

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Ingeniería en Computación

Proyecto Terminal

“Morphing de imágenes 3D en base a sus puntos de luz”

Trimestre Lectivo 11-O

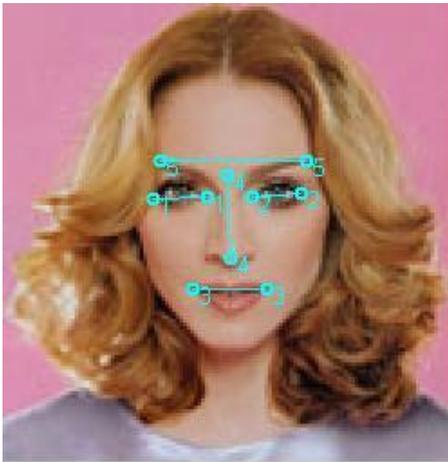
Alumno: Eduardo Rojas Silva 204304042

Asesor: Dr. Risto Fermín Rangel Kuoppa

Marco Teórico

Morphing es un efecto que consiste en el cambio de una imagen en otra, es decir, crea la ilusión de la transformación de una imagen en otra.

El desarrollo de esta técnica empezó al principio de los 90's. Uno de los primeros artículos que reportan un método de Morphing es "Feature-Based Image Metamorphosis" de Beier y Neely esta técnica consistía en distorsionar una imagen y al mismo tiempo desvanecerla tomando como referencia algunos vectores definidos en la imagen fuente (antes) como en la imagen destino (después). Por ejemplo, uno podría transformar una cara en otra, marcando los puntos clave en la primera cara, como el contorno de la nariz o la ubicación de un ojo, y la marca en estos mismos puntos en la segunda cara. Con estos puntos se distorsionaría la primer cara en forma de la segunda y al mismo tiempo que se desvanece la primer cara y se va intensificando la segunda.



Objetivos Propuestos

Objetivo General

Desarrollar e implementar un software que implemente una aproximación del método reportado por Z. Zhang en para realizar *Morphing* a imágenes 3D en base a sus puntos de luz.

Objetivos Particulares

- Crear un módulo que cargue la imagen a transformar
- Crear un módulo para identificar los puntos de luz de una imagen
- Crear un módulo para transformar los puntos de luz en polígonos
- Crear un módulo que realice un ajuste entre los polígonos de las imágenes.
- Crear un módulo que permita ver el progreso de la transformación de la imagen
- Documentar la transformación de al menos cinco *imágenes significativas*¹

Antecedentes

Dado a que desde la aparición de esta técnica, ha sido explotada por la industria cinematográfica, hoy en día se cuenta con software dedicado para la implementación de esta técnica con muy poca interacción por parte del usuario.

Algunos de estos son:

Abrosoft FantaMorph

<http://www.fantamorph.com/index.html>

Es un software que mediante el ingreso de dos imágenes este va haciendo la transformación y el resultado lo guarda en un archivo de video, para poder observar cómo es que se va realizando la transformación paulatina de una imagen a otra.

Morpheus Photo Morpher

<http://www.morpheussoftware.net/>

Este software permite que mediante el ingreso de dos imágenes realice la transformación y las imágenes que va generando las guarda en formato BMP o JPG, adicionalmente tiene la opción de generar un video con la transformación realizada.

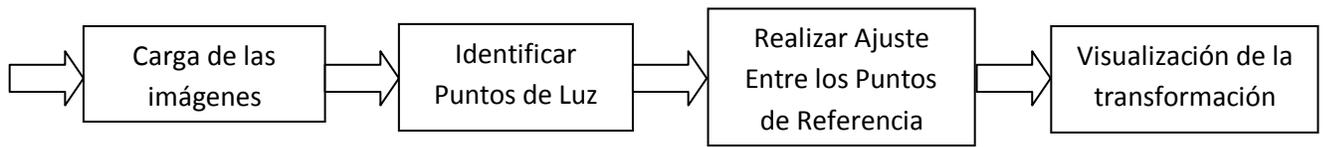
¹Una imagen significativa, es una imagen que ha sido usada en artículos anteriores, que se utilizará en el presente trabajo para hacer una comparación de resultados.

