilva6@udi.edu.co

Resumen

En este trabajo se presenta la evaluación del desempeño de un algoritmo de seguimiento de las características faciales, aplicando modelos de forma activa (ASM) y utilizando el sensor Kinect como dispositivo de captura de imagen. El desarrollo se realizó mediante las librerías de OpenCV, en un PC portátil con Procesador Core i5 a 2.4Ghz, 4 Gigabytes de memoria RAM, que corre bajo sistema operativo Windows 7. Para la evaluación, se ejecutó el algoritmo para observar la respuesta respecto a las distintas posturas y expresiones faciales. Se tomaron los tiempos de estabilización de los puntos sobre la imagen y se analizó punto a punto y con criterio humano la localización de los puntos sobre la imagen. Para facilitar el análisis se agrupan los puntos acordes a la zona del rostro: Contorno de la cara, cejas, nariz, ojos y boca. Por último, se presentan los resultados del tiempo promedio de ajuste del modelo, el promedio de frames, así como un error promedio de posicionamiento en las distintas condiciones del rostro, lo cual muestra la robustez de este trabajo y la adaptabilidad para trabajos futuros.

Palabras clave: Kinect sensor, Modelos Activos de Forma ASM, PCA, OpenCV, Reconocimiento de puntos característicos.

Abstract.

This work shows the performance evaluation of a face tracking features algorithm, applying active shape models (ASM) and using the Kinect sensor as the capture image device. The development was implemented by using OpenCV libraries, in a laptop with processor core i5 at 2.4 GHz, 4 GB of RAM, and Windows 7 operative system. In order to perform the evaluation, the algorithm was run to analyze the response under different facial expression poses. The stabilization times of the points over the image were measured and the localization of the points in the image was manually evaluated. The face was divided in regions: face contours, eyebrows, nose, eyes and mouth. Finally, the results are presented as the average time of the face matching, the average number of frames required to perform matching, and the average error of the positioning in different faces conditions. The results show the strength of this work and the adaptability for future work.

Introducción

El ajuste de un modelo deformable es el gran problema de registrar un modelo de forma parametrizada en una imagen de manera que su marcas correspondan a las ubicaciones consistentes en el objeto de interés (Y. Cheng, 1995). Este es un problema difícil, ya que implica una optimización en las dimensiones de altura, donde la apariencia puede variar mucho entre instancias del objeto, debido a las condiciones de iluminación, ruido de la imagen, la resolución y las fuentes intrínsecas de variabilidad.

Se han propuesto muchos enfoques para esto con diversos grados de éxito (Y. Cheng, 1995; Cootes & Taylor, 1992; Cristinacce & Cootes, 2004, 2006, 2007). De ellos, uno de los más prometedores es el que utiliza una representación basada en parches (Baltrusaitis, Robinson, & Morency, 2012) en donde se tiene en cuenta las observaciones hechas en cada marcación de la imagen y su relación con las demás.

Este proyecto se desarrolló en una PC, empleando el Sensor Kinect como dispositivo de adquisición de imagen, sistema operativo Windows 7 de 32 Bits, el SDK Kinect for Windows, las librerías OpenCV, lenguaje de programación Python2.7 de 32 Bits, y paquetes básicos para Python, entre los que se destacan el PyKinect, Numpy, Scipy, Cython y PyQt. Como IDE se utiliza el programa Geany y como GUI el paquete de PythonPyQt. Todo dentro de un ambiente de Software de uso libre.

Las técnicas que se utilizaron para el seguimiento de las características faciales de forma activa son: el reconocimiento de rostros, el PCA(Principal Component Analysis) (Delac, Grgic, & Grgic, 2005), el PDM (Point Distribuid Model) (Chen & Davoine, 2006; Viola & Jones, 2001), CLM (Constrained Limited Model) (Baltrusaitis et al., 2012; Dornaika & Davoine, 2004) y el modelo de parches (Baltrusaitis et al., 2012; S. Cheng, Zafeiriou, Asthana, & Pantic, 2014; Cristinacce & Cootes, 2004).

El entrenamiento de dicho modelo se hizo con ayuda de la base de rostros MUTC desarrollada por la Universidad del Cabo, para uso investigativo. La cual contiene fotografías para cada individuo en diferentes condiciones de luminosidad y posición.

