

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`: `wrap`, `morph` y `create morphing video`: La transformación de una imagen según un par de conjuntos de líneas (`wrap`), 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.

Wrapping

Para el wrapping se utiliza el algoritmo propuesto por (paper), que estp:

Algoritmo 1: Histograma de Orientaciones Simple

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 in image do
    DSUM = (0,0)
    weightsum = 0
    for Line  $P_iQ_i$  in linessrc and  $P'_iQ'_i$  in linesdst do
        if angle < 0 then
            angle  $\leftarrow$  angle +  $\pi$ 
        end if
        DSUM = DSUM + Line * weightsum
    end for
    return h
```

En el módulo de utilidades, están implementadas las funciones para el cálculo de u , v , X'

La implementación usa el algoritmo en su versión de mapeo inverso, 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:

- 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

Video

EXPERIMENTACIÓN

Se realizaron pruebas con los siguientes valores para medir los pesos

CONCLUSIÓN

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]