

# Belleza artificial: evolucionando partes del rostro

Ricardo Solano Monje<sup>1</sup>, Nayeli Joaquinita Meléndez Acosta<sup>2</sup>, Sergio Juárez Vázquez<sup>1</sup>, Homero Vladimir Ríos Figueroa<sup>3</sup>

<sup>1</sup> Universidad del Istmo, Campus Sto. Domingo Tehuantepec, Oax., México

<sup>2</sup> Universidad del Istmo, Campus Ixtepec, Oax., México

<sup>3</sup> Universidad Veracruzana, Xalapa, Ver., México

rsolanomonje@gmail.com, sjuarez@bianni.unistmo.edu.mx,  
nayelimeledez@gmail.com, hrios@uv.com.mx  
<https://www.uv.mx/dia/>

**Resumen.** Este artículo se enfoca en el desarrollo de una maquina embellecedora facial - algoritmo para evolucionar la máscara del rostro de entrada y embellecerlo-que opera como herramienta automatizada de retoque. El aspecto más importante que se considera en este artículo es el de enfocar. La idea de enfocar es identificar las características faciales que necesitan una mejora y concentrar el método de optimización únicamente sobre estas. Entonces la complejidad de manejar un sub-problema, en general, es más pequeña que tratar con el problema entero. Enfocar es implementado para permitir al usuario elegir las características a embellecer. Este algoritmo evolutivo utiliza un modelo SVR, entrenado anticipadamente usando la calificación de belleza generado por Leyvand. El algoritmo nos da la posibilidad de obtener varias “versiones del YO original”, algunas más embellecidas que otras. Hemos usado la base de datos Fg-Net (Face and Gesture Recognition Research Network).

**Palabras clave:** algoritmo evolutivo, embellecimiento artificial del rostro, aprendizaje automático.

## 1. Introducción

El embellecimiento artificial tiene sustento en estudios científicos sobre psicología social y ciencias de la psicología.

En [1] concluyen que son suficientes unos pocos milisegundos para crear una buena o mala impresión sobre una persona. Por otro lado, se sabe que la mente guarda o modela el rostro con rostros prototipo, los cuales son burdos o son carentes de detalles, de forma que se pueden corregir o complementar en la realidad [2]. La mente por

consiguiente, reconoce los rostros con información de bulto o sin detalles. Esto significa que los detalles suelen ser omitidos la primera vez que se ve un rostro, o compensados cuando se ve el rostro por segunda vez. Es decir, que por analogía al reconocimiento de rostros, el embellecimiento artificial pasaría desapercibido para el observador hasta cierto punto [3], [4].

Este artículo presenta un algoritmo evolutivo para el embellecimiento artificial del rostro completo o enfocado a cierta área del rostro. El contenido del artículo está organizado en seis secciones de la siguiente manera: la segunda sección trata sobre cómo se ha tratado el problema del embellecimiento digital en algunos trabajos relacionados. En la tercera sección se muestra el proceso de embellecimiento general en este artículo. En la cuarta sección se explica el funcionamiento del algoritmo evolutivo utilizado en este trabajo. La quinta sección muestra los resultados obtenidos utilizando nuestro método de embellecimiento. Y finalmente en la sexta sección se muestran las conclusiones.

## 2. Embellecimiento del rostro

Un grupo de investigadores en [4] crearon un modelo de aprendizaje automático capaz de calificar la belleza de rostros, dando resultados semejantes a los dados por jueces humanos. Es entonces que Leyvand en [5] propone la técnica para embellecer rostros haciendo uso de dicho modelo de aprendizaje automático.

En [5] para realizar el embellecimiento digital, se parte de la imagen facial de entrada y se determina un conjunto de características faciales predefinidas. Las características faciales -8 en total- están formadas por puntos de referencia y corresponden a 2 cejas, 2 ojos, labio superior, labio inferior, contorno de la nariz y contorno del rostro. A partir de las características faciales y de la posición de los puntos de referencia se calcula un conjunto de distancias entre dichas características.