Desarrollo del Proyecto

El propósito de este proyecto es poder realizar Morphing entre dos imágenes, mostrando el paulatino progreso de la transformación.

La unidad mínima de procesamiento de una imagen es el "pixel", por lo que para el desarrollo de este proyecto se basa en operaciones entre ellos, así como en la descomposición de los pixeles en sus 3 fuentes de color (rojo, azul y verde), y las operaciones entre ellos.

Los pasos generales para el procesamiento de imágenes de este proyecto son:



Carga de las Imágenes

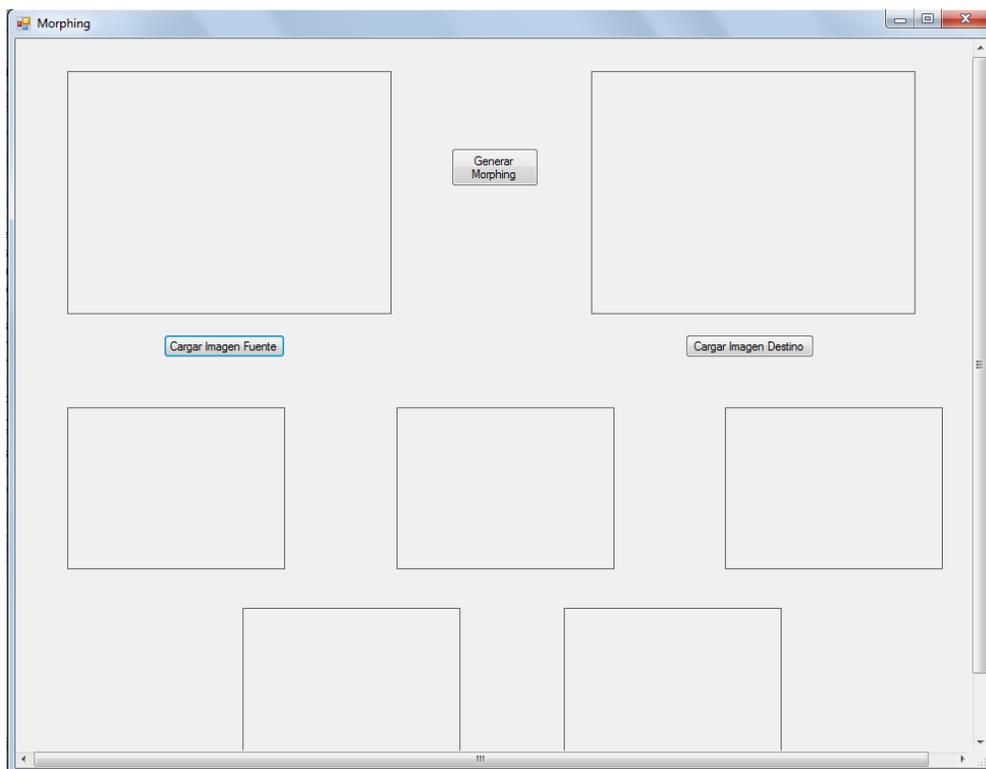
Es el proceso mediante el cual a través de la interfaz se cargan las dos imágenes que se tendrá como referencia para realizar el Morphing.

Imagen 1: Imagen Fuente

Imagen2: Imagen Destino

Las imágenes son en formato jpg, en un principio se tenía considerado como tamaño máximo de las imágenes 640x480 pixeles, sin embargo por cuestiones de tiempo de procesamiento se decidió reducir a la mitad el tamaño máximo de las imágenes quedando con un máximo de 320x240 pixeles, las imágenes a procesar deberán de tener el mismo tamaño y tener un fondo oscuro para aumentar el contraste para así maximizar la visión de la imagen a procesar.

Al inicio del programa se muestra una plantilla como la siguiente, donde se comienza cargando las imágenes.



