

# Fragment-Based Image Completion

Guilherme Schirmer de Souza  
Ivan Monteiro de Castro Conti  
José Luiz Gomes Júnior

Baseado no trabalho:

*Fragment-Based Image Completion, de*

Iddo Drori, Daniel Cohen-Or e Hezy Yeshurun  
School of Computer Science, Tel Aviv University

# 1. Introdução

## 1.1 Definição do problema

Dada uma imagem e selecionado um objeto desta a ser retirado, o objetivo é completar a região desconhecida basendo-se em elementos da parte conhecida.

## 1.2 O processo utilizado

Usaremos um processo de interpolação suave com captura de fragmentos da própria imagem. O preenchimento da região desconhecida é feito pela inserção desses fragmentos.

É um processo de composição que requer um conjunto amostral com informações bastante relevantes e um relativo grau de semelhança entre os elementos da imagem.

É um método 2D, não levando em considerando informações de profundidade.

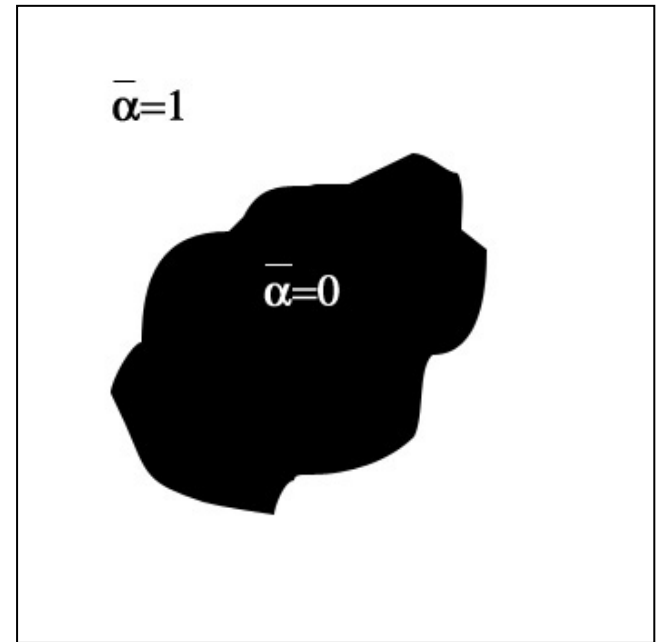


## 2. Visão Geral

### 2.1 Determinando o objeto a ser retirado

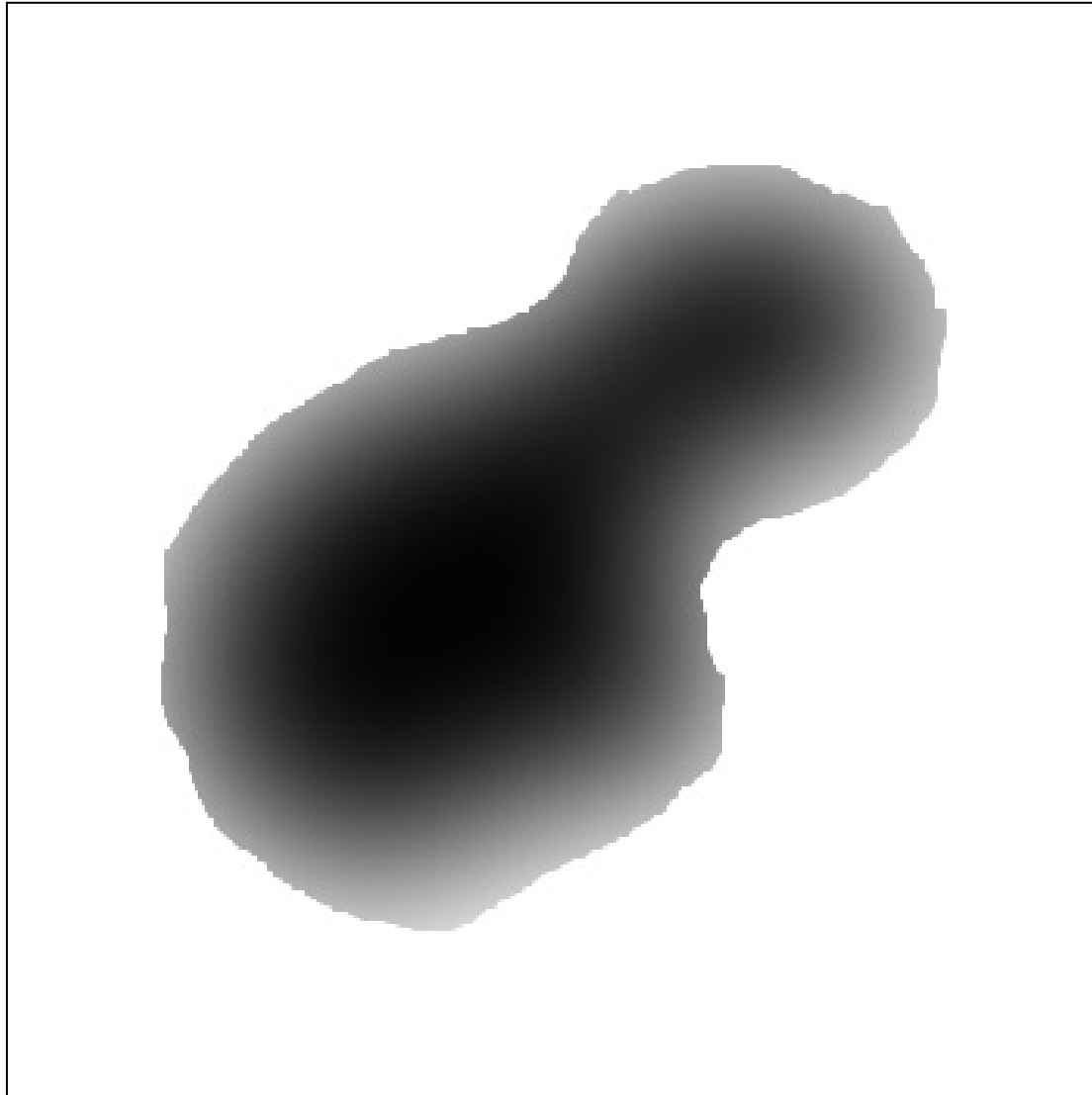
A imagem inicial é marcada através de uma ferramenta de edição de imagem para demarcar a região desejada, criando-se o “canal alpha”. Definiu-se então:

$\bar{\alpha} = 1$	Região conhecida
$0 < \bar{\alpha} < 1$	Região parcialmente conhecida
$\bar{\alpha} = 0$	Região desconhecida



### 2.2 O procedimento

Gera-se uma aproximação inicial através de uma suavização simples da imagem na parte sem informação, a partir de sucessivas filtragens (até a convergência). Assim, os pixels de baixa confiança concordam grosseiramente com a vizinhança, de maior confiança.



Mapa de confiança, em escala de cinza

À aproximação são acrescentados detalhes de um fragmento de confiança alta, no decorrer do processo.

Um fragmento é definido como uma vizinhança quadrada em torno de um pixel. O tamanho da vizinhança é adaptativo.



Escala: aproximação "coarse-to-fine"

Cada passo: detalhes de um fragmento origem (source) de maior confiança são adicionados a um fragmento alvo (target).

A procura do 'source fragment' é realizada de modo a formar uma região coerente com a imagem.

Atualiza-se então os valores de confiança, de modo que a confiança média tenda a 1.