Leyvand en [5] localiza 84 puntos de referencia, mientras que Eisenthal en [4] localiza 37 puntos de referencia. Con estos puntos se construye un vector  $\mathbf{v}$  una máscara. El objetivo de la máscara es formar un vector de distancias o longitudes entre los puntos de referencia que definen las características faciales. Leyvand obtiene 234 distancias. La diferencia de localización de puntos entre estos dos trabajos tiene una explicación sencilla, Leyvand usa la máquina calificadora de Eisenthal, pero mientras que Eisenthal persigue en su trabajo calificar la belleza de un rostro, Leyvand realiza la modificación y embellecimiento de un rostro.

## 3. Embellecimiento evolutivo

El principal componente de esta investigación es la máquina embellecedora - algoritmo para evolucionar la máscara del rostro de entrada y embellecerlo-, que hace uso de un conjunto de imágenes de rostros de mujeres -el ensamble de entrenamiento y sus calificaciones de belleza asociadas. El proceso de embellecimiento se muestra en la Fig. 1.

Dada una foto frontal como entrada donde el usuario ya seleccionó las características faciales a modificar –aunque podría ser todo el rostro de forma automática- se identifican en primera instancia los puntos de referencia que conllevan a especificar varias características faciales.

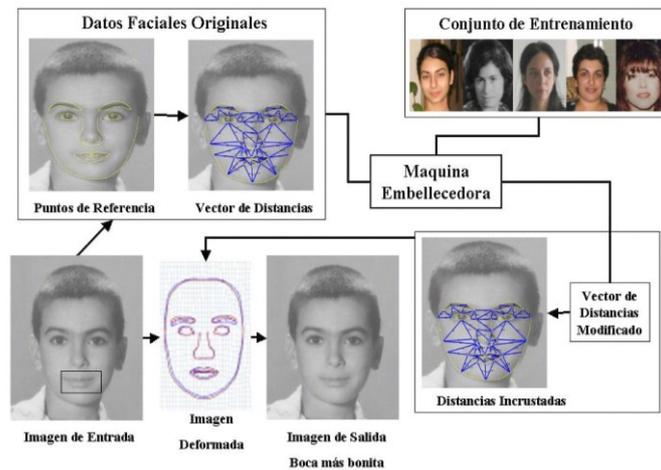


Fig. 1 Procedimiento de Embellecimiento

A partir de los puntos de referencia y entre las características faciales, se extrae un vector de distancias, que corresponde a la malla o máscara del rostro de entrada. Esta máscara alimenta al Algoritmo Evolutivo, el cual produce un vector de distancias modificado, que posee una calificación de belleza más alta que la del vector original.

La modificación de la imagen de entrada se llevará a cabo con transformaciones warping, este proceso mapea el conjunto de puntos de referencia  $\{p_i\}$  de la imagen origen en el correspondiente conjunto de puntos de referencia  $\{q_i\}$  de destino. De la Fig. 1 recuérdese que la máscara -el vector de distancias- permite realizar la transformación warping del rostro de entrada para producir el rostro de salida embellecido.

ENFOCAR. La idea de enfocar es identificar las características faciales que necesitan una mejora y concentrar el método de optimización únicamente sobre estas. Entonces la complejidad de manejar un sub-problema, en general, es más pequeña que tratar con el problema entero. Enfocar es implementado para permitir al usuario elegir las características a embellecer. Los puntos de referencia de las características elegidas son llamados puntos seleccionados, el resto son llamados puntos fijos. Con esto, el método de optimización se enfoca en embellecer las características faciales seleccionadas, mientras los puntos de referencia fijos preservan sus coordenadas espaciales. La Fig. 2 muestra una imagen de entrada donde se elige la boca.

