

# MORPHING DE IMAGENS

SARA DO NASCIMENTO<sup>1</sup>  
 SONJA AVARESE LEITE MEERBAUM<sup>2</sup>  
 JOSÉ AUGUSTO PEREIRA BRITO<sup>3</sup>

Laboratório de Computação Gráfica / COPPE / UFRJ  
 C.P. 68511  
 21945-970 Rio de Janeiro RJ  
<sup>1</sup>sara@lcg.ufrj.br  
<sup>2</sup>sonja@lcg.ufrj.br  
<sup>3</sup>brito@lcg.ufrj.br

**Abstract.** This paper presents the LCGMORPH program, which is an implementation of an algorithm that uses Delaunay triangulation to generate image metamorphosis, also called morphing.

## 1 - Introdução

Neste trabalho apresenta-se os aspectos teóricos, de elaboração e implementação computacional de um procedimento para efetuar o morphing de duas imagens, com base no algoritmo apresentado no item 2. O programa implementado é denominado "LCGMORPH". Alguns exemplos são apresentados no item 3.

## 2 - Algoritmo

Como entrada, faz-se uso de duas imagens, uma inicial e outra final, a partir das quais obter-se-á as transformações intermediárias.

Usa-se um conjunto de pares de pontos para controlar a transformação da geometria, onde cada par de pontos possui um ponto na imagem inicial e um ponto na imagem final.

Seja :

$A_i$ : Pontos de controle marcados pelo usuário na imagem inicial;

$B_i$ : Pontos de controle marcados pelo usuário na imagem final;

M: Número de pares de pontos;

N: Número de quadros a serem gerados na transformação .

Para cada  $i=1, \dots, N-1$ , fazer:

1 - Determinar as coordenadas dos pontos nas imagens intermediárias, por interpolação linear, segundo a expressão:

$$C_{ij} = \frac{i}{N} B_j + \frac{N-i}{N} A_j, \quad j=1, \dots, M$$

2 - Obter a triangulação de Delaunay  $T_i$  dos pontos  $C_{ij}$ ,  $j=1, \dots, M$

3 - Para cada imagem intermediária  $I_i$ , determinar a cor de cada pixel  $p$  da seguinte forma:

a) Sejam  $C_{ik}$ ,  $C_{il}$  e  $C_{im}$  os vértices de um triângulo e  $\alpha_k$ ,  $\alpha_l$ ,  $\alpha_m$  as coordenadas baricêntricas de  $p$  em relação a esses vértices. Na estrutura de dados, percorrer cada um dos triângulos, varrendo cada um de seus pixels.

b) Interpolarmos a cor de cada pixel  $p$  de cada triângulo da imagem intermediária, através da equação abaixo:

$$q(p) = \sum_{s \in (k,l,m)} \alpha_s \left( \frac{i}{N} q_f(B) + \frac{N-i}{N} q_0(A) \right)$$

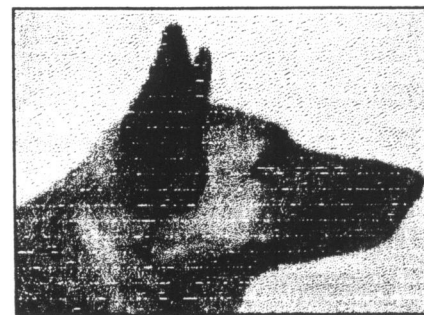
onde  $q_0(\cdot)$  e  $q_f(\cdot)$  representam o valor do atributo de cor  $q(\cdot)$  nos pixels das imagens inicial e final, respectivamente.

4 - Armazenar a imagem interpolada  $I_i$  em arquivo.