**Input:** image  $C$ , inverse matte  $\tilde{\alpha}$  ( $\exists$  pixel with  $\tilde{\alpha} < 1$ )

**Output:** completed image,  $\tilde{\alpha} = 1$

**Algorithm:**

for each scale from coarse to fine

    approximate image from color and coatset scale

    compute confidence map from  $\tilde{\alpha}$  and coarser scale

    compute level set from confidence map

    while mean confidence  $< 1 - \epsilon$

        for next target position  $p$

            compute adaptive neighborhood  $N(p)$

            search for most similar and frequent source match  $N(q)$

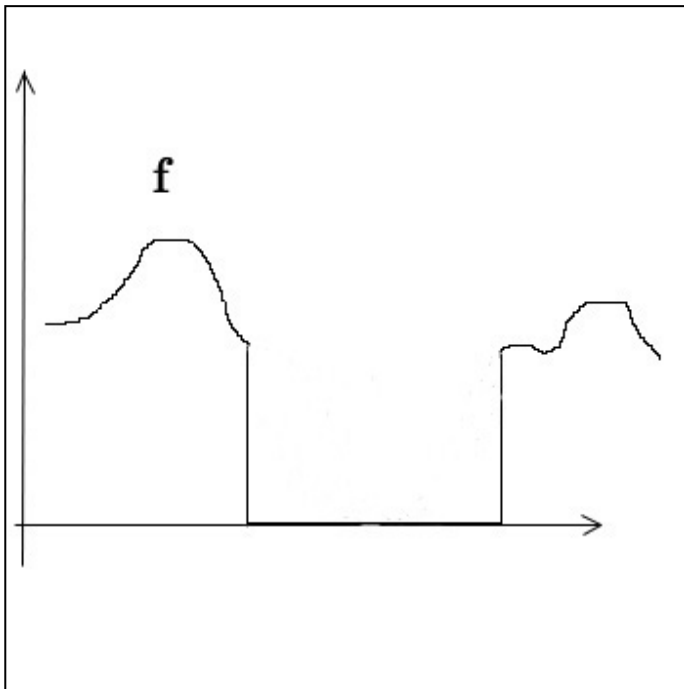
            composite  $N(p)$  and  $N(q)$  at  $p$ , updating color and  $\tilde{\alpha}$

            compute approximation, confidence map and update level set

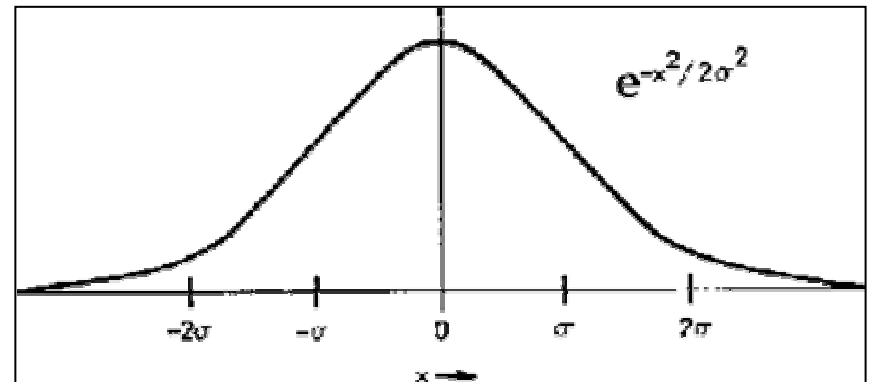
### 3. Aproximação Inicial

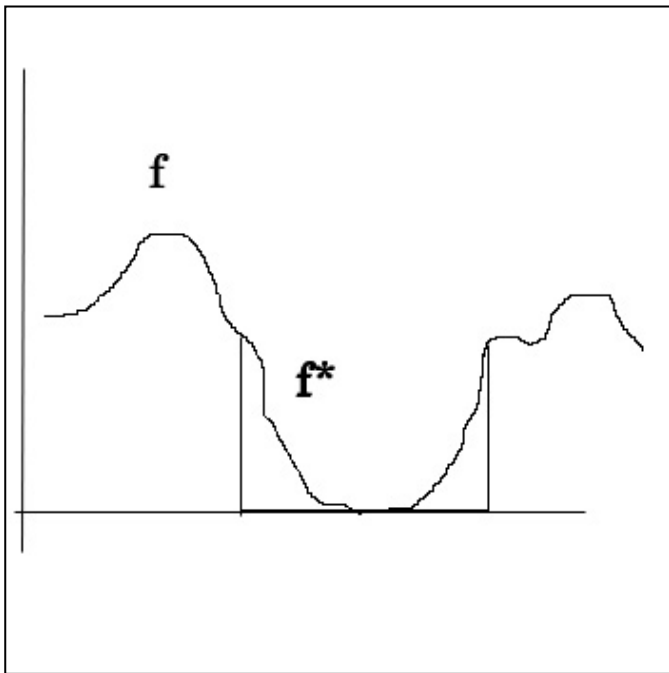
Um processo de filtragem iterativa gera uma aproximação inicial das cores na região desconhecida. É um processo de interpolação de dados.

Problema: dada uma imagem  $f$ , com um conjunto de pontos conhecidos, criar uma estimativa suave de  $f$  nos pontos desconhecidos da imagem.