Enfocar no es un proceso completamente automático, se considera una aplicación interactiva donde el usuario juega un rol activo para ayudar en el método de optimización -en realidad al proceso de afinar el embellecimiento a la medida- en la cual el usuario selecciona una o varias de las características faciales del rostro de entrada. Al interactuar con el método de optimización el usuario proporciona la o las características a embellecer, este ajuste reduce el espacio de soluciones a ser explorado, pues de

no especificar características faciales específicas, el rostro es embellecido en su totalidad.



Fig. 2 Rostro a embellecer con una característica facial seleccionada –boca

#### 4 Evolucionando la mascara

En nuestro caso y para fines explicativos, se considera que partiendo del rostro de entrada se extrae un total de 66 puntos de referencia. Al igual que [5], los puntos de referencia son localizados en los contornos de las 8 diferentes características faciales: las dos cejas, los dos ojos, el labio inferior y superior de la boca, la nariz y límites del rostro, Fig. 3a. La posición de cada uno de los 66 puntos de referencia extraídos son usados para construir la máscara, Fig. 3b. La máscara consiste de 89 distancias y éstas forman el vector de distancias, Fig. 3c.

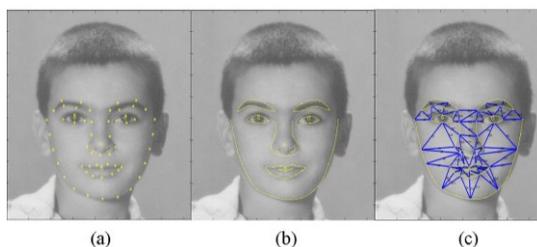


Fig. 3 (a) Los 66 puntos de referencia. (b) Las características faciales. (c) Las 89 distancias entre puntos de referencia

##### 4.1. Individuos del algoritmo evolutivo

Sea  $M$  la máscara del rostro. A partir de ésta, se construye un individuo  $I$  formado por los puntos de referencia seleccionados de las características seleccionadas de la máscara, y un sub-vector de distancias, formado por las distancias entre los puntos de referencia seleccionados. La Fig. 4 muestra la estructura genética de un individuo  $I$ . Un individuo  $I$  ésta compuesto de dos partes principales:

1. PuntosDeReferencia: un vector de longitud  $n$  fija, formado por las coordenadas  $X - Y$  de los  $n$  puntos de referencia seleccionados.

2. SubVectorDistancias: un vector de longitud  $m$  fija, con las distancias entre cada uno de los puntos de referencia seleccionados.

El algoritmo genera diferentes individuos, por ejemplo, si la característica seleccionada fuese la boca, la Fig. 5 muestra un conjunto de individuos generados por el Algoritmo Evolutivo, cada uno representa una característica mutada con una calificación diferente dependiendo de las proporciones (distancias entre los puntos de referencia) del individuo  $I$  ya colocado en la máscara.

PuntosDeReferencia					
$X_1$	$X_2$	$X_3$	$X_4$	...	$X_n$
$Y_1$	$Y_2$	$Y_3$	$Y_4$	...	$Y_n$
SubVectorDistancias					
$D_1$	$D_2$	$D_3$	$D_4$	...	$D_m$

Fig. 4 Representación genética de un individuo

Recuérdese que un individuo sólo contiene información de los puntos de referencia seleccionados y el sub-vector de distancias, de esta forma el algoritmo evolutivo manipula la máscara completa y una lista de restricciones del individuo.

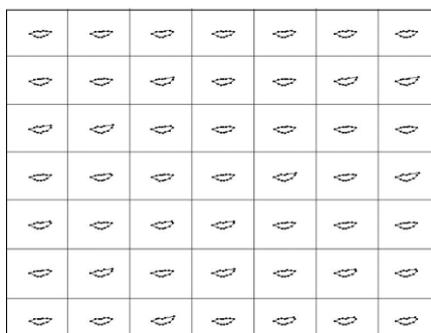


Fig. 5 Bocas -individuos- generadas por evolución usando mutación por escala

#### 4.2. Extracción y unión

Un individuo es extraído directamente de la máscara completa  $M$ , mientras esta contenga un conjunto de puntos de referencia seleccionados. A esta operación se le llama Extracción y se usa la notación Extraer ( $M$ ) para indicar la extracción de un nuevo individuo  $I$  de  $M$ . Extracción es usada por el Algoritmo evolutivo para producir el primer individuo.