Mediante el paquete PyKinect y el SDK Kinect se tiene acceso al sensor, para realizar la toma de muestra. Esta se procesa mediante el OpenCV, y se analiza con la matriz de entrenamiento, si hay resultado satisfactorio tendremos vector de con las coordenadas de los puntos identificados, el cual es usado para graficar los resultados y permite hacer el cálculo de error.

Metodología utilizada

DetECCIÓN DE ROSTRO

Una vez tomada la matriz de la imagen el primer paso es la detección de rostro con lo cual se busca definir el ROI (*Region of Interest*) (Delac et al., 2005) y de esta forma continuar el análisis. Dado que es la búsqueda de un rostro y no la detección de un individuo en particular, OpenCV facilita esta tarea usando HaarCascades basadas en el algoritmo de Viola-Jones (Viola & Jones, 2001).

Restricción de los modelos locales

La mayoría de los métodos de ajuste emplean una aproximación lineal de cómo su marco de objeto no rígido se deforma, esto da origen el PDM (Modelo de distribución de puntos) (Cootes & Taylor, 1992). El PDM Modela las variaciones del marco no rígidos linealmente y la constituye con una transformación rígida global, ubicando el marco dentro de la imagen como se muestra en la ecuación 1.

$$x_i = sR(x_i + \varphi_i q) + t \quad (1)$$

Donde x_i son las coordenadas de las marcas de la distribución de puntos i y el conjunto de valores $\{s, R, \varphi, t\}$ son los parámetros del PDM, los cuales son s escala global, R rotación, t traslación, y φ un conjunto de parámetros no rígidos.

Modelos locales limitados (CLM)

Para evitar los inconvenientes de enfoques holísticos, tales como la complejidad del modelo y la sensibilidad a los cambios de iluminación, se usa un enfoque global que utiliza un conjunto de detectores (Y. Cheng, 1995; Cootes & Taylor, 1992; Cristinacce & Cootes, 2004, 2006, 2007; Wang, Lucey, & Cohn, 2008). Todas las instancias de CLMs buscan dos

I+D Revista de Investigaciones ISSN 22561676 Volumen XX Número XX Año 2016
Enero-Junio 2016 pp. xx-xx

objetivos: (i) realizar una búsqueda local exhaustiva para cada marca del PDM en torno a su estimación actual utilizando algún tipo de detector de función, y (ii) optimizar los parámetros PDM de tal manera que la respuesta de detección sobre la totalidad de sus marcas de referencia sea conjuntamente máxima. La figura 1, muestra el ajuste de los componentes CLM.

Intensa búsqueda local

En la primera etapa de ajuste CLM (Baltrusaitis et al., 2012), se genera un mapa de probabilidad para cada marca de posición aplicando detectores locales para restringir las regiones alrededor de la estimación actual. Se ha propuesto un número de detectores de característicos para este propósito. Uno de los más sencillos, propuesto por Wang, Lucey y Cohn en (Wang et al., 2008), es el regresor lineal logístico que da el siguiente mapa respuesta para el i -ésimo punto de referencia obtenido a través de la ecuación 2.