Identificar Puntos de Luz

Consiste en reducir las imperfecciones que pueda tener la imagen, para este paso primero se utilizó el algoritmo de Canny Edge para poder obtener los contornos de las imágenes para en base a estos seleccionar los puntos de luz de referencia para el siguiente paso del procesamiento.

Como entrada se tiene las dos imágenes que fueron cargadas y como salida se obtienen dos imágenes sólo con los contornos definidos de donde se seleccionarán los puntos de luz. Todo este procesamiento es invisible para el usuario.



Algoritmo de Canny Edge

El algoritmo de Canny Edge es un algoritmo que sirve para detectar bordes en las imágenes. Este algoritmo consiste en dos etapas, en la primera etapa se busca reducir el ruido en base a un filtro gaussiano.

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Matriz del filtro Gaussiano utilizado para reducir las imperfecciones.

En la segunda etapa se debe obtener el gradiente de cada pixel, esto se hace mediante la utilización de dos máscaras de convolución de 3x3, una para estimar el gradiente en la dirección 'x' (columnas) y la otra para estimar el gradiente en la dirección 'y' (renglones).

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

Matriz que obtiene el gradiente en la dirección "x".

$$Gy = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Matriz que obtiene el gradiente en la dirección "y".

Realizar ajustes entre los puntos de referencia

Una vez que se tienen los contornos de las imágenes se procede a hacer la interpolación de las imágenes empezando por los puntos de referencia previamente obtenidos, y continuando con los demás puntos visibles del contorno. Esto se realiza recorriendo primero los renglones para localizar los puntos de ambas imágenes (la fuente y la de destino) para realizar dicha interpolación y después con las columnas realizamos el mismo procedimiento.

Una vez hecho esto lo que nos queda es el contorno de ambas figuras interpoladas, para los píxeles dentro del contorno también se realiza una interpolación pero esta interpolación se realiza en base al color que tiene cada píxel correspondiente en la imagen fuente como en la de destino.

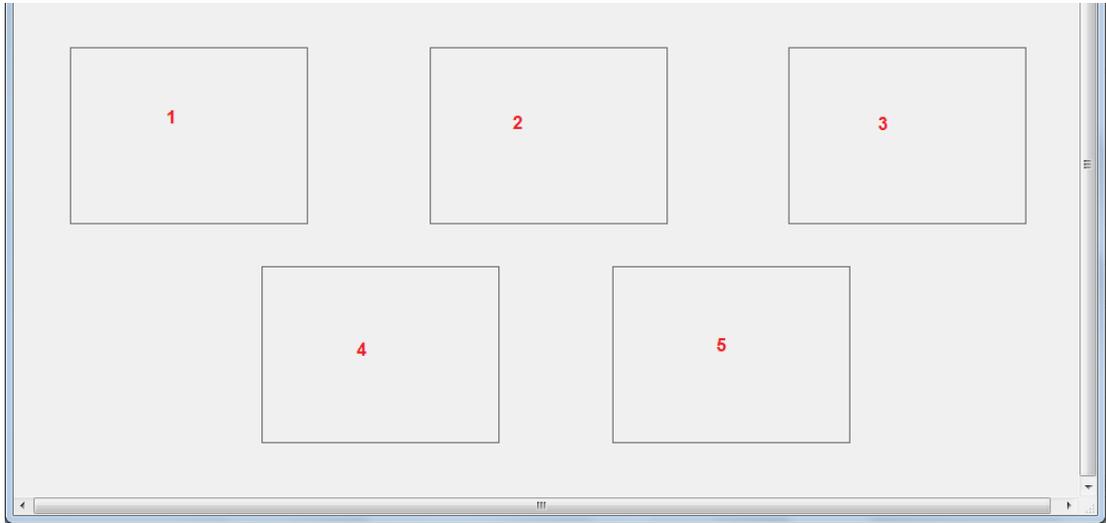
Este procedimiento tiene como entrada las imágenes fuente y destino, así como las imágenes con solo el contorno de estas, y tiene como salida 5 imágenes que muestran la transformación gradual que se llevó a cabo.