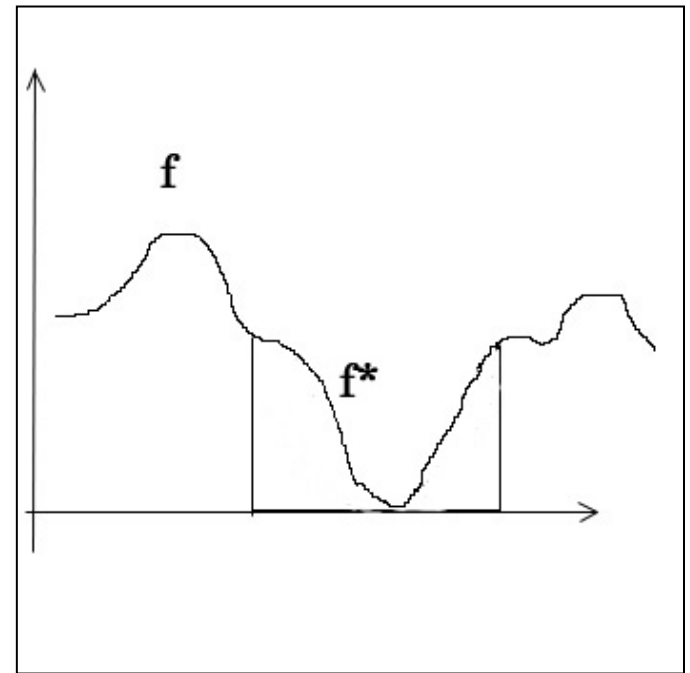
Utilizando filtragem gaussiana:





k-ésima iteração

$k < n$



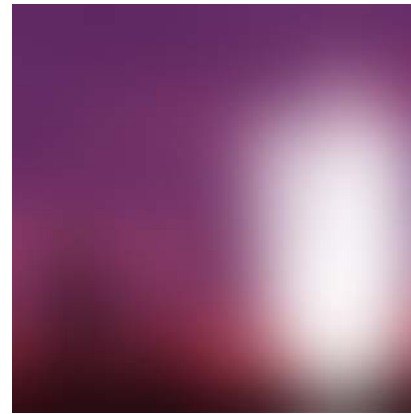
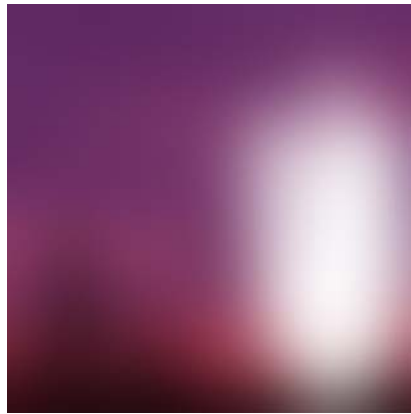
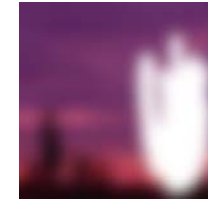
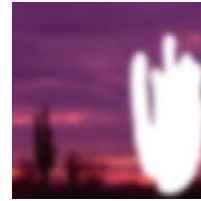
n-ésima iteração

Método utilizado: push-pull method (convolução gaussiana em multi-resolução)

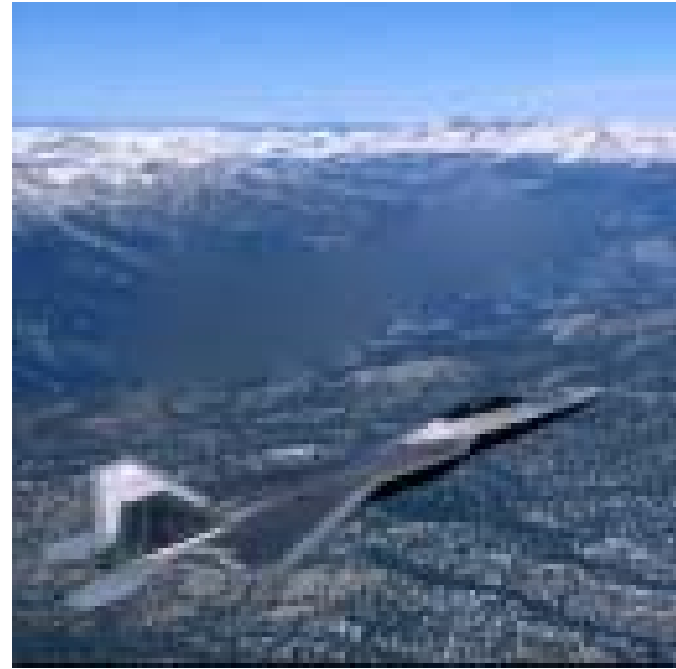
$$\bar{C} = C\bar{\alpha}$$

$$Y_{t+1}^l = (Y_t^l \alpha + \bar{C})(*K_\varepsilon \downarrow)^l (\uparrow *K_\varepsilon)^l$$





