

Paralelização do algoritmo de Estimativa de Densidade (KDE) utilizando GPU/CUDA

Juan Emanuel Hipólito Valenzuela
Raul Sena Ferreira
Orientador: D. Sc. Marcelo Zamith

Sumário

- Motivação
- KDE (Kernel Density Estimation)
- Paralelização com CUDA/GPU
- Resultados Computacionais
- Trabalhos Futuros
- Bibliografia



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Motivação

- Estatisticamente, alguns dados ou populações não possuem estruturas ou parâmetros característicos(não-paramétricos)
- Nesse caso, os dados precisam ser visualizados e/ou tratados de forma diferente dos dados convencionais
- Um algoritmo amplamente usado para este tipo de amostra é o KDE

KDE (Kernel Density Estimation)

- O algoritmo de Estimativa de Densidade pelo método Kernel (KDE) é um método não paramétrico de análise de dados.
- O estimador kernel pode ser pensado como uma generalização do histograma
- Complexidade $O(n^2k)$
- A partir de um dado número de observações n , calculamos curvas de densidade delas em relação a distância de um valor central μ



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

KDE (Kernel Density Estimation)

- Muito utilizado em:
 - Correção de ruídos em sinais elétricos
 - Análise estática de dados
 - Estimativa de densidade populacional
 - Estimativa de Densidade Geográfica de uma região