Es posible combinar la máscara completa con un individuo, esta operación es llamada Unión. La notación  $\text{Unir}(I, M)$  representa la unión de la máscara completa  $M$  con el individuo  $I$ . La operación unión actualiza las coordenadas de todos los puntos de referencia seleccionados en  $M$  de acuerdo a la información en  $I$ .  $M$  podría ya tener un diseño para los elementos seleccionados, el cual es reemplazado por la información en el individuo.

Extracción y Unión son usadas por el Algoritmo evolutivo en las diferentes etapas de mejoramiento de la máscara. Por ejemplo, el primer individuo producido es extraído de la máscara de entrada. Posteriormente nuevos individuos son unidos con esta máscara para crear una máscara completa para ser evaluada. Estas operaciones se pueden observar en la Fig. 6.

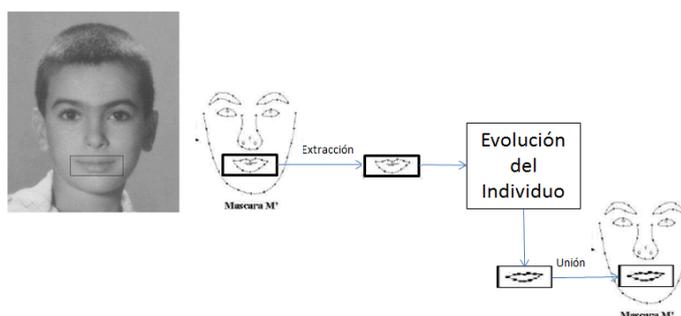


Fig. 6 Operaciones extracción y unión

### 4.3. Función de evaluación

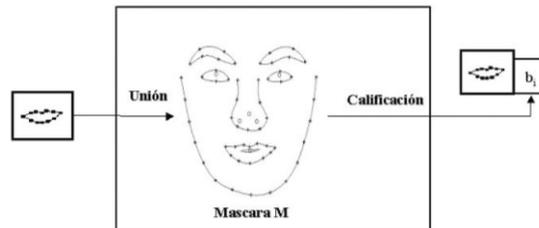
La función de evaluación proporciona una calificación de belleza  $b$  a cada máscara modificada. Un Support Vector Regression (SVR) es construido basado en entrenamiento con un conjunto de ejemplos  $(x, y)$ , donde  $x \in R^d$  y  $y \in R$ . El Algoritmo evolutivo utiliza un modelo SVR, entrenado anticipadamente usando la calificación de belleza generado por [6].

El vector de distancias de 89 dimensiones y su correspondiente calificación  $b$  de belleza (en una escala de 1 a 7) son usados como ejemplo de entrenamiento para construir un modelo SVR. El SVR define una función  $f_b: R^d \rightarrow R$ , la cual se usa para estimar las calificaciones de belleza, de los vectores de distancias de los rostros que no se encuentran el conjunto de entrenamiento.

Sea  $v$  el vector de distancias normalizado extraído de las características seleccionadas en la máscara de una imagen facial de entrada. El objetivo del proceso de embellecimiento usando Algoritmos evolutivos es crear un vector vecino  $v'$  con una calificación de belleza más alta  $f_b(v') > f_b(v)$ .

El modelo SVR verifica cada una de las proporciones (distancias) de la máscara del rostro formado al unir la máscara  $M'$  (una copia de  $M$ ) y un individuo  $I$ , para proporcionar una calificación de belleza, como se muestra en la Fig. 7. En general, las

calificaciones más altas tienen como resultado más belleza y por consiguiente son una mejor solución.



