

# Interpolação de Curvas de Nível por Difusão de Calor



Roberto de Beauclair Seixas

Luiz Henrique de Figueiredo

Cláudio Antonio da Silva

{tron,lhf,cads}@visgrafimpa.br



# Motivação

- ✍ Identificação de um método de interpolação que tenha uma qualidade nos resultados visuais e precisão numérica;
- ✍ Implementação simples e eficiente;
- ✍ Projeto do MCT de implantação de um banco de dados de modelos de terreno de todo o Brasil.



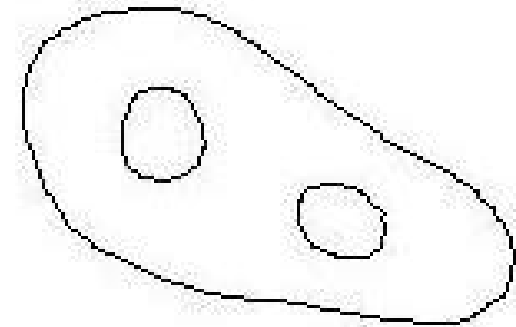
# Objetivo

- ✍ Técnica de interpolação para curvas de nível inspirada em difusão de calor;
- ✍ Comparação com método de interpolação por morfologia;
- ✍ Técnica rápida para a geração de DEMs.



# Técnicas de Interpolação

- ✍ Métodos de interpolação aritméticos
  - ✍ Linear
  - ✍ Inverso da distância
- ✍ Não podem ser usados pois não conseguem resolver nem os casos básicos





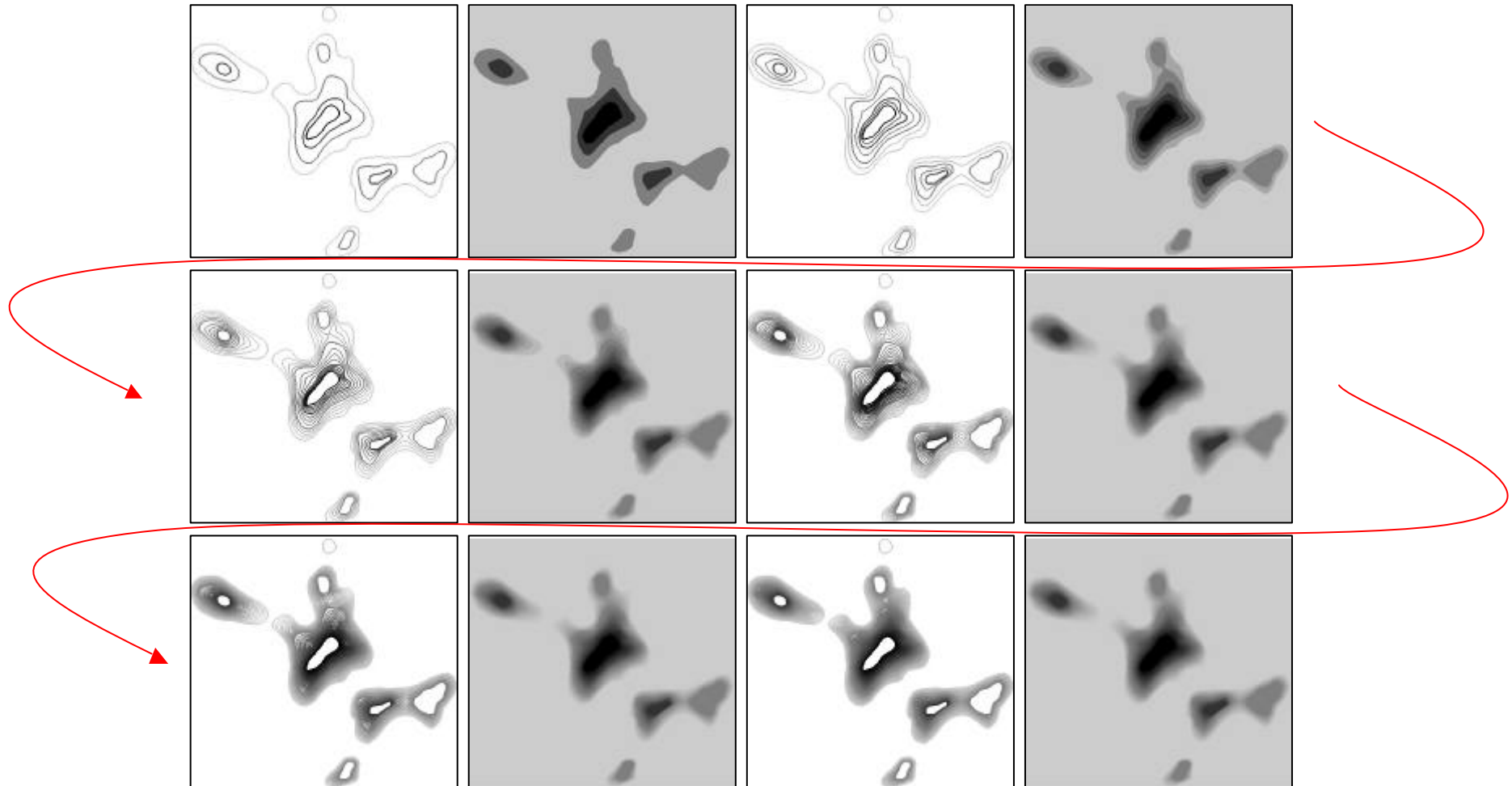
# Técnicas de Interpolação

## ✍ Morfologia Matemática

- ✍ Método que força as curvas a se expandirem “naturalmente”, através de operador de dilatação, até se encontrarem. Nesses pontos, definem-se as novas curvas intermediárias coma altura igual a média das alturas das curvas que se encontraram.