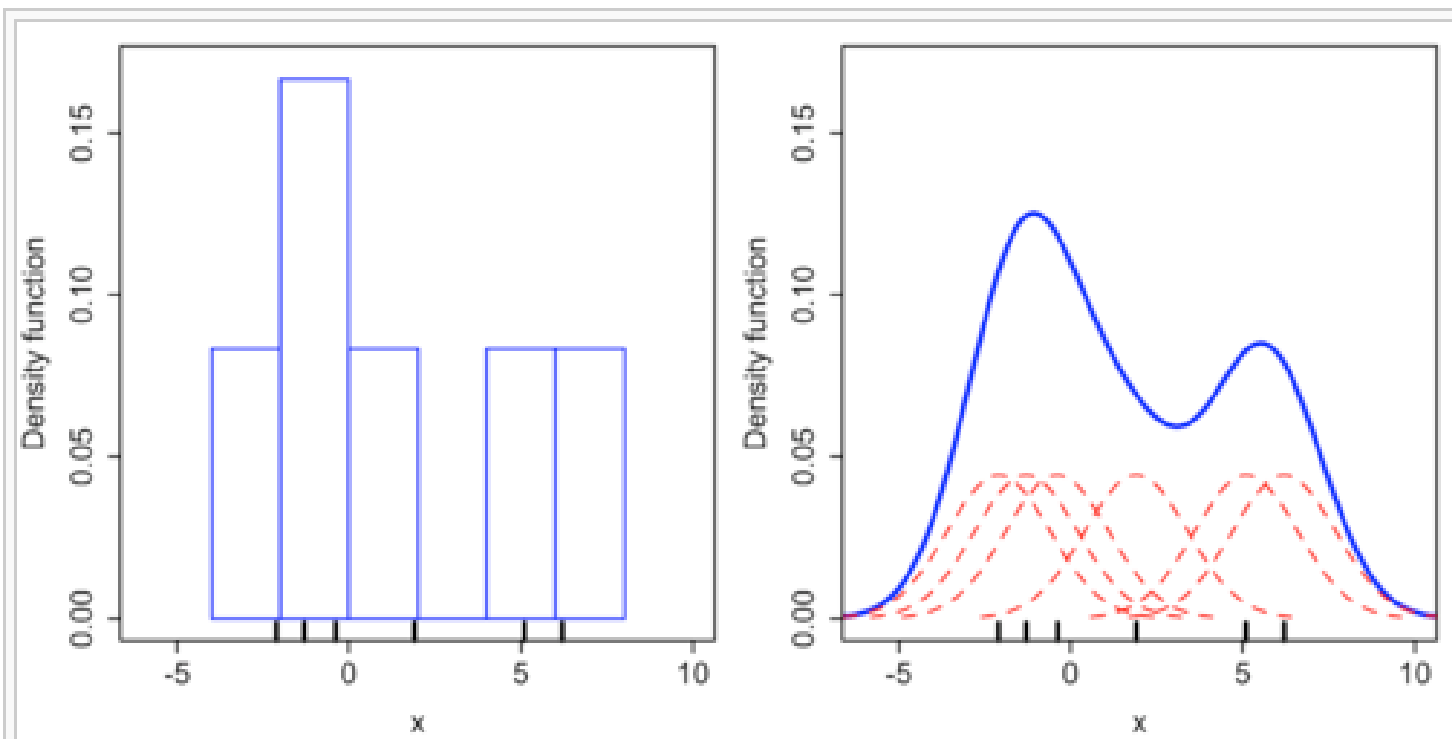


UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

KDE (Kernel Density Estimation)



Comparison of the histogram (left) and kernel density estimate (right) constructed using the same data. The 6 individual kernels are the red dashed curves, the kernel density estimate the blue curves. The data points are the rug plot on the horizontal axis.



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

KDE (Kernel Density Estimation)

Algorithm 3 KDE Multivariante

```
1: for  $i \leftarrow 0$  to  $n$  do
2:    $soma\_kernel \leftarrow 0.0$ 
3:   for  $j \leftarrow 0$  to  $n$  do
4:      $prod\_kernel \leftarrow 1.0$ 
5:     for  $k \leftarrow 0$  to  $xLen$  do
6:        $prod\_kernel \leftarrow prod\_kernel * K((x[i][k] - x[j][k])/h)/h$ 
7:     end for
8:      $soma\_kernel \leftarrow soma\_kernel + prod\_kernel$ 
9:   end for
10:   $pdf[i] \leftarrow soma\_kernel/n$ 
11: end for
```

Paralelização com CUDA/GPU

- Quando o número de indivíduos é grande o tempo do algoritmo tende a crescer rapidamente.
- Por ser um problema altamente escalável, podemos tirar proveito da GPU para executar o algoritmo de forma paralelizada.

Paralelização com CUDA/GPU

- Utilizando o modelo de programação unificado CUDA (*Compute Unified Device Architecture*), criamos um programa *Host*, que irá gerenciar a troca de informações entre a GPU e o CPU.
- A GPU tem um grande número de processadores e milhares de *threads* em cada um deles, logo, a velocidade de execução do algoritmo aumenta drasticamente.



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Resultados Computacionais

- Todos os resultados utilizando CUDA foram obtidos usando 1000 Threads por bloco
- Os dados utilizados no experimento foram as latitudes e longitudes convertidos a partir do CEP dos alunos matriculados na UFRRJ
- Os testes foram executados 3 vezes com exclusividade na máquina e depois foi extraído a média

Resultados Computacionais

Pontos(x,y)	Tempo Sequencial	Tempo MatLAB	Tempo CUDA
1000	0.181333s	0.126229s	0.020041s
2000	0.647333s	0.357517s	0.040032s
5000	4.045333s	1.192155s	0.188097s
10000	16.220666s	3.495502s	0.577605s
14027	31.317333s	6.275648s	1.033042s
30000	149,018333s	24.398795s	4.321454s