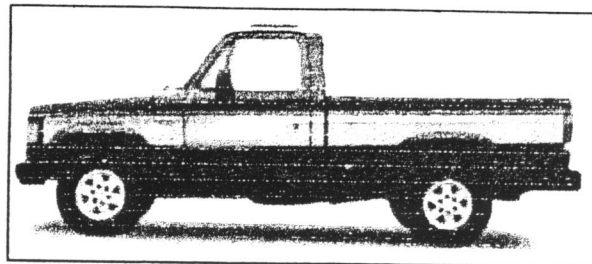
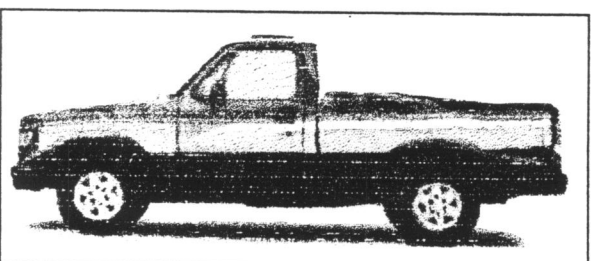
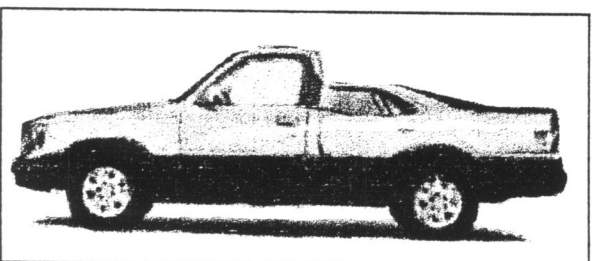
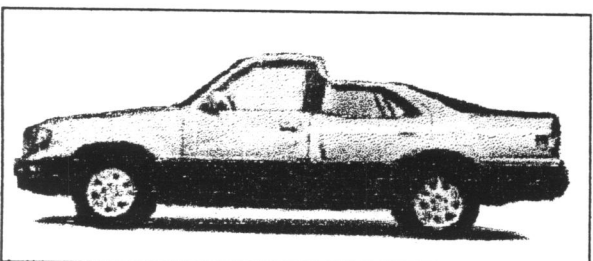
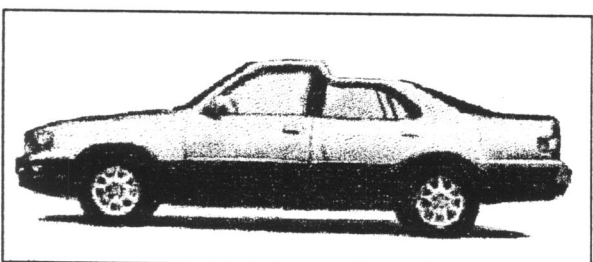
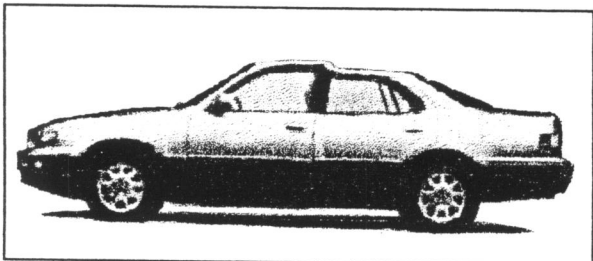
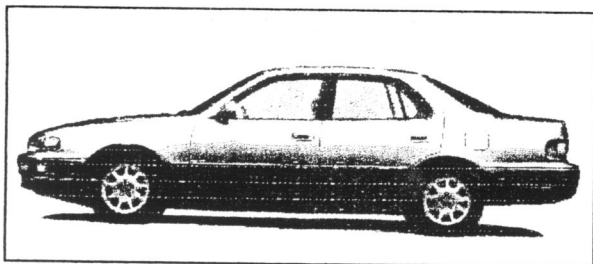
A complexidade da triangulação de Delaunay com abordagem "Divide-and-Conquer" é da ordem  $O(M \cdot \log M)$ , onde  $M$  é o número de pontos introduzidos pelo usuário. No caso da varredura da imagem para cada quadro, a complexidade é linear  $O(n)$ , onde  $n$  é o número de pixels da imagem. Portanto, a complexidade deste algoritmo é da ordem  $O(N \cdot n + N(M \cdot \log M))$ .

### 3- Exemplos de Transformações do Tipo Morphing

Apresenta-se, a seguir, dois exemplos obtidos através do programa LCGMORPH. O primeiro exemplo mostra a transformação de um homem em um cão.



O segundo exemplo apresenta a transformação de um automóvel em uma caminhonete pickup:



#### 4 - Descrição do Programa

O programa LCGMORPH foi desenvolvido em linguagem C, em estação gráfica Sun SPARCStation2, utilizando-se o sistema operacional UNIX. As rotinas gráficas, de manipulação e de aquisição de eventos foram geradas pelo "software" XWindow. A interface gráfica do programa foi implementada com o gerador interativo GUIDE. A figura do anexo mostra a tela principal do programa. A estrutura de dados usada na implementação do programa LCGMORPH para a triangulação de Delaunay é do tipo Half-Edge [Mantyla (1988)].

#### 5- Conclusões

O programa LCGMORPH foi implementado para processar somente imagens do tipo sunraster com tonalidades de cinza. Para que se possa fazer transformações com imagens coloridas, as alterações no programa serão mínimas.

O tempo de processamento é proporcional ao tamanho das imagens e ao número de quadros.