# Morfologia Matemática





# Morfologia Matemática

## ✍ Prós:

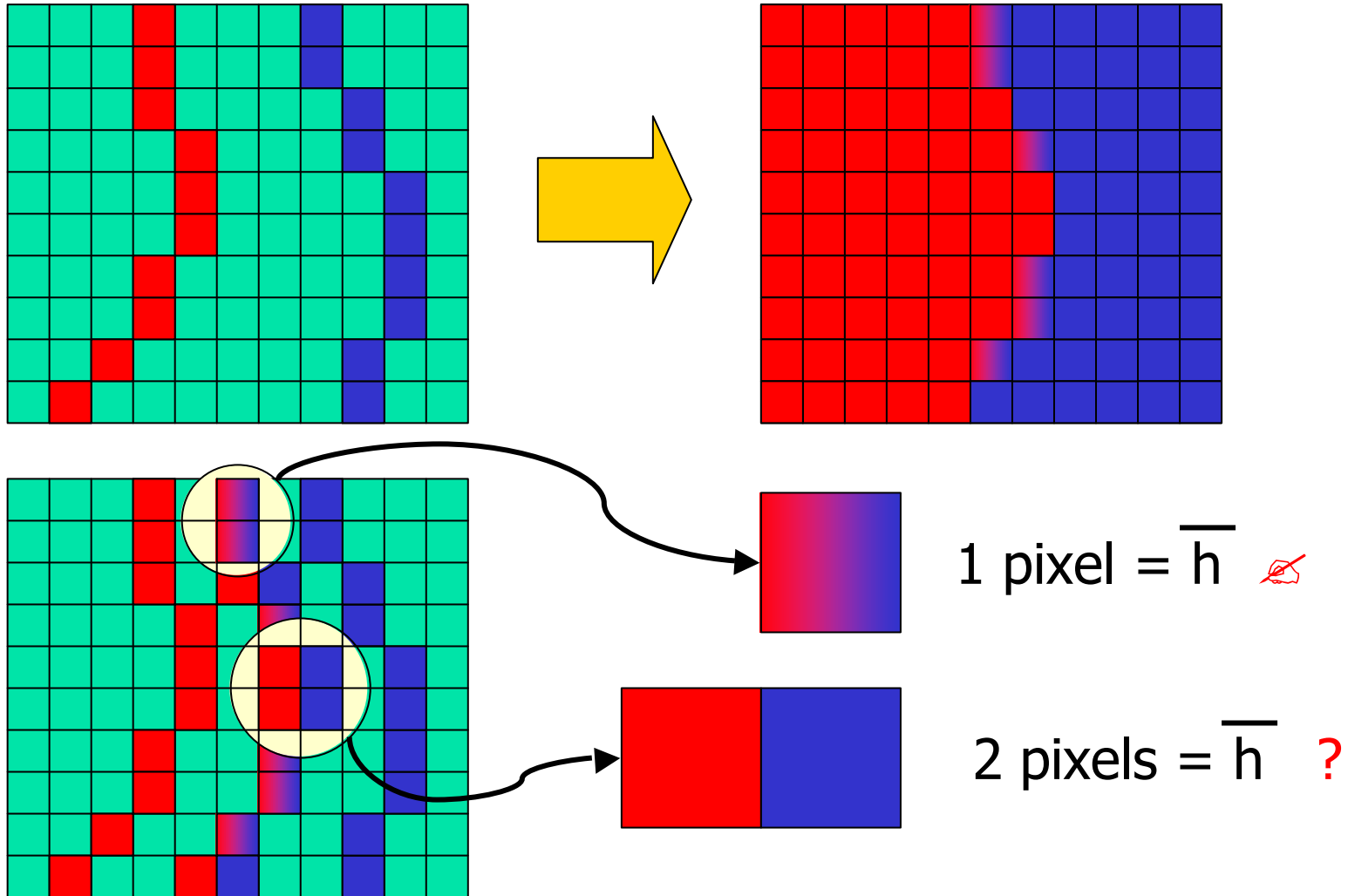
- ✍ Resultados visuais e numéricos são bastante satisfatórios
- ✍ Algoritmo de fácil entendimento

## ✍ Contras:

- ✍ Tempo de processamento é diretamente proporcional à resolução da imagem
- ✍ Implementação com tricks



# Morfologia Matemática: Tricks







# Difusão de Calor

- ✍ Técnica baseada em uma metáfora física inspirada em difusão de calor:
  - 1) A região do terreno que queremos modelar é vista como uma placa de metal, na qual cada curva de nível original é mantida com uma temperatura constante igual a sua cota e o resto da placa está inicialmente à temperatura ambiente;
  - 2) À medida que o tempo passa o resto da placa vai se aquecendo até que se atinja o equilíbrio térmico;
  - 3) No equilíbrio, a temperatura em cada ponto da placa corresponde a altura do terreno.