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

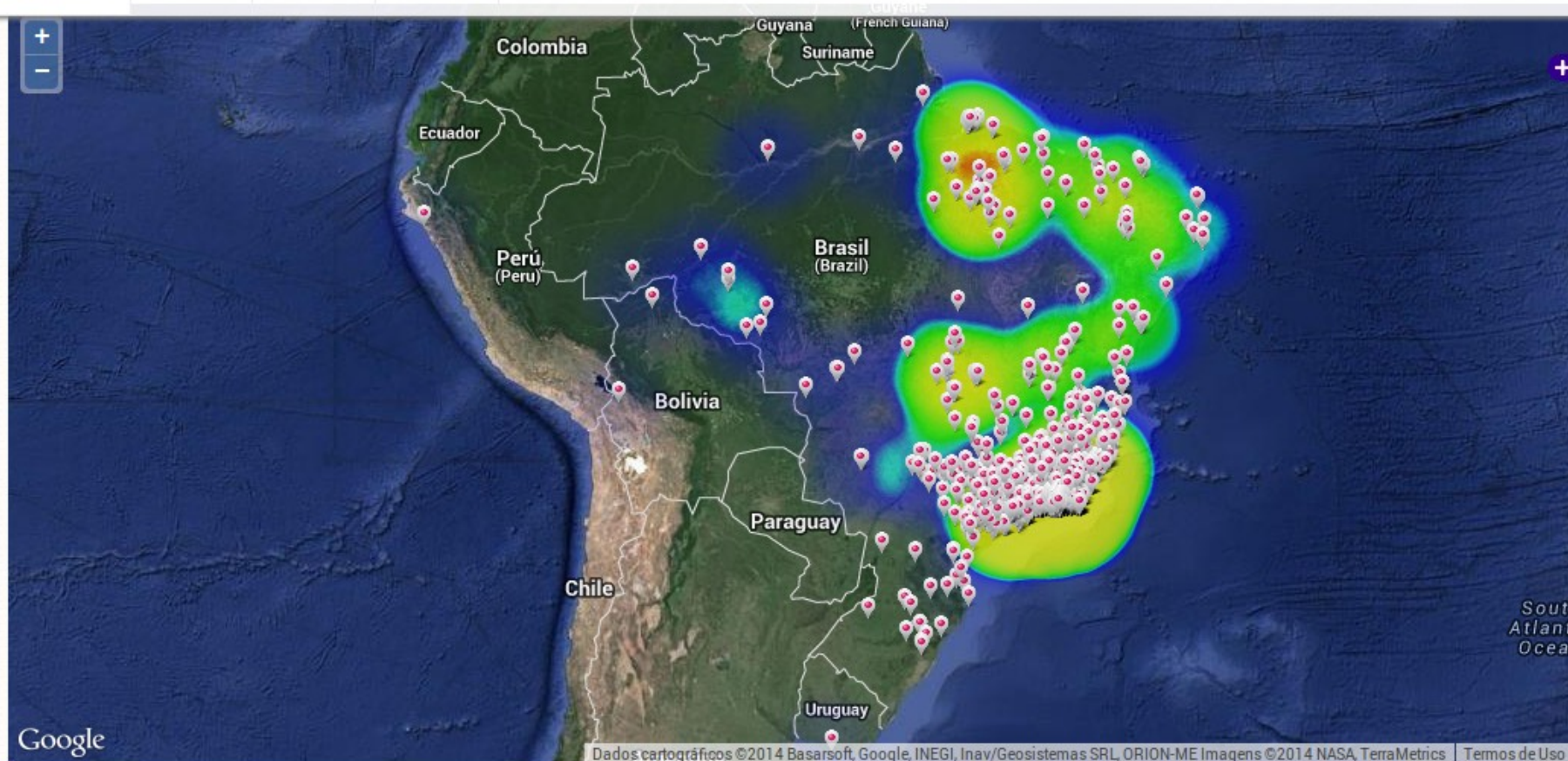
Resultados Computacionais



Mapa

Gráfico

Contato




Buscar


Nova Busca



Google

Dados cartográficos ©2014 Basarsoft, Google, INEGI, Inav/Geosistemas SRL, ORION-ME Imagens ©2014 NASA, TerraMetrics | Termos de Uso

Use o botão com a  para poder navegar no mapa.

Use o botão com o  para conseguir desenhar um polígono no mapa.