Observa-se que o número de pares de pontos usados na triangulação não tem influência significativa no tempo de processamento. Para que se tenha um processamento mais rápido, é conveniente se usar imagens de tamanho reduzido e/ou diminuir o número de quadros.

A complexidade do algoritmo pode ser reduzida para  $O(N.n + M.\log M)$ , ou seja, tornando-se linear e a triangulação independente do número de quadros. A

deste novo algoritmo é fazer a triangulação de Delaunay somente uma vez sobre o primeiro quadro, sendo que nos outros quadros, a triangulação será local, caso algum triângulo não seja mais de Delaunay.

#### 6- Agradecimentos

Este trabalho foi possível graças ao auxílio financeiro da CAPES e do CNPq, através da concessão de bolsas de pesquisa. Os autores agradecem aos colegas do LCG pelo apoio e sugestões e, ao Professor Antônio Oliveira, orientador deste trabalho.

#### 7 - Referências Bibliográficas

M. Mäntylä, *An introduction to solid modeling*, Computer Science Press, 1988.5

G. Wolberg, *Digital image warping*, IEEE Computer Society Press Monograph, 1990.

F. Preparata & M. Shamos, *Computational Geometry: an introduction*, P Springer Verlag, 1988.

A. A. F. de Oliveira, *Notas de aula da Disciplina Geometria Computacional - Engenharia de Sistemas e Computação - COPPE -UFRJ*, 1993.

L. Guibas, J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics*, Vol 4, N. 2, April 1985.

L. C. Serpa, *Apostila de XWindow*, UFRJ 1993.

#### 8 - Anexo

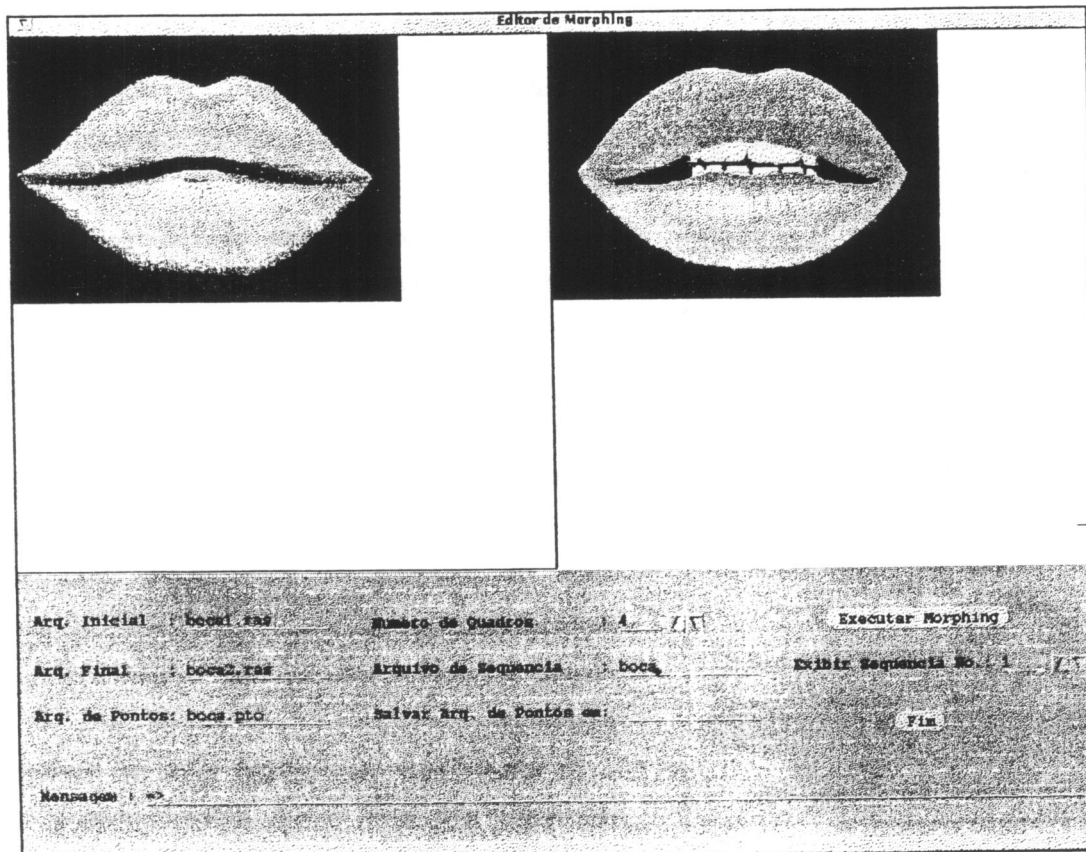


Figura 1: Tela Principal do Programa LCGMORPH