O método é repetido até convergência das cores de uma quantidade  $N$  representativa dos pontos dentro da região. Os pontos são sorteados aleatoriamente.



**Exemplo de resultado da aproximação inicial**

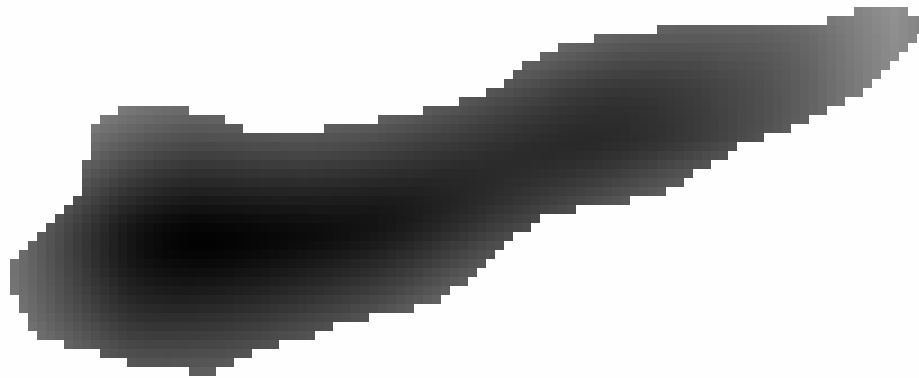
## 4. O Mapa de Confiança

Criação do mapa de confiança  $\beta$

$$\beta_i = 1 \quad \text{se} \quad \bar{\alpha}_i = 1$$

(Pixel  $i$ , gaussiana  $g$  e vizinhança  $N$ )

$$\beta_i = \sum_{j \in N(i)} g_j \bar{\alpha}_j^2 \quad \text{caso contrário}$$



Mostra quanto de confiança um pixel possui

A confiança determina uma ordem para a seleção do 'target fragment' em um conjunto de candidatos.

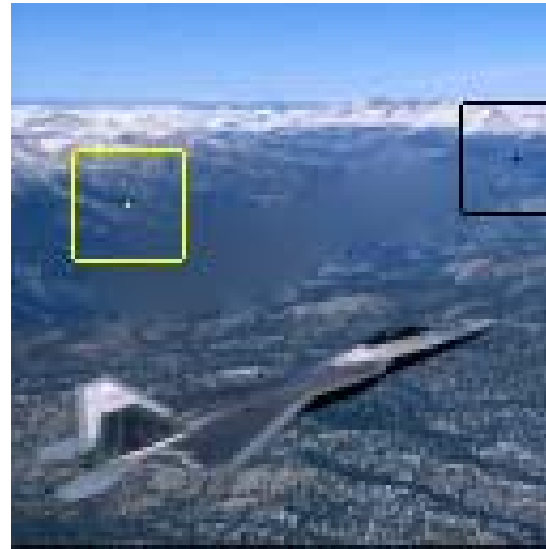
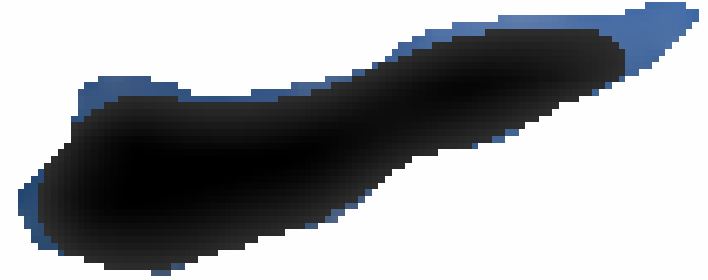
$$v_i \leftarrow 0 \quad \text{se } \beta_i > \mu(\beta)$$

$$v_i \leftarrow \beta_i + \rho(0, \sigma(\beta)) \quad \text{caso contrário}$$

$\rho(0, \sigma(\beta))$  é um ruído aleatório entre 0 e o desvio padrão de  $\beta$

$\mu(\beta)$  é a média do mapa de confiança

Depois da composição, os pontos contidos dentro do fragmento são descartados do conjunto de candidatos.