**Fig. 7** Función de Evaluación

A cada individuo producido por el sistema es asignando una calificación. Sean dos individuos  $I_1$  e  $I_2$  con una calificación de belleza  $b_1$  y  $b_2$  respectivamente, se dice que  $I_1$  es mejor que  $I_2$ , denotado por  $I_1 > I_2$ , si  $b_1$  es mas grande que  $b_2$ . Usando este ordenamiento, el objetivo es producir mascarar con una calificación mayor que la máscara de la imagen de entrada, lo que nos lleva al Ciclo Evolutivo.

#### 4.4. Ciclo evolutivo

La entrada del algoritmo evolutivo es una máscara  $M$  inicial obtenida del rostro de entrada y una lista  $R$  de restricciones. El algoritmo produce una población de individuos representada por sub-mascaras diferentes a la máscara de entrada.

Entonces, el algoritmo evolutivo ejecuta los siguientes pasos:

1. Crea una copia  $M'$  de la máscara  $M$ . La copia es usada durante todo el proceso del algoritmo.
2. Una población inicial  $P_0$  es creada extrayendo un individuo  $I$  de  $M'$  y aplicándole los operadores de mutación reiteradamente a  $I$ , para producir nuevos individuos.  
Se usa una población de 21 individuos.  $P_0$  contendrá el primer individuo  $I$  y 20 versiones mutadas.
3. Se inicia un contador de generaciones  $iter = 0$ .
4.  $P_{iter}$  es evaluada para calcular una calificación  $b_i$  para cada individuo en la población.
5. Una Sub-población  $S$  es seleccionada de  $P_{iter}$  por *certamen*: el mejor individuo en  $P_{iter}$  es directamente insertado en  $S$  (El mejor individuo es aquel que tiene la mejor calificación proporcionada por la función de evaluación). El resto de individuos en  $P_{iter}$  toman parte en competencias partwise: para cualquier  $I_1$  e  $I_2$  que aún no han participado en una competencia y elegidos aleatoriamente de  $P_{iter}$ , si  $I_1 < I_2$  se agrega  $I_1$  en  $S$ ; en caso contrario se agrega en  $I_2$  a  $S$ . En total 10 concursos partwise son ejecutados, resultando 10 ganadores. Al final de este paso,  $S$  tiene 11 individuos.
6.  $iter$  es incrementado  $iter \leftarrow iter + 1$

7. Una nueva población,  $P_{iter}$ , es creada: el mejor individuo en  $S$  (decidido en el paso 5) es transferido directamente a  $P_{iter}$ ; entonces dos individuos  $I_1$  e  $I_2$  son elegidos aleatoriamente y removidos de  $S$ . Estos individuos son combinados por un operador de cruce y son agregados a  $P_{iter}$ . El proceso se repite hasta que  $S$  este vacía. Para completar la nueva población, los otros 10 individuos son producidos por mutación del mejor individuo en  $P_{iter-1}$ .
8. Si se alcanza el criterio de paro, entonces termina el ciclo evolutivo, y la salida del algoritmo evolutivo es  $P_{iter}$ ; en otro caso, se regresa al paso

En el algoritmo evolutivo, los nuevos individuos son creados a través de mutación o combinación (cruza) de los existentes. Buenos individuos (con las calificaciones más altas), se espera que tengan éxito en el torneo selectivo y propaguen sus características a las siguientes generaciones. El algoritmo evolutivo se detiene hasta que la máscara del mejor individuo ya no tiene mejoras después de  $L$  iteraciones consecutivas del ciclo, y sólo  $50 * L$  generaciones tienen que ser producidas. La Fig. 8 ilustra la funcionalidad interna del algoritmo evolutivo.

#### 4.5. Operadores

En esta subsección se describen los operadores de mutación y cruce implementados en el algoritmo evolutivo propuesto. La cruce combina información genética y requiere de dos individuos, mientras que la mutación cambia al individuo mismo.