# Difusão de Calor

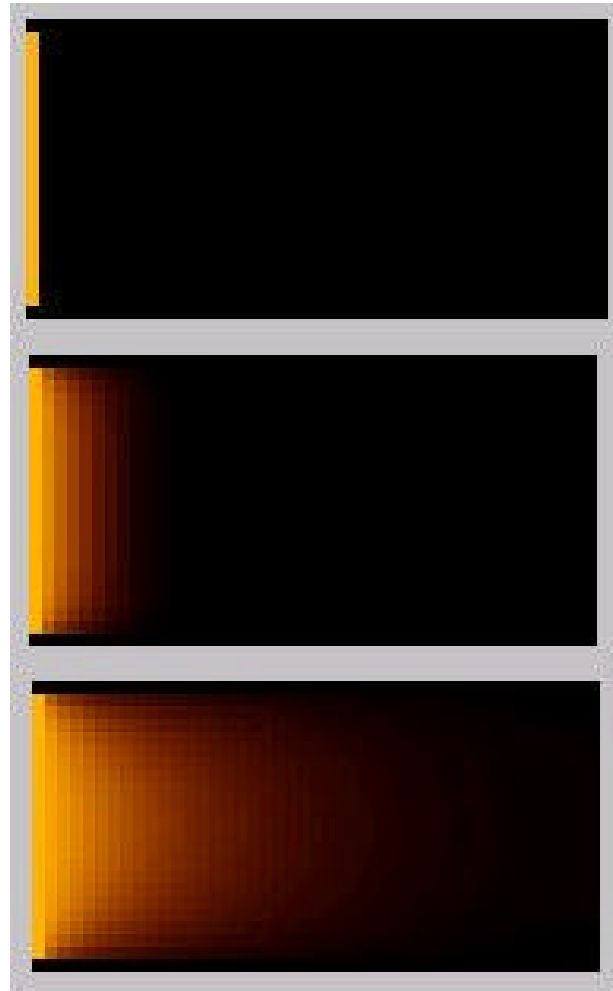
28	28	28	28	28	28	28	28	28	28	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	0	0	0	0	0	0	0	0	0	34
28	28	28	28	28	28	28	28	28	28	34

28	28	28	28	28	28	28	28	28	28	34
28	28	28	28	28	28	28	28	29	30	34
28	28	28	28	28	28	28	28	29	31	34
28	28	28	28	28	28	28	29	30	32	34
28	28	28	28	28	28	29	29	30	32	34
28	28	28	28	28	28	29	29	30	32	34
28	28	28	28	28	28	29	29	30	32	34
28	28	28	28	28	28	29	29	30	32	34
28	28	28	28	28	28	28	29	29	31	34
28	28	28	28	28	28	28	28	28	28	34

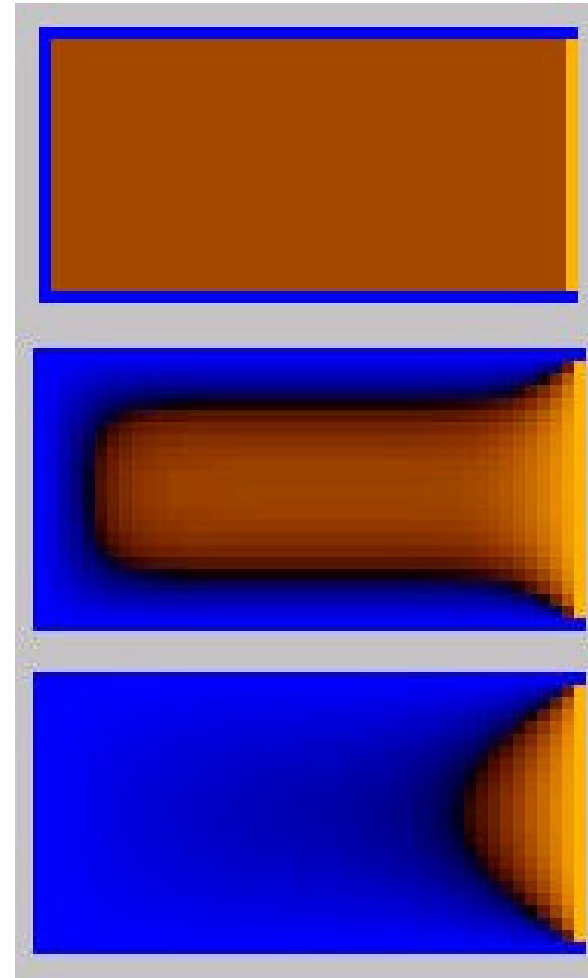
exemplo



# Difusão de Calor



■ 0°  
■ 400°



■ -40°  
■ 25°  
■ 60°



# Formulação Matemática

- ✍ A evolução térmica é regida pela equação de Laplace:

$$\frac{\partial^2 h}{\partial X^2} + \frac{\partial^2 h}{\partial Y^2} = 0$$

onde  $h(x,y)$  é a temperatura no ponto  $x,y$ .