Quando o conjunto de candidatos estiver vazio, ele é recomputado.

## 5. Procura de Fragmentos

Utiliza um método 2D de “matching” que considera a confiança de cada pixel e, como nos métodos tradicionais, as características semelhantes.

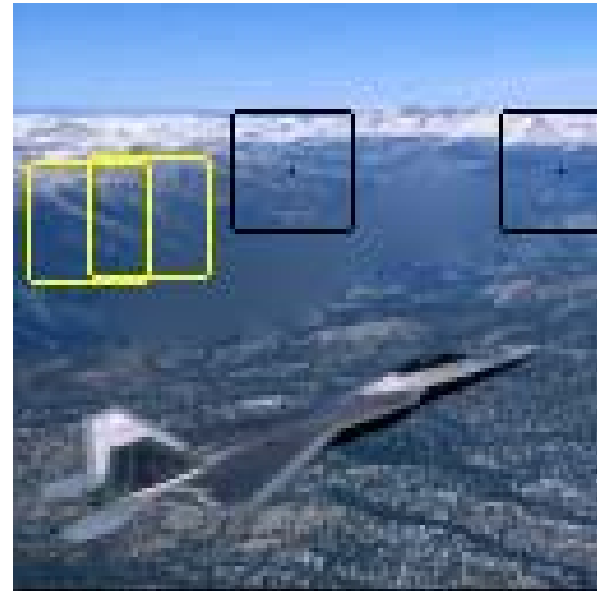
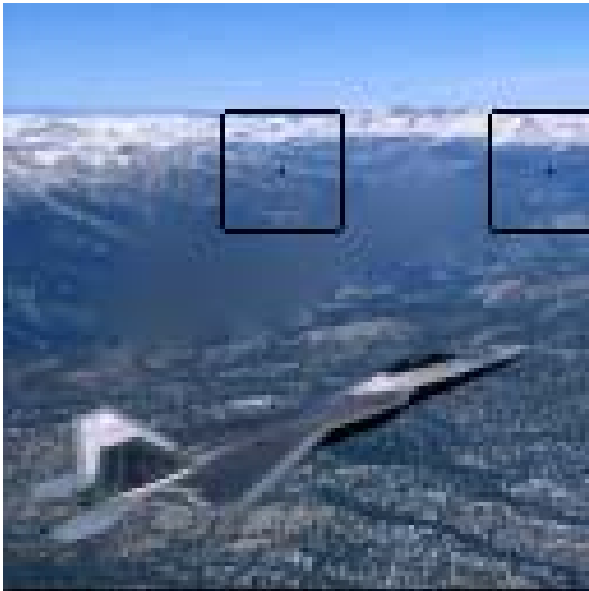
Para cada ‘target fragment’ procura-se o melhor ‘source fragment’, em todas as posições (x,y), em 5 escalas e 4 orientações.

Encontra-se o ‘source fragment’ que minimiza a função:

$$r^* = \arg \min_r \sum_{s=S_r(i), t=T(i), i \in N} (d(s,t)\beta_s\beta_t + (\beta_t - \beta_s)\beta_t)$$

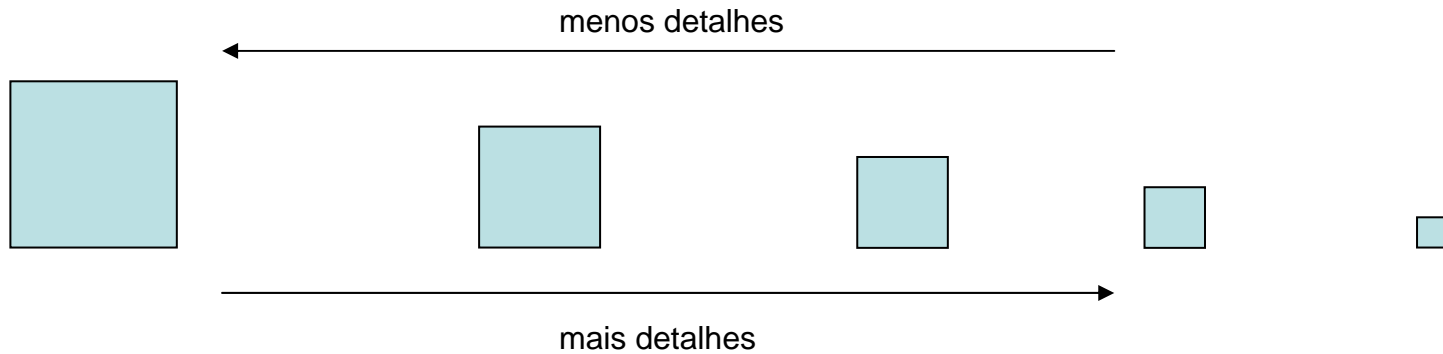
$d(s,t)$  denota a semelhança entre os fragmentos