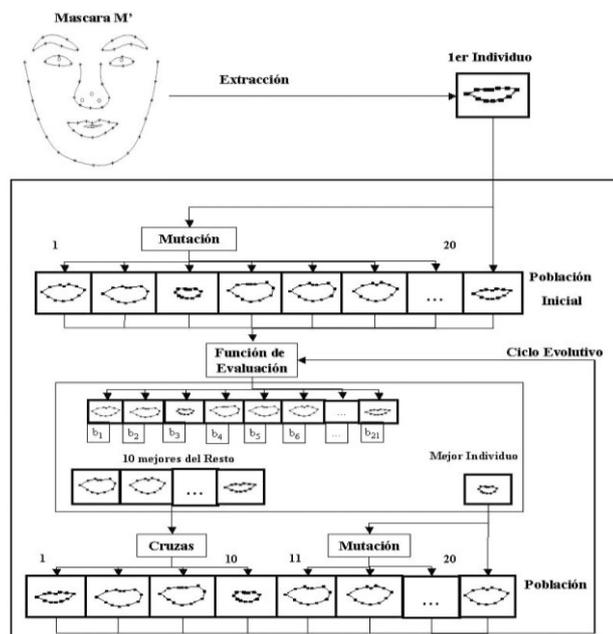


Fig. 8 Funcionalidad Interna del Algoritmo evolutivo

**Mutación.** La cruce sólo puede encontrar nuevas combinaciones de material genético ya presente. Para introducir nuevo material genético en la población, el cambio aleatorio de individuos es necesario por ejemplo a través de la mutación. En nuestra propuesta se implementaron cinco operadores de mutación.

*ESCALAR* (Individuo  $I$ ). Este operador cambia las coordenadas de los puntos de referencia seleccionados que conforman el individuo  $I$ . Aleatoriamente se elige un número  $n$  de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia  $a$ , una dirección de escala  $d = -1$  (Reducir) o  $d = 1$  (Aumentar) y una escala  $S$  dependiendo de la dirección, si  $d = -1$  entre  $[-1, 0)$  y si  $d = 1$  entre  $(0, 1]$  y se toman en cuenta las restricciones que el usuario introdujo, si la escala rebaza un límite. La estrategia para escalar un objeto con respecto al punto arbitrario  $P_1$  es la siguiente: primero se hace el traslado de manera que  $P_1$  pase al origen, después se escala y luego se traslada de regreso a  $P_1$ . La Ecuación (1) muestra la matriz resultante de multiplicar las tres transformaciones mencionadas anteriormente, necesarias para escalar. El punto arbitrario  $P_1$  representado en la ecuación con su coordenada  $(x_1, y_1)$  es el punto medio del individuo.  $S(sx, sy)$  es la escala en  $x$  y  $y$  respectivamente.

$$T(x_1, y_1) * S(sx, sy) * T(-x_1, -y_1) = \begin{pmatrix} sx & 0 & x_1(1-sx) \\ 0 & sy & y_1(1-sy) \\ 0 & 0 & 1 \end{pmatrix} \quad (1)$$

*ROTAR* (Individuo  $I$ ). Este operador cambia las coordenadas de los puntos de referencia seleccionados del individuo  $I$ . Aleatoriamente se elige un número  $n$  de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia  $a$ , un grado a rotar  $\alpha$  (entre  $0^\circ$  y  $45^\circ$ ) y una dirección de rotación  $d = -1$  (Derecha) o  $d = 1$  (Izquierda), si la rotación sobre pasa los límites de las restricciones, se ejecuta otra iteración.