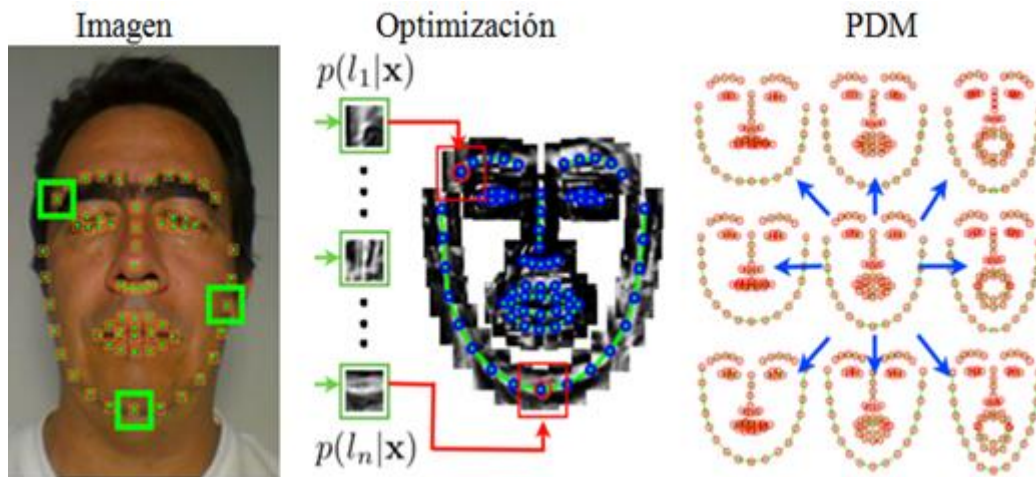


Figura 1. Ajuste de CLM y sus componentes: (i) intensa búsqueda local de la ubicación de los rasgos para obtener el mapeo $\{p(l_i = alineado | I, x)\}_{i=1}^n$ y (ii) una respuesta de optimización para maximizar la respuesta PDM del ajuste de las marcas.

$$p(l_i = alineado | I, x) = \frac{1}{1 + \exp\{\alpha C_i(I: x) + \beta\}} \quad (2)$$

Donde l_i es una variable aleatoria discreta que denota si la señal i -ésima está alineada correctamente o no. I es la imagen, x es una ubicación 2D en la imagen, y C_i es un clasificador lineal dado por la ecuación 3.

$$C_i(I: x) = w_i^T [I(y_1); \dots; I(y_m)] + b_i \quad (3)$$

Con $\{y_i\}_{i=1}^m \in \Omega_x$ (por ejemplo un parche de imagen). Una ventaja de usar este clasificador es que el mapa puede ser calculado usando eficientes operaciones de convolución.

Optimización

Una vez que se han encontrado los mapas de respuesta para cada lugar de interés, asumiendo la independencia condicional, se optimizan los procesos mediante la maximización, con respecto a los parámetros p del PDM, donde x_i es parametrizado como en la ecuación (1) y la dependencia de la imagen desciende para ser breve. Cabe señalar que algunas formas de CLMs plantean la ecuación (4) minimizando la suma de las respuestas locales.

$$p(\{l_i = \text{anileado}\}_{i=1}^n | P) = \prod_{i=1}^n p(l_i = \text{alineado} | x_i) \quad (4)$$

El método simple utilizado por Cristinacce y Cootes en (Cristinacce & Cootes, 2004) es un optimizador genérico, sin embargo, la convergencia puede ser lenta cuando se utiliza para un PDM complejo con un gran número de parámetros.

ASM

La estrategia de optimización simple adecuada para CLM es que la utilizada en el modelo de la forma activa (ASM) (Cootes & Taylor, 1992). El método implica primero la búsqueda de la ubicación dentro de cada mapa respuesta para el cual se alcanzó el máximo valor dado por $\mu = [\mu_1; \dots; \mu_n]$. El objetivo del procedimiento de optimización es entonces minimizar la diferencia ponderada de mínimos cuadrados entre el PDM y las coordenadas de las respuestas de los picos:

$$Q(p) = \sum_{i=1}^n w_i \|x_i - \mu_i\|^2 \quad (5)$$

Donde los pesos $\{w_i\}_{i=1}^n$ reflejan la confianza sobre coordenadas de respuesta pico y se establecen normalmente a alguna función de las respuestas en $\{\mu_i\}_{i=1}^n$ haciéndolo más resistentes a cosas tales como la oclusión parcial, donde puntos de referencia ocluidos se ponderan más débilmente.

La ecuación (5) se minimiza de forma iterativa mediante el primer orden de expansión de Taylor de las marcas del PDM tal como la ecuación 6.

$$x_i \approx x_i^c + J_i \Delta p \quad (6)$$

Resolviendo para la actualización de parámetros, siendo Δp la diferencia ponderada del mínimo cuadrado de las distancias:

$$\Delta p = \left(\sum_{i=1}^n w_i J_i^T J_i \right)^{-1} \sum_{i=1}^n w_i J_i^T (\mu_i - x_i^c) \quad (7)$$

Que luego se aplica de forma aditiva a los parámetros actuales: $p \leftarrow p + \Delta p$. Aquí, $J = [J_1; \dots; J_n]$ es el Jacobiano y $x^c = [x_1; \dots; x_n]$ es el actual marco estimado.