# Formulação Matemática

- Como a imagem é tratada como uma função discreta de duas dimensões, não podemos utilizar a formulação contínua.
- Formulação discreta dada por diferenças centrais:

$$? \quad 4h_{i,j} \quad ? \quad h_{i-1,j} \quad ? \quad h_{i+1,j} \quad ? \quad h_{i,j-1} \quad ? \quad h_{i,j+1} \quad ? \quad 0$$

onde  $h_{i,j}$  é a temperatura no ponto  $i,j$ .



# Consideração sobre os dados

## ✎ Sistema linear:

- ✎ Terreno:  $N \times N$
- ✎  $N^2$  incógnitas ( $h_{i,j}$ )
- ✎ Matriz do sistema linear:  $N^2 \times N^2$

## ✎ Exemplo:

- ✎  $N = 1000 ? \quad 10^6 \times 10^6$
- ✎ Impraticável para máquinas e algoritmos de propósito geral.



# Primeiro Algoritmo

```
evolve( $h$ ):  
   $a \leftarrow h$   
  loop  
     $b \leftarrow a$   
     $a \leftarrow h$  OVER  $\mathcal{L}(a)$   
  if  $a = b$  return  $a$ 
```

- ✍ O operador Laplaciano implementa a difusão de calor na placa
- ✍ É determinado a partir da média dos valores dos vizinhos de  $a_{i,j}$

$x$  OVER  $y$  em  $(i,j)$  é  $x_{i,j}$  se  $x_{i,j} > 0$  e  $y_{i,j}$  se  $x_{i,j} = 0$



# Primeiro Algoritmo

- ✎ Convergência lenta
  - ✎ igualdade numérica (característica do processo)
- ✎ Alteração da condição de convergência

- ✎ intervalo numérico

```
evolve( $h, a$ ):
```

```
  loop
```

```
     $b \leftarrow a$ 
```

```
     $a \leftarrow h$  OVER  $\mathcal{L}(a)$ 
```

```
  if  $|a - b| < \epsilon$  return  $a$ 
```

- ✎ Problemas:

- ✎ Tempo de convergência
  - ✎ Influência do valor inicial





# Influência do Valor Inicial

- ✍ Muitos pontos com temperatura (altura) inicial igual a zero
  - ✍ Muita energia é necessária para “esquentarmos” esses pontos a uma temperatura comparável a das curvas
- ✍ Então, se começássemos a simulação já “próxima” do equilíbrio a convergência seria mais rápida.



# Algoritmo Multiresolução

- ✍ Resolver o problema de difusão em várias resoluções
  - ✍ Usa-se o resultado uma resolução como valor inicial para a próxima
  - ✍ O aumento da resolução não provoca um aumento muito grande no tempo de processamento, pois já partimos de “bons” valores iniciais



# Algoritmo Multiresolução

```
m-evolve( $h$ ):  
  for  $k = 6$  to  $10$   
     $r_k \leftarrow \text{reduce}(h, 2^k)$   
   $s \leftarrow r_6$   
  for  $k = 6$  to  $10$   
     $t \leftarrow \text{evolve}(r_k, s)$   
     $s \leftarrow \text{duplicate}(t)$   
  return  $t$ 
```

- ✍ A função evolve foi modificada para receber os valores iniciais;
- ✍ A difusão de calor é, então, simulada em cada resolução;
- ✍ A função m-evolve é muito mais rápida que a evolve.

$\text{reduce}(h, 2^k)$  ? reduz a escala da matriz  $h$  para a resolução  $2^k$

$\text{duplicate}(t)$  ? ajusta a resolução da matriz  $t$

$k = 6..10$  ? significa resoluções de  $64 \times 64$  ( $2^6$ ) a  $1024 \times 1024$  ( $2^{10}$ )



# Resultados: Numéricos

✎ Utilizou-se funções analíticas do tipo  $f(x,y)=h$  para podermos determinar todos os pontos de uma determinada região.

✎  $f(x,y) = x^2 + y^2$

✎  $f(x,y) = ? \sqrt{x^2 + y^2}$

Obs.: Essas funções foram escolhidas pois suas variações são linear e quadrática, respectivamente



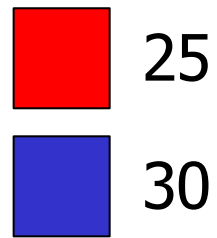
# Resultados: Numéricos

curvas de controle	morfologia	difusão de calor
cota de 43	5	10
cota de 90	6	10
cota de 130	2	6