En este caso, la transformación de rotar utiliza la Ecuación 2. Esta mutación utiliza una estrategia similar al de *ESCALAR*. La rotación es con respecto a un punto arbitrario  $P_1$ : primero se traslada  $P_1$  al origen, se rota y se traslada para que el punto en el origen regrese a  $P_1$ . El punto arbitrario representado con la coordenada  $(x_1, y_1)$  en la ecuación, es el punto medio del individuo.

$$T(x_1, y_1) * R(\theta) * T(-x_1, -y_1) = \begin{pmatrix} \cos\theta & -\text{sen}\theta & x_1(1-\cos\theta)+y_1\text{sen}\theta \\ \text{sen}\theta & \cos\theta & y_1(1-\cos\theta)+x_1\text{sen}\theta \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

*TRASLADAR* (Individuo  $I$ ). Este operador cambia las coordenadas de los puntos de referencia (seleccionados), que conforman el individuo  $I$ . Aleatoriamente se elige un  $n$  número de iteraciones, después por cada iteración: se elige aleatoriamente un punto de referencia  $a$ , una dirección (arriba-abajo o vertical-horizontal) y un desplazamiento  $d$ .

El valor a  $d$  es de acuerdo a una distribución lineal de probabilidades donde los valores pequeños de desplazamiento tiene un probabilidad más alta de ser elegidos, si el desplazamiento rebaza un límite, esa iteración no es tomada en cuenta. La traslación se realiza utilizando la Ecuación 3, donde el desplazamiento en  $x$  y  $y$  es representado por  $(dx, dy)$  respectivamente

$$T(dx,dy)= \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

**CAMBIO-ALEATORIO** (Individuo  $I$ ). Este operador cambia las coordenadas de un punto de referencia  $a$  en el individuo  $I$ . Aleatoriamente se elige un punto de referencia  $a$ , una dirección (arriba-abajo o vertical-horizontal) a mover y un desplazamiento  $k \in Z^*$ . El valor a  $k$  es de acuerdo a una distribución lineal de probabilidades donde los valores pequeños absolutos de desplazamiento tienen una probabilidad más alta de ser elegidos, tomando en cuenta las restricciones que el usuario introdujo.

**CAMINAR** (Individuo  $I$ ). Este operador sigue la misma idea de la mutación Walk-BC introducida por [7]. Nuestra versión de Walk-BC solo mueve los puntos de referencia seleccionados. Aleatoriamente se elige un punto de referencia  $a$  del individuo  $I$  y una dirección vertical  $d = 1$  (para Arriba) o  $d = -1$  (para Abajo). La coordenada  $X$  de  $a$  en  $I$  es igual a la media aritmética de todos los puntos de referencia adyacentes de  $a$  en la capa  $v.l - d$ . Entonces un punto de referencia  $u$  adyacente a  $v$  en la dirección  $d$  es elegido aleatoriamente y el proceso se repite para  $u$ . La mutación continúa hasta que ya no haya más puntos de referencia adyacentes a elegir en la dirección  $d$ .

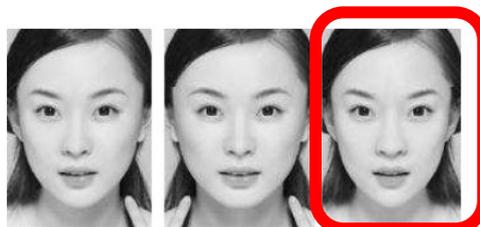
**Cruza.** El propósito del operador de cruza es crear nuevos individuos que se espera que sean mejores que los padres. El algoritmo evolutivo tiene dos operadores de cruza:

**COMBINAR-PUNTOS** (Individuo  $I_1, I_2$ ). Elige aleatoriamente  $n$  puntos de referencia  $a_n$  que tiene diferentes coordenadas en  $I_1$  e  $I_2$ , y cambia las posiciones  $X - Y$  de todos los  $n$  puntos de referencia entre los individuos.