$\beta_s, \beta_t$  são as confianças do ‘source fragment’ e do ‘target fragment’

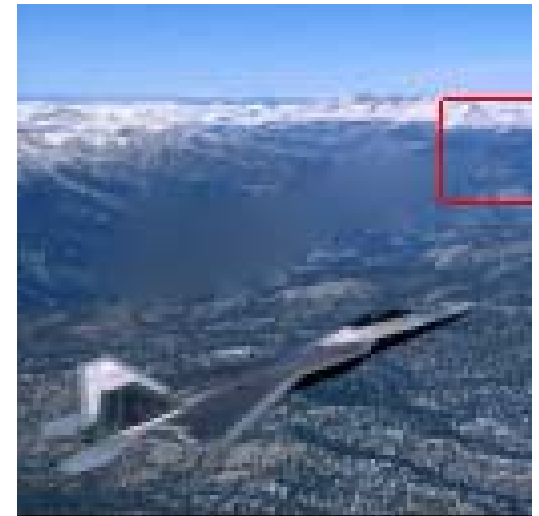
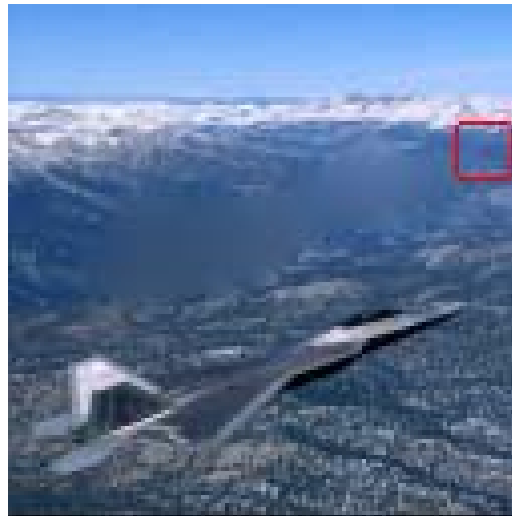
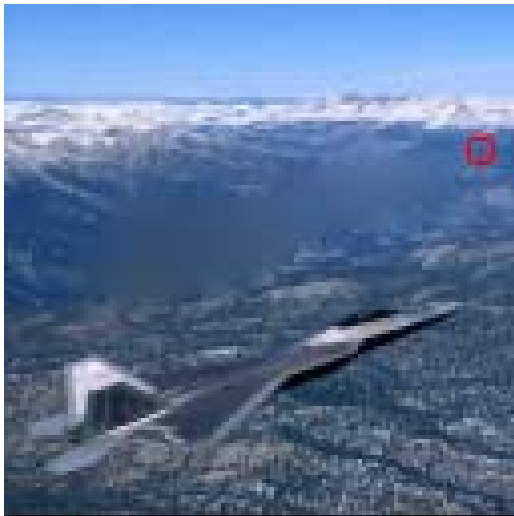


## 5.1 Vizinhaça adaptativa

Um modo adaptativo pra determinar o tamanho da vizinhaça é importante para capturar características em várias escalas.



Critério: maximizar a função: 
$$u = \sum_{i=0}^k (C_i - \bar{C})^2$$



## 6. Composição dos Fragmentos

Prioridade para o 'target fragment', logo realiza-se T OVER S, para criar uma transição entre T e S.