Visualización de la transformación

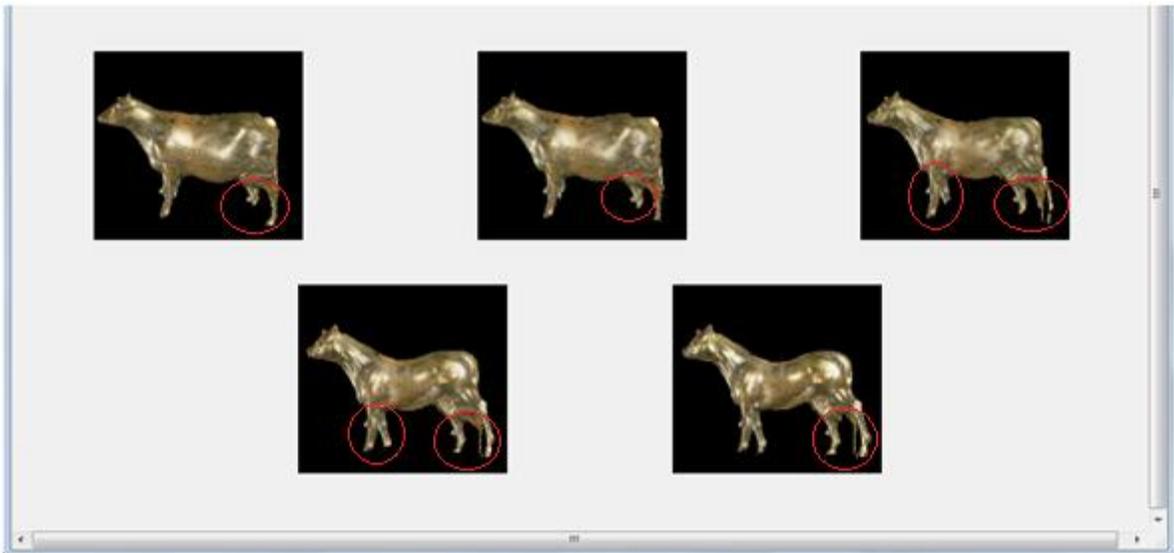
Una vez que tenemos las imágenes que muestran la transformación paulatina éstas se muestran en la interfaz.

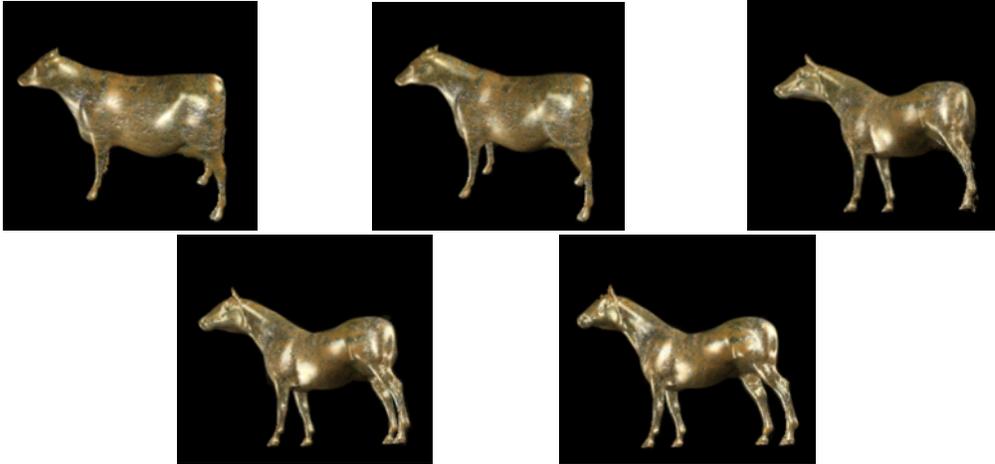
A continuación se muestran numerados los cuadros en donde se mostrarán las imágenes, la imagen del recuadro 1 es la más parecida a la imagen fuente y la imagen 5 es la más parecida a la imagen destino.



Resultado

A continuación se muestran las imágenes que se obtuvieron como resultado en este proyecto, comparadas con las imágenes reportadas en el artículo fuente de inspiración des este proyecto.