**COMBINAR-PARTES** (Individuo  $I_1, I_2$ ). Elige aleatoriamente un punto de referencia  $a$ , los nuevos individuos están conformados como sigue:  $I_{1Hijo}$  está formado por los puntos de referencia anteriores a  $a$  de  $I_1$  y los puntos de referencia a partir de  $a$  hasta el final de  $I_2$ , y  $I_{2Hijo}$  está formado por los puntos de referencia anteriores a  $a$  de  $I_2$  y los puntos de referencia a partir de  $a$  hasta el final de  $I_1$ .

## 5 Resultados

En la Fig. 9 se muestran resultados del embellecimiento evolutivo, los resultados son evidentes, la posibilidad de la obtención de “muchas versiones del YO original”, algunas embellecidas más que otras.



**Fig. 9.** Las imágenes al centro y der. son versiones embellecidas que se obtuvieron evolucionando el original a la izq. La imagen de la der. tiene mayor puntuación de belleza que la del centro. Imagen izq. (imagen original) tomada de Huiyun, et al. [8]

La Fig. 10 y Fig. 11 muestran más resultados que se obtuvieron utilizando el sistema implementado en este artículo. Es importante mencionar que nuestro algoritmo evolutivo esta implementado en Matlab y el tiempo de ejecución fue de aproximadamente 45 segundos.



**Fig. 10.** Algunos resultados haciendo uso del algoritmo evolutivo. Las imágenes de la der. tienen mayor puntuación de belleza que las de la izquierda. Nótese cómo la boca a sufrido cambios positivos en ambos rostros.



**Fig. 11.** Otros resultados haciendo uso del algoritmo evolutivo. Las imágenes de la der. tiene mayor puntuación de belleza que las de la izquierda. Nótese cómo la boca ha sufrido cambios positivos en ambos rostros. La imagen superior también muestra cambios en el mentón.

Actualmente nuestra técnica está limitada a rostros de vista frontal y bajo una expresión neutral, en los resultados experimentales se utilizó base de datos de imágenes Fg-Net [9].

## 6. Conclusiones

La innovación de este trabajo son las características de *ENFOCAR*, para embellecer algunas partes del rostro de entrada. En nuestra propuesta se le permite al usuario sólo trabajar con una parte del rostro a través de la máscara, mejorando únicamente la característica seleccionada que no le son favorables en la imagen. Se diseñó un algoritmo evolutivo para generar un rostro más bello que el de entrada.

## Referencias

1. Moshe, B., Maital, N. and Heather, L.: Very first impressions. *Emotion by the American Psychological Association*, 6(2):269–278 (2006)
2. Winkielman, P., Halberstadt, J., Fazendeiro, T. and Steve, C.: Prototypes are attractive because they are easy on the mind. *Psychological Science*, 17(9):799–806 (2006)
3. Pawan, S., Benjamin, B., Ostrovsky, Y. and Richard, R.: Face recognition by humans: 20 results all computer vision researchers should know about. Technical report, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology (2005)
4. Eysenck, H., Dror, G. and Ruppin, E.: Facial attractiveness: Beauty and the machine. *Neural Computation*, 18(1):119–142 (2006)
5. Leyvand, T., Cohen Or, D., Dror, G. and Lischinski, D.: Data Driven Enhancement of Facial Attractiveness. *ACM Trans. Graph.*, 27(3):9 (2008)
6. Kagian, A., Dror, G., Leyvand, T., Cohen-Or, D. and Ruppin, E.: A Humanlike Predictor of Facial Attractiveness. In: Schölkopf B., Platt J., and Hoffman T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press. (2007)
7. Utech, J., Branke, J., Schmeck, H. and Eades, P.: An Evolutionary Algorithm for Drawing Directed Graphs. In: *International Conference on Imaging Science, Systems, & Technology (CISST'98)*, pages 154–160 (1998)
8. Huiyun, M., Lianwen, J. and Minghui, D.: Automatic Classification of Chinese Female Facial Beauty using Support Vector Machine. In: *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4987– 4991 (2009)
9. Cootes, T.: The Fg-Net Aging Database, <http://sting.cycollege.ac.cy/~alanitis/fgnetaging/index.htm>