Deve-se levar em consideração os valores de  $\alpha$ . Utiliza-se LAPLACIAN OVER.

$$L(C_{out}) = L(C_T)G(\alpha_T) + L(C_S)G(\alpha_S)G(1 - \alpha_T)$$

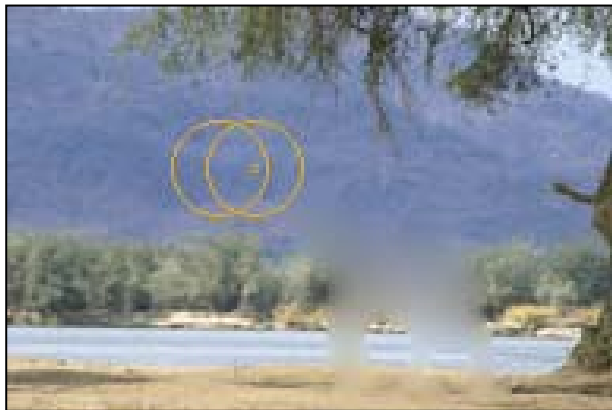
L Operador Laplaciano

G Operador Gaussiano

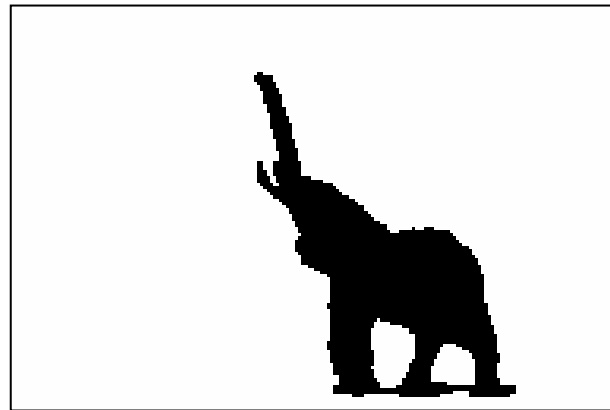
O fragmento de saída é então:  $C_{out} = C\bar{\alpha} + L(C_{out})$

Atualiza-se então os valores de  $\alpha$  do 'fragment source' através de:

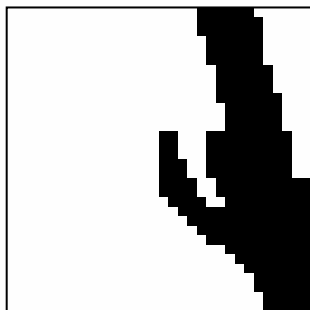
$$\alpha_T = G(\alpha_T) + G(\alpha_S)G(1 - \alpha_T)$$



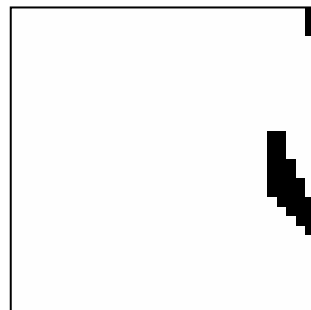
$C$



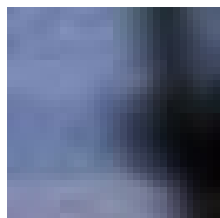
$C \bar{\alpha}$



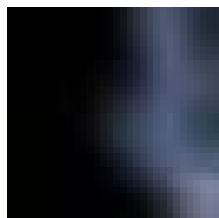
$\alpha_F$



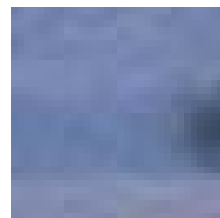
$\alpha_B$



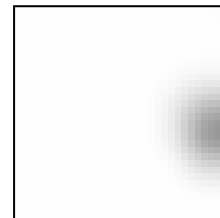
$(t1)$



$(t2)$



$C_{out}$



$\alpha_{out}$



## 7. Implementação

A imagem é completada inserindo-se cada vez mais detalhes.

O resultado de uma escala maior (menos detalhes), o que corresponde a usar um fragmento grande, é atualizado da seguinte forma:

$$f^{l+1} = \lambda f^{l+1} + (1 - \lambda)(C^l * K_\varepsilon)$$

$\Rightarrow \lambda=0.5$  e  $l$  representa uma escala com menos detalhes que  $l + 1$

O mesmo é feito para o mapa de confiança:

$$\beta^{l+1} = \lambda \beta^{l+1} + (1 - \lambda)(\beta^l * K_\varepsilon)$$

## 8. Resultados