La principal diferencia que se puede apreciar, es en las partes más finas de la imagen, marcadas por un círculo rojo, en donde no se logró un ajuste adecuado en las imágenes intermedias, pero se puede apreciar que en la imagen más cercana a la imagen destino, las patas van tomando el aspecto de deberían de tener.

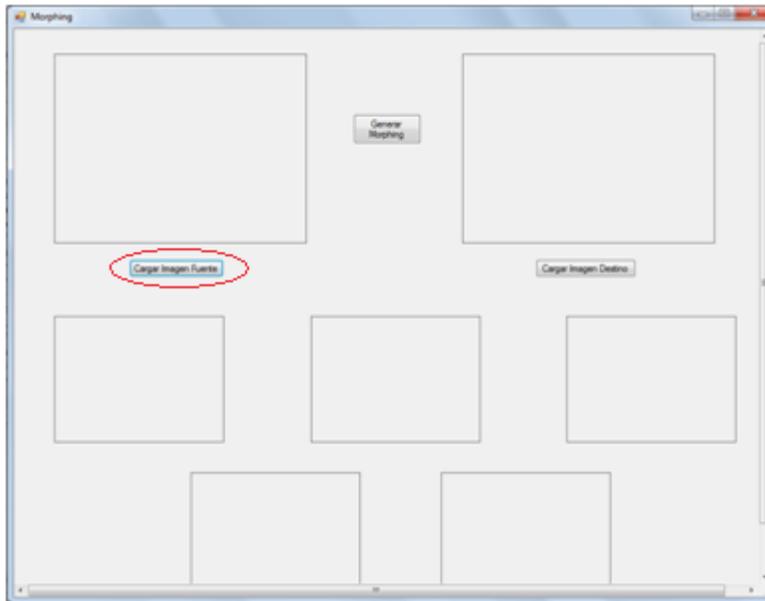
En cuanto a los demás aspectos de la imagen ésta es bastante consistente con las imágenes contra las que se compara, por lo que se puede concluir que se obtuvo un Morphing de buena calidad.

Manual del Usuario

El formato de las imágenes de entrada es JPG.

El tamaño máximo de imágenes a ingresar es de 320x240 pixeles, entre menor sea el tamaño de la imagen mayor será la velocidad de procesamiento.

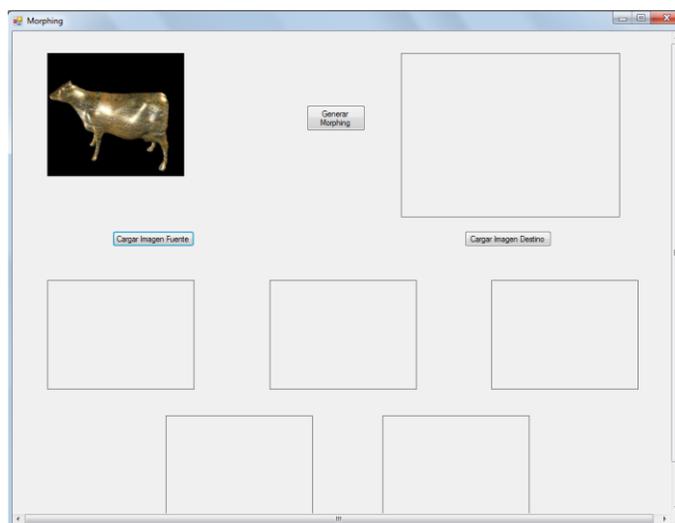
Al inicio del programa se muestra una pantalla como la siguiente, en la cual se comienza cargando las imágenes.



Cargar Imagen Fuente

Se selecciona el botón “Cargar Imagen Fuente” a continuación aparecerá un cuadro de dialogo donde se buscará la imagen que se desea cargar. En caso de que la imagen exceda el tamaño máximo permitido el programa lanzará una advertencia notificando que la imagen seleccionada no puede ser cargada porque sus dimensiones son muy grandes.

