

Morphing

Joaquín Pérez Araya
Departamento de Ciencias de la Computación
Universidad de Chile
Santiago, Chile
joaquin.perez.a@ug.uchile.cl

Resumen—

INTRODUCCIÓN

El morphing es el efecto visual el cual se produce al cambiar una imagen a otra con un efecto de metamorfosis, actualmente se utiliza principalmente para el entretenimiento. En este documento se implementará el algoritmo descrito por Beier-Neely, que consiste en utilizar líneas de correspondencia entre la imagen de partida y la imagen de destino para describir la forma en que la mutación se va a llevar a cabo. Primero se describirá el diseño

DISEÑO E IMPLEMENTACIÓN

El código implementado se define en 3 funciones las cuales están implementadas en `morphing.py`: `warp`, `morph` y `create morphing video`: La transformación de una imagen según un par de conjuntos de líneas (`warp`), la creación de múltiples imágenes que son las que forman parte del morphing (`morph`) y finalmente la creación, a partir de las imágenes calculadas en el proceso anterior el vídeo que muestra el cambio de las imágenes a lo largo del tiempo `create morphing video`. También se dispone de un módulo adicional llamado `util.py` el cual contiene funciones auxiliares.

Warping

Para el warping se utiliza el algoritmo propuesto por (paper), que está especificado en el Algoritmo 1, el cual crea una imagen vacía con las mismas dimensiones de la imagen fuente, y itera sobre ésta, calculando qué colores debería tener según las líneas de correspondencia y la imagen fuente.

Para los parámetros del peso se eligió: $A = 1, B = 1, p = 0,5$. En el módulo de utilidades, están implementadas las funciones para el cálculo de u, v, X' y $weight$ según como se indica en el artículo.

La implementación usa el algoritmo en su versión *inverse mapping*, es decir durante la ejecución se recorre la imagen de destino calculando qué píxeles de la imagen original deberían estar allí utilizando interpolación bilineal de los píxeles más cercanos de la imagen original, este método ofrece más simplicidad dado que se conoce los píxeles de destino de antemano por lo que el único inconveniente es el caso de que se requieren píxeles fuera de la imagen original por lo que se interpola según los píxeles ya calculados anteriormente, en cuyo caso sería:

Algoritmo 1: Warpping

Data: *image* imagen fuente, $lines_{src}$ líneas fuente, $lines_{dst}$ líneas de destino.

Result: Imagen que corresponde a la imagen fuente modificada según las líneas dadas.

$destinationImage = zeros(shape(image))$

for Pixel X in *image* **do**

$DSUM = (0,0)$

$weightsum = 0$

for Line P_iQ_i in $lines_{src}$ and $P'_iQ'_i$ in $lines_{dst}$ **do**

$u, v = calculate_u_v(X, P_i, Q_i)$

$X' = calculate_X'(u, v, P'_i, Q'_i)$

$D_i = X'_i - X$

$weight = calculate_weight(X, u, v, P_i, Q_i)$

$DSUM += D_i * weight$

$weightsum += weight$

$\bar{X} = X + DSUM/weightsum$

$destinationImage(X) = image(\bar{X})$

return *destinationImage*

- Si se está en el primer pixel de la imagen, el de la esquina superior izquierda, éste se calcula como el pixel del mismo punto de la imagen de origen.
- Si se está en la fila superior, se calcula usando el pixel anterior calculado, el de la izquierda de éste.
- Si se está en la columna de la izquierda, se calcula usando el pixel de la fila superior.
- Si se está en la columna de la derecha, se calcula utilizando los 3 píxeles que están cercanos a éste: Superior izquierdo, superior e izquierdo.
- De otra forma se calcula utilizando 4 píxeles cercanos: Superior izquierdo, superior, superior derecho e izquierdo.

Morphing y Video

Para la creación del conjunto de imágenes que forman el Morphing completo se realiza lo siguiente: se itera por el número de imágenes n que uno quiere incluir en el vídeo, se calcula el valor de entrelazado t que va desde 0 hasta 1 de $\frac{1}{n}$ en $\frac{1}{n}$ pasos, según el grado de avance del Morphing.

Dado un t de éste proceso, para crear una imagen, se realiza un cross-fade entre dos imágenes: el `warp` de la imagen fuente con sus líneas y la interpolación de las líneas fuente hacia las

líneas de destino según t , es decir $t \cdot \text{lines}_{src} + (1-t) \cdot \text{lines}_{dst}$, y el *warp* entre la imagen de destino con sus líneas y la interpolación de las líneas de destino hacia las fuentes según t , que es $(1-t) \cdot \text{lines}_{src} + t \cdot \text{lines}_{dst}$. Al final se añade a la colección de imágenes la ponderación de t por la primera imagen más $1-t$ de la segunda.

Luego de finalizado la creación de la colección de imágenes del morphing, se utiliza la librería de OPENCV para formar un vídeo con éstas, en esta implementación, los vídeos generados están a 10 cuadros por segundo y en formato .avi.