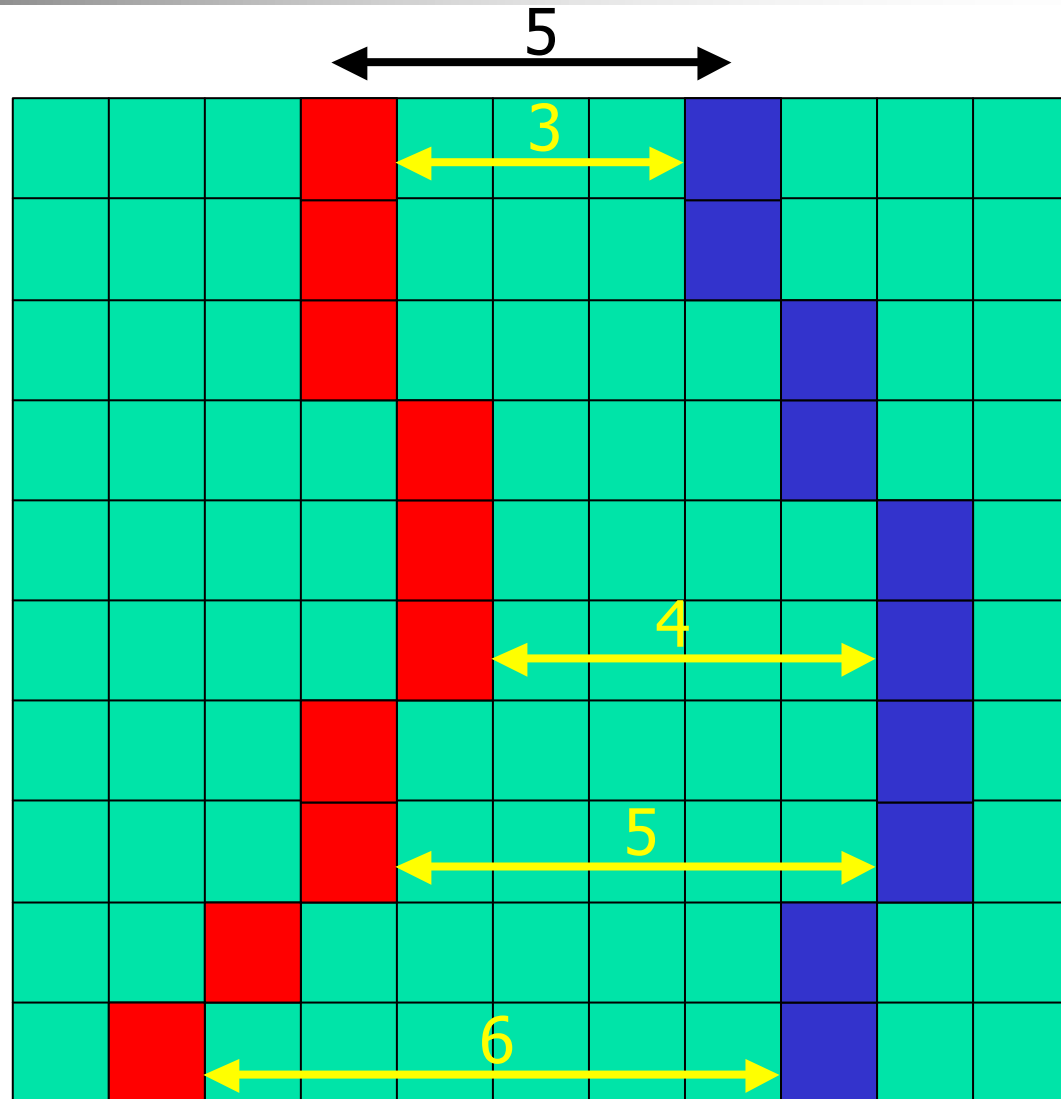
- ✍ A tabela acima mostra a máxima diferença entre as curvas real e interpolada pelos métodos de morfologia matemática e difusão de calor.
- ✍ As curvas interpoladas não atingiram as curvas reais porque a imagem de teste possuía resolução (pixels) menor do que o intervalo numérico entre as curvas.



# Resultados: Numéricos



28 ?