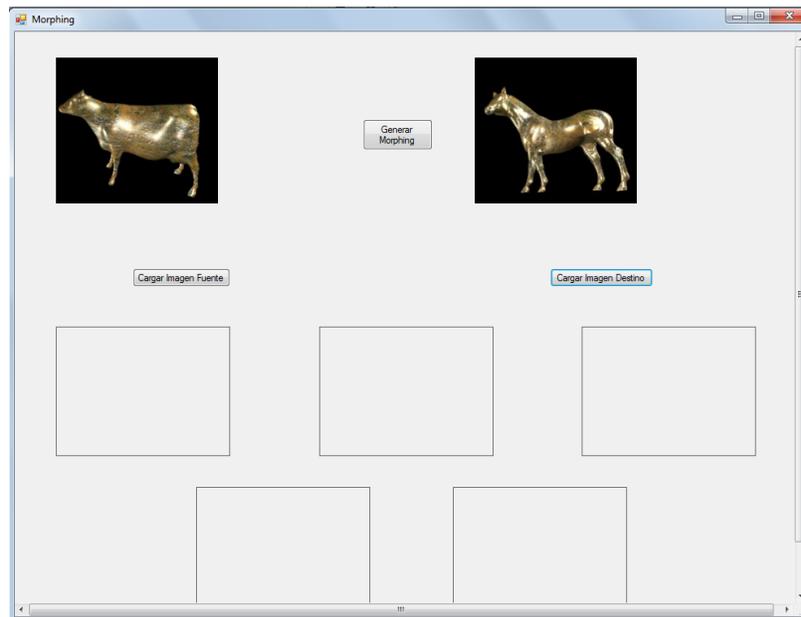
Una vez cargada la imagen se muestra la siguiente pantalla.



Cargar Imagen Destino

Después se selecciona el botón “Cargar Imagen Destino” al igual que con el botón anterior aparecerá un cuadro de dialogo donde se buscará la imagen a cargar. De igual manera que con la imagen anterior en caso de que ésta exceda las dimensiones permitidas se le lanzará una advertencia al usuario, adicionalmente se debe comprobar que la imagen destino tenga las mismas dimensiones que la imagen fuente, de no ser así se le mandará una advertencia al usuario notificándole que ambas imágenes deben poseer las mismas dimensiones.

Una vez cargada la imagen destino se muestra la siguiente pantalla.