Desde el punto de vista probabilístico introducido en CLM, el procedimiento de optimización de la ASM es equivalente a la aproximación de los mapas de respuesta con un estimador isotrópico de Gauss.

$$p(l_i = \text{alineado} | x) \approx N(X; \mu_l, \sigma_l^2 I) \quad (8)$$

Donde $w_i = \sigma_i^2$. Con esta aproximación, tomando el logaritmo negativo de la probabilidad en la ecuación (4) da como resultado el objetivo en la ecuación (5). Es de aclarar que entre más complejo sea el marco y más cantidad de puntos tenga el costo computacional se eleva lo que se refleja en tiempos más grandes de procesamiento.

Resultados

En esta sección se describen los resultados de las pruebas realizadas para evaluar la eficiencia y el tiempo del algoritmo en el PC con sistema operativo Windows 7 32 bits con Procesador Core i5 @ 2.4Ghz, 4 Gigabytes de memoria RAM, y un disco de 256 GB.

Para realizar la implementación de las técnicas de reconocimiento hay dos cosas importantes el entrenamiento y el algoritmo.

Para realizar la detección se activa el sensor Kinect, se carga la imagen pasándola a escala de grises, y se aplica la detección de rostro por HAAR, lo cual determina el área de interés, ver figura 3. A dicha región se hace la búsqueda del centro de los ojos y de la boca para hacer el análisis de posición, escalado, rotación y traslación. Ya con esto, se aplica el vector de entrenamiento para establecer las coincidencias de las marcas, posteriormente se aplica la optimización ASM, y así corregir posibles desviaciones por las oclusiones, ver figura 2.

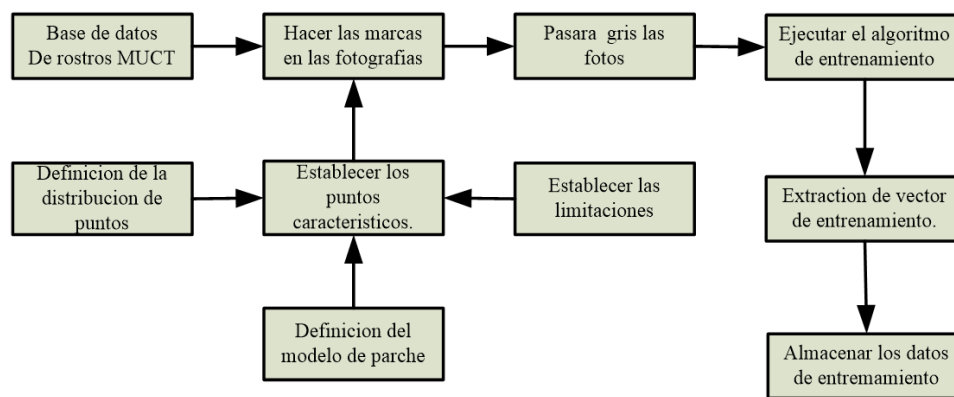


Figura 2. Mapa de las etapas del proceso de entrenamiento.

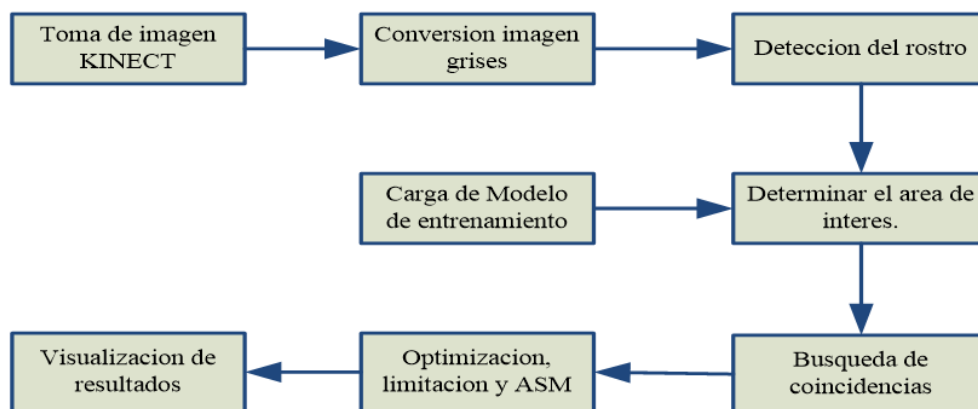


Figura 3. Mapa de las etapas del proceso de reconocimiento.