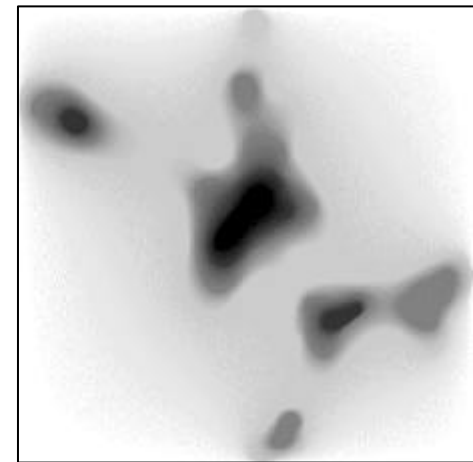
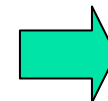
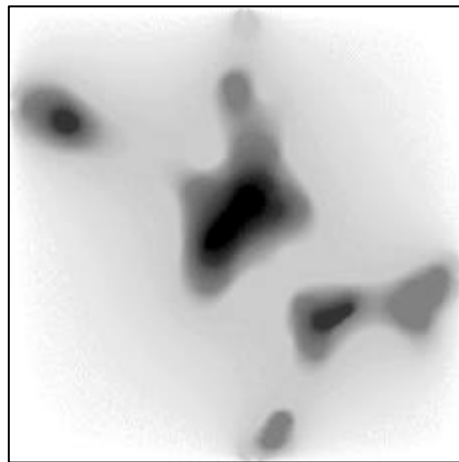
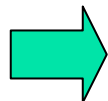
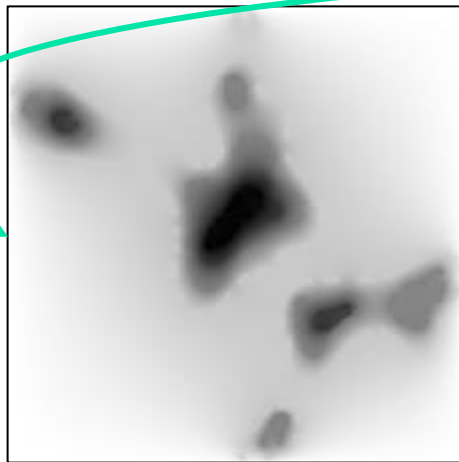
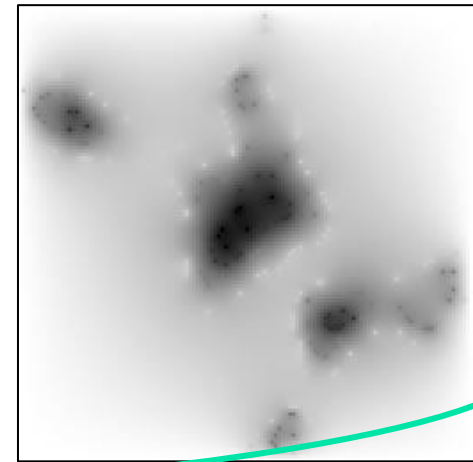
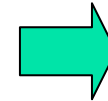
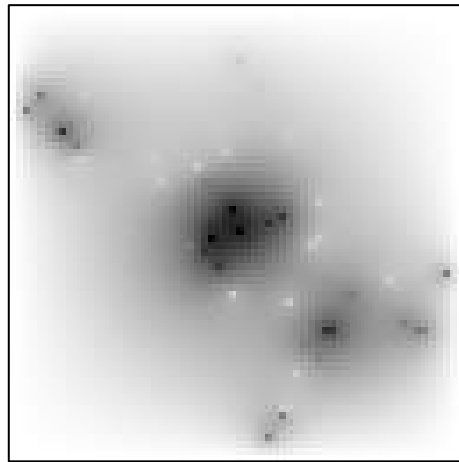
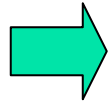
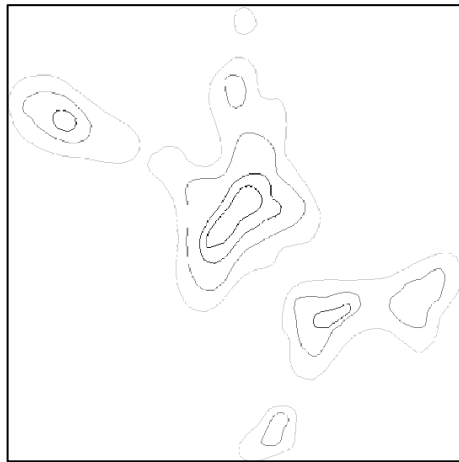
# Resultados: Visuais

---

- ✍ A etapa de inspeção visual consiste em gerar os modelos a partir de dados, cuja topografia seja conhecida, de modo que seja possível identificar com maior facilidade artefatos ou discrepâncias no modelo de elevação gerado.