EXPERIMENTACIÓN

Se realizaron pruebas dos pares de imágenes: Un par con dos felinos distintos, un puma y una leona, y el otro par con dos actores famosos. Para par de imágenes se crearon 3 archivos con líneas, uno con líneas de contorno (véase las líneas verdes en la figura a continuación), uno con líneas de expresión (las líneas azules en la figura) y por último uno con ambas líneas en conjunto.



Figura 1. Imágenes usadas para la experimentación, están marcadas las líneas utilizadas para el Morphing, las verdes corresponden a contornos de la figura mientras que las azules corresponden a características intrínsecas de la imagen. Las imágenes de la izquierda corresponden a las de fuente mientras que las de la derecha corresponden a las de destino.

Para cada prueba, se ejecutó la implementación requiriendo 50 imágenes en total, de las cuales sólo se presentarán la 1^{ra}, 10^{ma}, 20^{va}, 30^{va}, 40^{va} y 50^{va} imagen de dichas ejecuciones.

Imágenes de Felinos

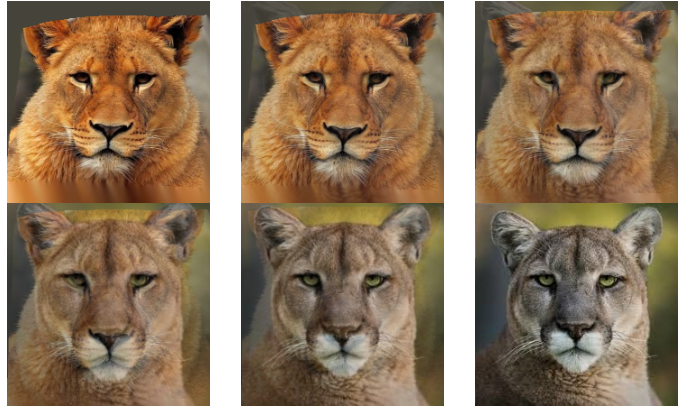


Figura 2. Imágenes creadas con la implementación y con las líneas verdes. El progreso va de izquierda a derecha de arriba hacia abajo.

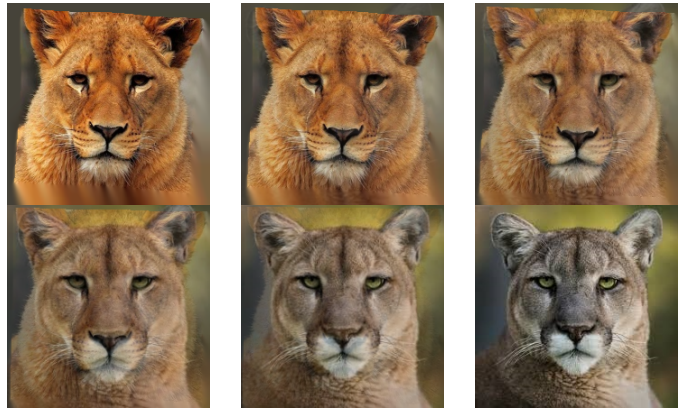


Figura 3. Imágenes creadas con la implementación y con las líneas azules. El progreso va de izquierda a derecha de arriba hacia abajo.

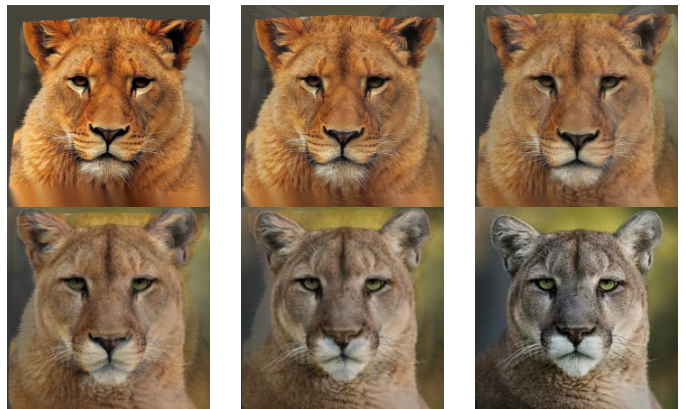


Figura 4. Imágenes creadas con la implementación y con ambos pares de líneas. El progreso va de izquierda a derecha de arriba hacia abajo.

La técnica implementada de Morphing, es bastante lenta, dado que su tiempo de ejecución aumenta según la cantidad de píxeles, la cantidad de líneas determinadas y la cantidad de imágenes totales que se quieren, sin embargo esto otorga más control a la hora de elegir la forma de cómo se va animando la imagen a lo largo del morphing completo.

REFERENCIAS

[1]



Figura 5. Imágenes creadas con la implementación y con ambos pares de líneas. El progreso va de izquierda a derecha de arriba hacia abajo.



Figura 6. Imágenes creadas con la implementación y con ambos pares de líneas. El progreso va de izquierda a derecha de arriba hacia abajo.



Figura 7. Imágenes creadas con la implementación y con ambos pares de líneas. El progreso va de izquierda a derecha de arriba hacia abajo.