Dentro del aplicativo del proyecto es posible grabar video para su posterior análisis de error o simplemente visualizar su comportamiento en tiempo real, adicionalmente se tiene la opción de manipular el motor para ajustar la inclinación del sensor Kinect.



Figura 4. Etapas del proceso de reconocimiento.

Se realizaron pruebas para observar la respuesta del algoritmo respecto a las distintas posturas y expresiones faciales, ver figura 4. Para esto se hizo correr el algoritmo diez veces sobre una postura inmóvil, al final se tomó el tiempo que tomo estabilizar los puntos sobre la imagen y sobre la imagen final se analizó punto a punto y con criterio humano donde deberían estar localizados los puntos sobre la imagen.

Para facilitar el análisis se agrupan los puntos acordes a la zona del rostro: Contorno de la cara, cejas, nariz, ojos y boca, e igualmente se grafica el error punto a punto.

Ya con estos datos se calcula el error de ubicación. A continuación, se observan las distintas posturas y las tablas de resultados.

Postura cara frontal

En la figura 5, se muestra la marcación de los puntos sobre la postura cara frontal. Los resultados del cálculo del error de ubicación, presentados en la Tabla 1, muestran un seguimiento bastante real de la localización de los puntos, encontrándose los puntos más desviados en la parte inferior de la mandíbula.

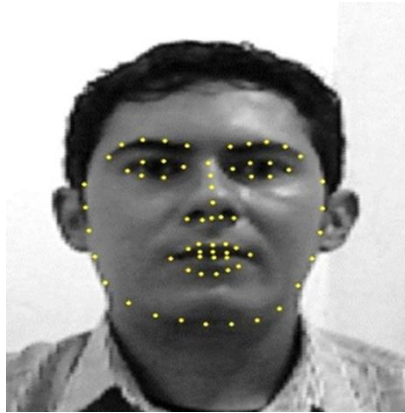


Figura 5. Puntos encontrados para una cara frontal, una vez estabilizado.

Tabla 1. *Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara frontal neutra*

Análisis de resultados cara frontal.							
Iteración	Tiempo seg.		Error RMS				
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.462	6.32	5.41	3.12	5.83	4.77	5.24
2	1.655	6.51	5.70	5.58	6.57	7.84	6.63
3	1.888	5.85	7.09	4.43	6.28	6.75	6.17
4	1.654	6.88	5.86	3.84	7.28	8.04	6.70
5	1.914	7.45	6.60	4.69	6.98	6.97	6.73
6	1.856	8.71	5.65	3.53	6.63	4.12	5.91
7	2.018	6.86	7.06	3.07	7.86	5.09	6.07
8	1.649	9.32	7.47	4.29	8.27	5.96	7.25
9	1.831	6.92	7.15	4.76	8.26	6.71	6.85
10	1.626	6.34	6.74	4.10	7.12	8.48	6.82
Promedio	1.755	7.12	6.47	4.14	7.11	6.47	6.43

Postura boca abierta

La figura 6, muestra la marcación de los puntos sobre el rostro frontal con boca abierta. Los resultados del cálculo del error de ubicación, presentados en la tabla 2, muestran un seguimiento bastante real de la localización de los puntos, notándose una desviación en general un poco mayor que al rostro frontal neutro. La boca abierta permite demarcar mejor los puntos.