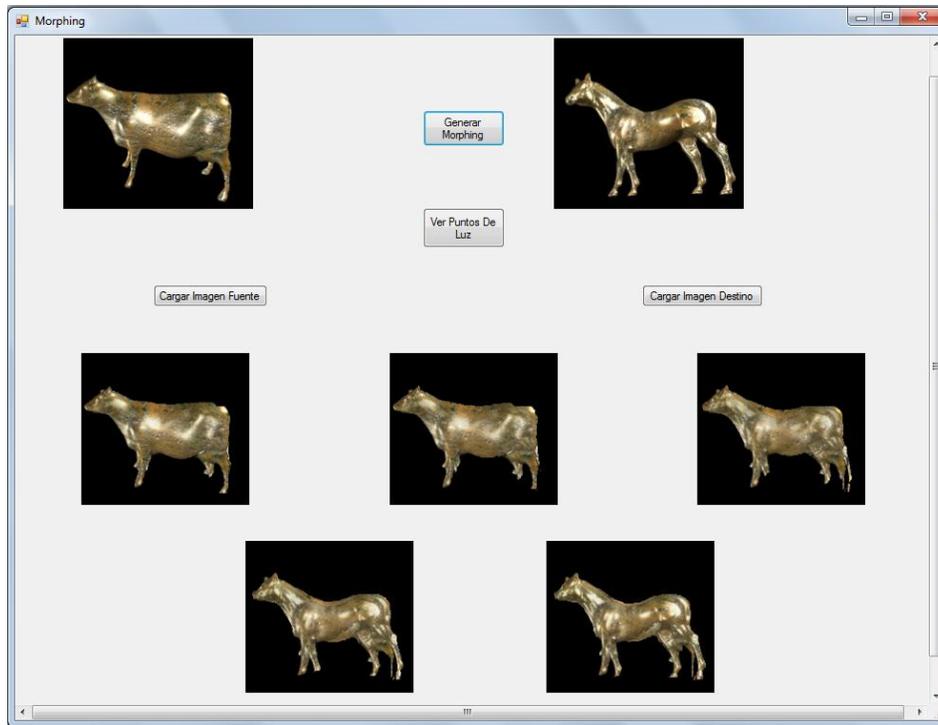


Generar Morphing

Se selecciona el botón “Generar Morphing”.

Una vez presionado, se empieza el procesamiento de las imágenes una vez terminado se muestran 5 imágenes con la transformación progresiva de las imágenes y aparece el botón “Mostrar Puntos de Luz”.

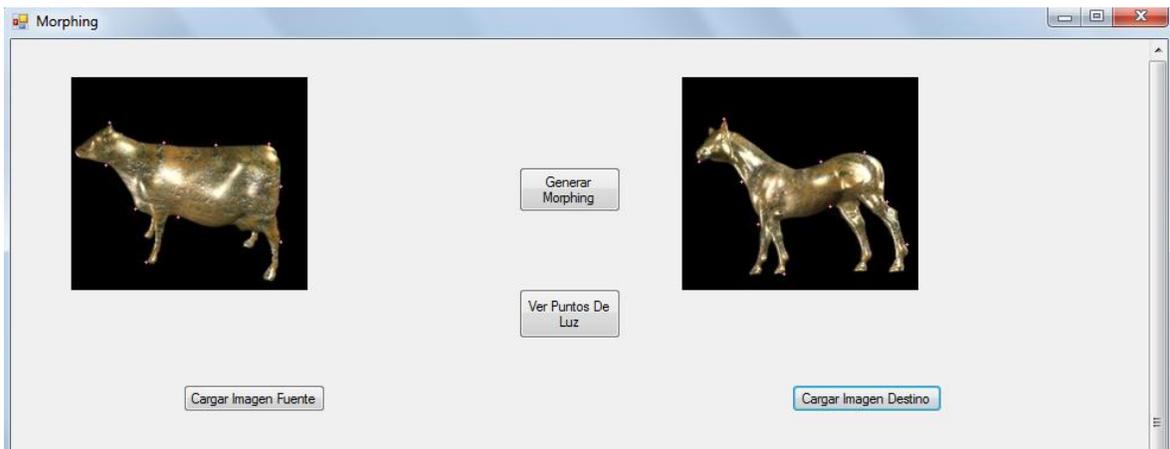
Una vez procesadas las imágenes se muestra la siguiente pantalla.



Ver Puntos de Luz

Se selecciona el botón "Ver Puntos de Luz".

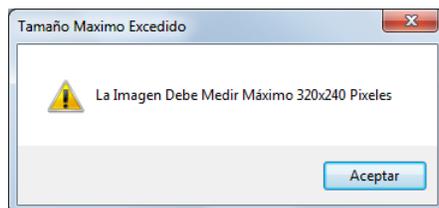
Al presionar este botón se muestran los puntos de luz utilizados como referencia para hacer la transformación de las imágenes, estos puntos se muestran marcado en las imágenes originales que fueron cargadas, estos puntos se muestran en un color rosa intenso para poderlos distinguir del resto de la imagen.



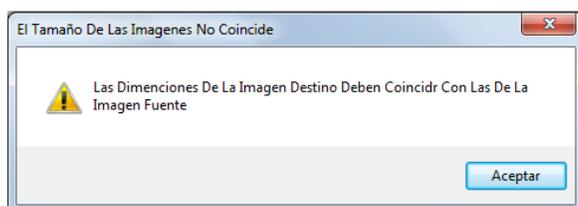
Errores en las Imágenes

Puede ser que el programa detecte algunos problemas al momento de cargar las imágenes, estos problemas son:

El tamaño de la imagen a ingresar excede los pixeles permitidos:



Las dimensiones de la imagen destino no coinciden con la imagen fuente:



Notas

Para un mejor funcionamiento del programa es recomendable que las imágenes que se ingresen tengan un fondo oscuro y la figura a transformar resalte de dicho fondo.