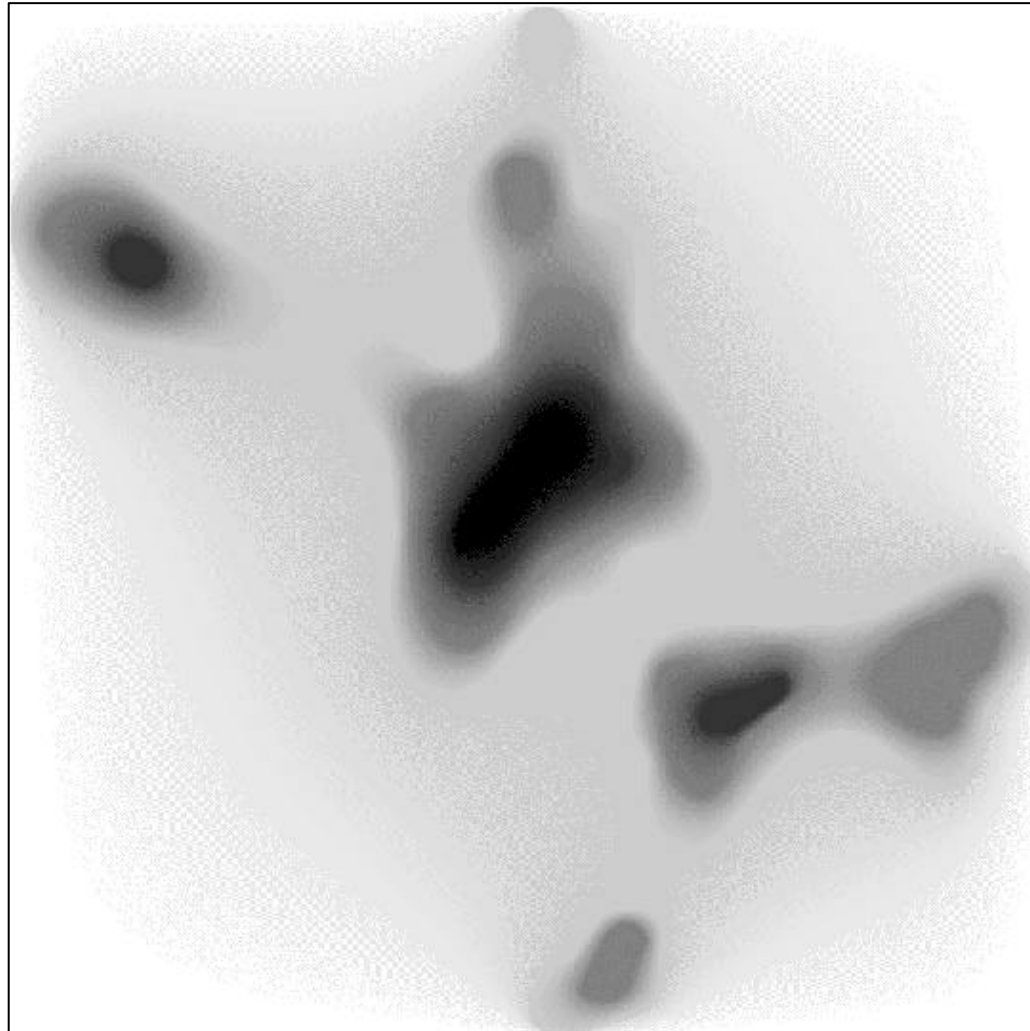
# Resultados: Visuais





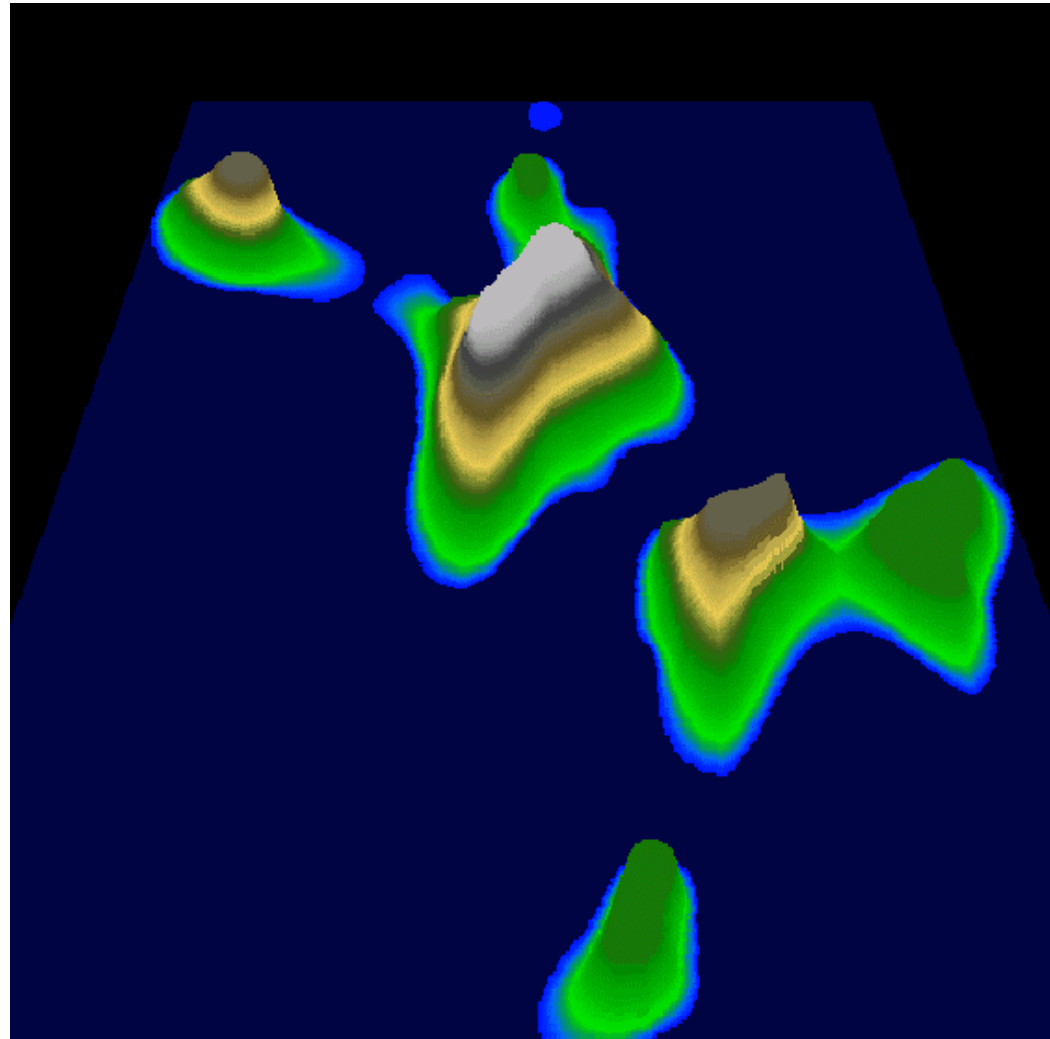
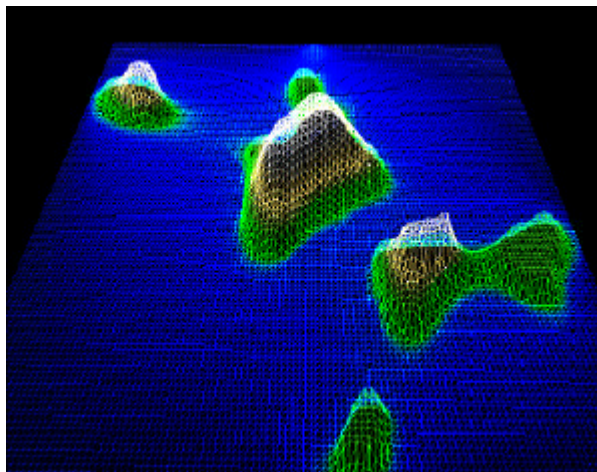
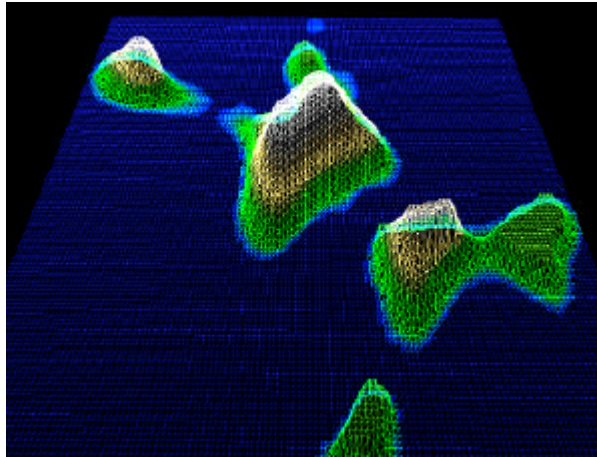