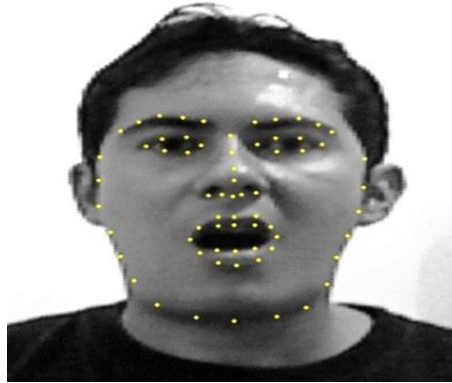


Figura 6. Puntos encontrados para una cara frontal con la boca abierta, una vez estabilizado

Tabla 2. Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara frontal con boca abierta

Análisis de resultados cara frontal boca abierta.							
Iteración	Tiempo seg.			Error RMS			
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.801	9.07	4.76	3.10	6.43	5.47	6.14
2	1.685	8.41	5.65	10.33	7.74	5.35	7.30
3	1.805	8.54	5.47	2.94	6.97	5.67	6.24
4	1.642	7.52	6.04	4.16	8.55	6.07	6.63
5	1.679	7.88	5.91	3.91	9.21	6.24	6.83
6	1.724	6.63	5.88	3.12	16.96	6.11	7.77
7	1.822	6.51	8.07	3.47	7.99	5.60	6.35
8	1.839	8.20	5.38	4.53	7.58	9.96	7.64
9	2.012	28.91	6.23	4.09	7.00	14.86	14.27
10	1.846	9.39	4.74	4.10	7.58	5.93	6.69
Promedio	1.785	10.11	5.81	4.38	8.60	7.13	7.59

Postura de sorpresa

La figura 7, muestra la marcación de los puntos sobre el rostro con gesto de sorpresa. Los resultados del cálculo del error de ubicación, con rostro frontal con sorpresa presentados en la tabla 3 continúan siendo muy acertados a la localización de los puntos, e inclusive la desviación es menor a la muestra de solo boca abierta, las cejas levantadas no representan problema siendo estos puntos muy ajustados a la realidad.

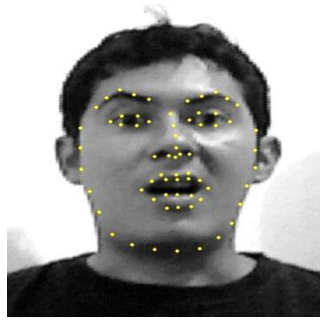


Figura 7. Puntos encontrados para una expresión de sorpresa, una vez estabilizado.

Tabla 3. Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara frontal sorpresa.

Análisis de resultados cara frontal sorpresa.							
Iteración	Tiempo seg.			Error RMS			
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.804	8.51	4.17	5.32	5.36	9.61	7.15
2	1.713	7.80	3.67	3.67	5.97	5.32	5.60
3	1.685	6.41	5.40	5.19	7.00	8.08	6.65
4	1.805	6.32	4.36	4.28	4.92	7.28	5.75
5	1.830	9.74	4.25	5.51	6.04	10.53	7.88
6	1.830	10.11	4.87	4.97	5.17	8.40	7.25
7	2.030	8.78	3.75	4.90	5.84	6.81	6.42
8	1.462	9.00	5.77	5.78	6.19	8.54	7.44
9	2.179	19.79	6.54	5.34	5.98	6.38	9.64
10	1.459	6.49	5.58	4.62	5.97	5.92	5.85
Promedio	1.779	9.30	4.84	4.96	5.84	7.69	6.96

Postura cara hacia arriba

La figura 8 muestra la marcación de los puntos sobre el rostro con postura cara hacia arriba. En la tabla 4 se presentan los resultados, los cuales siguen siendo acertados en la localización de los puntos, no obstante, para esta escena hay un incremento en desviaciones de los puntos del borde inferior de la mandíbula y de la nariz.

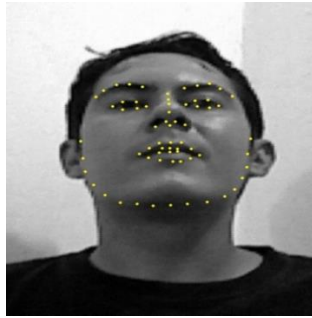


Figura 8. Puntos encontrados en un rostro inclinando la cabeza hacia atrás, una vez estabilizado

Tabla 4. Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara hacia arriba.