La visualización de las imágenes intermedias se realiza de izquierda a derecha y de arriba a abajo, de esta manera se puede apreciar el cambio gradual que va sufriendo la imagen.

La transformación del color en las imágenes se realizó de igual manera mediante una interpolación entre los colores de ambas imágenes de cada pixel correspondiente.

Recursos

Este proyecto se realizó en una Laptop Compaq Presario CQ40, con un Sistema Operativo Windows 7 Home Basic de 32 bits, con Procesador AMD Athlon X2 Dual-Core y memoria RAM de 3.00 GB.

Para el desarrollo del proyecto se utilizó el IDE Visual Studio 2008 Professional Express Edition, usando como lenguaje C++ y las bibliotecas de .NET para trabajar con imágenes.

Posible Continuación

Algunas propuestas como posible continuación para este proyecto son:

- Incluir el número de imágenes intermedias de la visualización del Morphing.
- Guardar las imágenes producidas y generar un video que muestre la transformación.
- Mejorar el acomodo e interpolación de los colores en las imágenes.
- Optimizar el código y las funciones que fueron usadas.
- Autoescalar las imágenes cuando éstas sean de diferentes dimensiones.
- Paralelizar el algoritmo para evitar los problemas de tiempo de procesamiento.

Conclusiones

Durante la realización de este proyecto comprendí cómo es que se realiza el procesamiento digital de imágenes de una forma básica.

Algunos de los problemas con los que me enfrenté durante la realización de este proyecto fue el no poder seleccionar los puntos de luz únicamente del contorno de la imagen, cosa que fue solucionada al implementar el algoritmo de Canny Edge.

Al estar asignando la interpolación de los colores a las imágenes intermedias al hacer esta interpolación parecía que solo estaba manipulando el canal alfa para hacer la transformación, es decir, parecía que solo estaba manipulando la transparencia del color, esto se logro solucionar condicionando a que si el color de alguna de las dos imágenes era muy bajo, tomara el color completamente de la otra imagen, de esta manera el Morphing se aprecia con una mejor calidad.

Respecto a los objetivos propuestos en un principio no fue necesario la implementación del módulo para generar polígonos, ya que éste se sustituyó al poder obtener los contornos con el algoritmo de Canny Edge.

En cuanto a los demás objetivos estos fueron realizados de manera satisfactoria, el objetivo principal se vió alcanzado una vez que se compararon las imágenes obtenidas en este proyecto con las imágenes obtenidas en un proyecto similar.

Bibliografía

Tutorial de implementación del Algoritmo de Canny Edge

http://dasl.mem.drexel.edu/alumni/bGreen/www.pages.drexel.edu/_weg22/can_tut.html

Biblioteca de .NET

<http://msdn.microsoft.com/library/60k1461a>

Morphing

<http://www.cg.tuwien.ac.at/hostings/cescg/CESCG97/penkler/morphing.htm>

T. Beier and S. Neely. "Feature-based image metamorphosis", *Computer Graphics Proceedings of SIGGRAPH 92*, 26(2):35–42, July 1992 [En Línea]. Disponible: <http://portal.acm.org/>

Z. Zhang, L. Wang, B. Gou y H. Shum, "Feature-Based light field morphing", *Proceedings of ACM SIGGRAPH 2002 Vol. 21* pp. 457 - 464, Julio 2002 [En Línea]. Disponible: <http://portal.acm.org/>

R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison-Wesley, 2a. edición, 2002.

A. Vega, "Reconstrucción 3D de objetos reales mediante adquisición calibrada de su contorno", Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2007.

I. Vázquez, "Modificación del algoritmo de trazado de rayos para simular deformaciones en objetos 3D y generar ambientes surrealistas". Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, D.F., México, 2010.

Morpheus Photo Morpher [En Línea]. Disponible: <http://www.morpheussoftware.net>