# Resultados: Visuais





# Resultados: Visuais





# Resultados: Tempos

resolução	iterações	tempo
64	665	0.63
128	551	2.06
256	616	10.40
512	328	23.76
1024	511	147.61
	tempo total	184.46

- Tempos de cada etapa na interpolação da figura do slide anterior.
- O tempo total para interpolar essa mesma imagem utilizando morfologia matemática foi de 1242.51 segundos, enquanto que por difusão de calor foi de apenas 184.46 segundos (? 15%).



# Resultados: Tempos

resolução	iterações	tempo
64	466	0.44
128	670	2.53
256	1155	19.21
512	512	37.07
	tempo total	59.25

- ✍ Tempos para a interpolação por difusão de calor da função analítica  $f(x,y)=x^2+y^2$ .
- ✍ O Tempo total gasto pela morfologia matemática foi de 154.71 segundos, enquanto que por difusão de calor foi de 59.25 segundos.



# Resultados: Tempos

resolução	iterações	tempo
64	883	0.84
128	822	3.17
256	964	15.96
512	1226	88.69
	tempo total	108.66

- ✍ Tempos para a interpolação por difusão de calor da função analítica  $f(x,y) = \sqrt{x^2 + y^2}$
- ✍ O Tempo total gasto pela morfologia matemática foi de 150.38 segundos enquanto que por difusão de calor foi 108.66 segundos.



# Conclusões

- ✍ Técnica simples de implementar
  - ✍ Formulação discreta sugere a média dos pixels vizinhos.
- ✍ Com o método de multi-resolução o tempo de processamento reduziu bastante
  - ✍ Como foi mostrado na seção “Resultados: Tempo”, os tempos para processamento das mesmas imagens utilizando morfologia matemática foram muito maiores.
- ✍ Os resultados visuais e numéricos foram bons
  - ✍ Não encontramos os platôs citados por Gousie (1998). A implementação deve ter usado aritmética inteira.



# Trabalhos Futuros

- ✍ Experimentar outros métodos de interpolação
  - ✍ modelo “thin-plate”.
  - ✍ outros métodos baseados em equações diferenciais parciais.
- ✍ Estudar métodos para avaliação dos terrenos interpolados





# Bibliografia

- ✦ W. Barrett, E. Mortensen, and D. Taylor. An image space algorithm for morphological contour interpolation. In *Proceedings of Graphics Interface '94*, pages 16–24, 1994.
- ✦ R. de Beauclair Seixas, L. H. de Figueiredo, C. A. da Silva, and P. C. P. Carvalho. Uma metodologia para geração de modelos de elevação a partir de curvas de nível. In *Anais do GeoInfo 2000*, pages 82–87, 2000.
- ✦ M. B. Gousie. *Contours to Digital Elevation Models: Grid-Based Surface Reconstruction Methods*. PhD thesis, Rensse-laer Polytechnic Institute, 1998.
- ✦ J. D. Wood and P. F. Fisher. Assessing interpolation accuracy in elevation models. *IEEE Computer Graphics and Applications*, 13(2):48–56, 1993.