Análisis de resultados cara hacia arriba.							
Iteración	Tiempos seg.		Error RMS				
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.061	6.60	8.40	4.23	5.34	2.46	5.19
2	1.268	12.79	8.00	4.82	7.02	2.54	7.13
3	1.392	24.46	7.62	29.42	5.37	4.87	13.77
4	1.924	9.63	7.80	4.33	10.86	3.35	7.14
5	2.950	17.40	7.09	33.48	5.76	5.56	12.69
6	1.381	6.09	4.24	3.71	6.23	2.62	4.56
7	1.440	8.75	6.45	3.98	6.99	3.31	5.95
8	1.281	12.47	6.86	4.25	6.83	3.50	7.03
9	1.564	12.21	6.51	2.97	7.43	3.88	6.95
10	1.396	7.97	6.96	4.14	8.23	3.79	6.20
Promedio	1.565	11.84	6.99	9.53	7.01	3.59	7.66

Postura rostro inclinado a la izquierda

La figura 9, muestra la marcación de los puntos sobre el rostro con postura de cara inclinada a la izquierda. En la tabla 5 se presentan los resultados del cálculo del error de ubicación, los cuales continúan siendo muy satisfactorios, su error es muy similar al obtenido con un rostro frontal neutro.

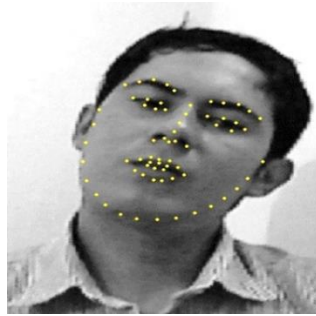


Figura 9. Puntos encontrados en un rostro inclinado a la izquierda, una vez estabilizado

Tabla 5. Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara inclinada a la izquierda

Análisis de resultados cara hacia la izquierda							
Iteración	Tiempo seg.		Error RMS				
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.221	10.48	5.61	5.34	7.33	3.11	6.46
2	1.576	16.25	5.47	4.82	6.13	3.69	7.79
3	1.529	6.45	4.78	5.64	5.50	3.09	5.00
4	1.202	11.20	4.31	6.53	4.91	3.51	6.28
5	1.170	7.38	2.98	6.30	5.23	4.04	5.26
6	1.355	11.62	2.54	5.74	5.58	4.34	6.36
7	1.373	6.69	4.25	6.88	5.32	3.05	5.10
8	1.358	11.31	3.94	6.87	5.34	2.69	6.15
9	1.528	9.37	4.92	5.98	5.47	3.33	5.87
10	1.404	13.05	5.03	5.58	5.25	3.35	6.75
Promedio	1.371	10.38	4.38	5.97	5.61	3.42	6.10

Postura rostro inclinado a la derecha

La figura 10 muestra la marcación de los puntos sobre el rostro con postura de cara inclinada a la derecha. En la tabla 6 se presentan los resultados del cálculo del error de ubicación, los cuales son muy similares a una inclinación a la izquierda y frontal neutro, el inclinar el rostro no representa desviaciones representativas.

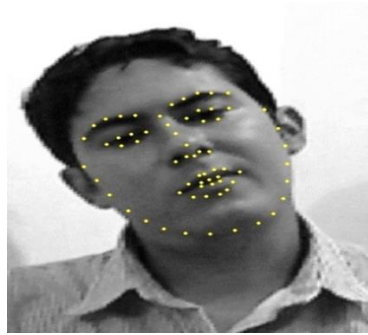


Figura 10. Puntos encontrados en un rostro inclinado a la derecha, una vez estabilizado.

Tabla 6. Resumen de tiempos de estabilización y cálculo de error de ubicación del punto sobre la imagen de cara inclinada a la derecha

Iteración	Análisis de resultados cara hacia la derecha.						
	Tiempo seg.			Error RMS			
	Estabilización	Contorno	Cejas	Nariz	Ojos	Boca	Global
1	1.247	5.51	6.42	6.50	6.46	4.85	5.87
2	1.286	6.94	5.82	5.20	6.62	4.08	5.69
3	1.251	9.08	3.02	4.90	6.63	4.40	5.87
4	1.767	5.24	5.73	5.31	6.17	6.27	5.77
5	1.943	19.34	3.70	5.24	6.31	3.95	8.48
6	1.246	9.62	4.13	3.35	5.87	3.90	5.69
7	1.389	12.07	15.96	5.92	8.17	6.77	9.67
8	1.415	7.02	7.96	15.32	8.14	5.14	7.98
9	1.300	10.33	5.60	4.54	7.10	4.96	6.77
10	0.896	5.43	4.38	4.18	6.30	4.94	5.13
Promedio	1.374	9.06	6.27	6.05	6.78	4.93	6.69

Conclusiones

Al culminar esta investigación acerca del desempeño del algoritmo de seguimiento de las características faciales con respecto a las distintas posturas y expresiones faciales se comprobó que el algoritmo emplea un tiempo promedio de ajuste del modelo menor a 2 segundos y un promedio de frames menor a 9. Además, que el contorno del rostro es el que mayor porcentaje de imprecisión presenta.

Bajo condiciones de luminosidad favorables, el desempeño del algoritmo es bastante bueno, sin embargo, existen posturas del rostro como girar la cabeza sobre el eje vertical en

donde es común que cometa errores con mayor frecuencia. Igualmente, si la persona hace gestos muy rápidos como gesticular la boca.

Para inicializar el modelo se debe iniciar con una postura natural normal, es decir con el rostro de forma frontal y sin inclinaciones, una vez detectado el rostro el modelo lo puede seguir a posiciones muy cercana a los noventa grados de inclinación, es decir casi con el rostro en posición horizontal.

En términos generales el algoritmo trabaja correctamente, aunque con limitaciones de posturas. Sin embargo, durante las pruebas se detectó que si el fondo es muy ruidoso o hay múltiples sombras y objetos, es posible tener un falso positivo haciendo que el algoritmo encuentre una región y haga marcación de los puntos en la imagen.

A través de la realización de este proyecto se pudo determinar que la creación de un algoritmo de seguimiento a las características faciales es un campo en continuo desarrollo. Si bien existen muchos enfoques, estos deben ser optimizados dado que el coste computacional debe ser bajo, de lo contrario su uso en tiempo real se verá muy limitado.

A futuro este proyecto se puede optimizar para realizar diferentes trabajos en el seguimiento de expresiones faciales en tiempo real con dispositivos de bajo costo y más pequeños que el sensor Kinect, pudiéndose aplicar a sistemas embebidos como el Raspberry Pi, que tengan una mejor eficiencia energética y menores requisitos computacionales.

Referencias

- Baltrusaitis, T., Robinson, P., & Morency, L.-P. (2012). 3D constrained local model for rigid and non-rigid facial tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 2610–2617). IEEE.
- Chen, Y., & Davoine, F. (2006). Simultaneous Tracking of Rigid Head Motion and Non-rigid Facial Animation by Analyzing Local Features Statistically. In *BMVC* (pp. 609–618).
- Cheng, S., Zafeiriou, S., Asthana, A., & Pantic, M. (2014). 3D facial geometric features for constrained local mode. In *2014 IEEE International Conference on Image Processing*
- I+D Revista de Investigaciones ISSN 22561676 Volumen XX Número XX Año 2016
Enero-Junio 2016 pp. xx-xx

(*ICIP*) (pp. 1425–1429). IEEE.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790–799.

Cootes, T. F., & Taylor, C. J. (1992). Active shape models—“smart snakes.” In *BMVC92* (pp. 266–275). Springer.

Cristinacce, D., & Cootes, T. F. (2004). A comparison of shape constrained facial feature detectors. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on* (pp. 375–380). IEEE.

Cristinacce, D., & Cootes, T. F. (2006). Feature Detection and Tracking with Constrained Local Models. In *BMVC* (p. 10).

Cristinacce, D., & Cootes, T. F. (2007). Boosted regression active shape models. In *BMVC* (p. 7).

Delac, K., Grgic, M., & Grgic, S. (2005). Independent comparative study of PCA, ICA, and LDA on the FERET data set. *International Journal of Imaging Systems and Technology*, 15(5), 252–260.

Dornaika, F., & Davoine, F. (2004). Head and facial animation tracking using appearance-adaptive models and particle filters. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (pp. 153–153). IEEE.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (pp. I–511). IEEE.

Wang, Y., Lucey, S., & Cohn, J. F. (2008). Enforcing convexity for improved alignment with constrained local models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (pp. 1–8). IEEE.