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Resultados Computacionais

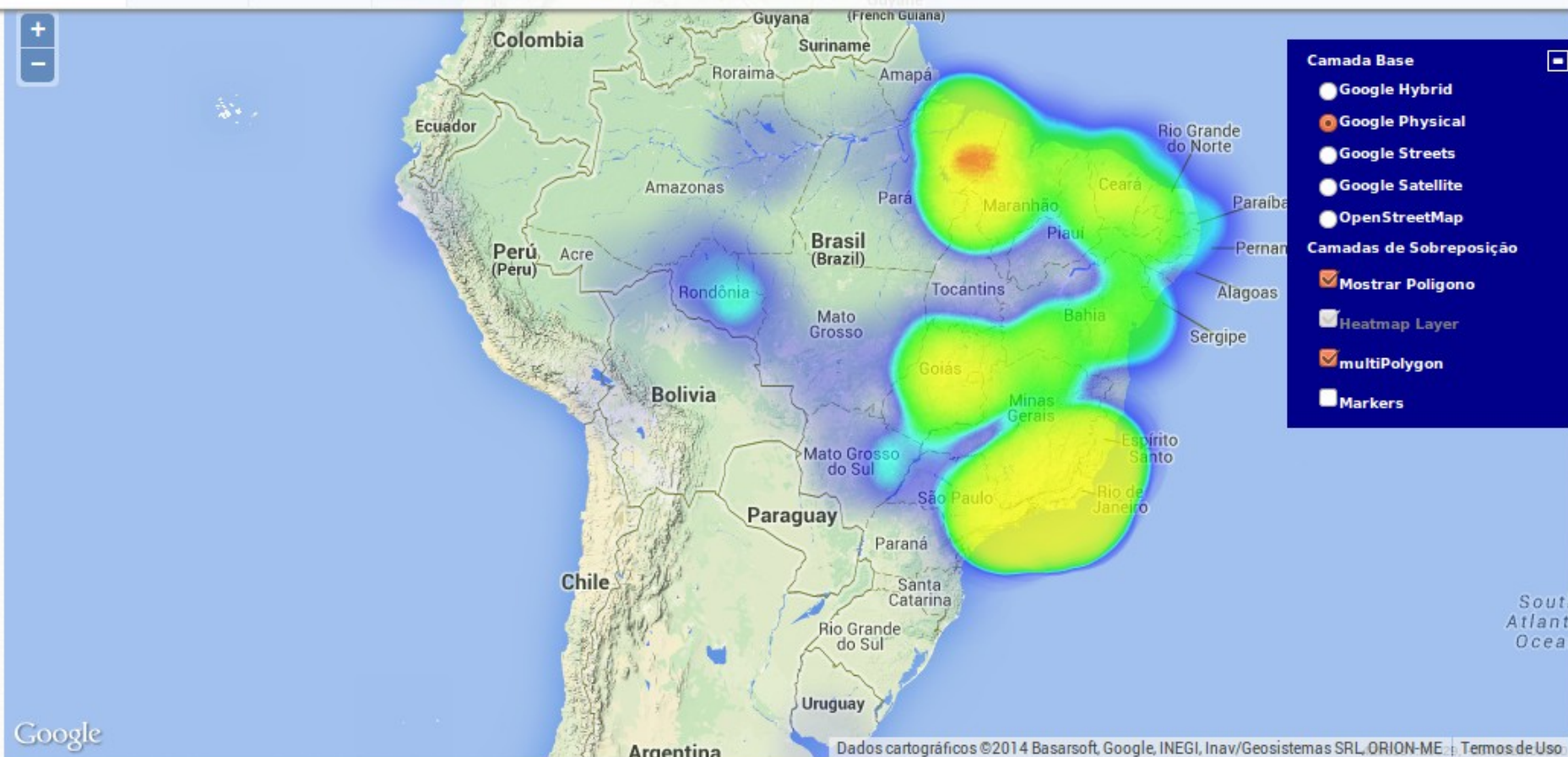


WSAGE

Mapa

Gráfico

Contato



Buscar

Nova Busca



Use o botão com a para poder navegar no mapa.

Use o botão com o para conseguir desenhar um polígono no mapa.

Trabalhos Futuros

- Otimização do uso das GPUs
- Automatizar alguns parâmetros de entrada do algoritmo
- Testar com populações maiores
- Fazer ajustes no algoritmo para tentar diminuir ainda mais o tempo



UFRRJ

UNIVERSIDADE FEDERAL RURAL
DO RIO DE JANEIRO

INSTITUTO MULTIDISCIPLINAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Bibliografia

- Programming Massively Parallel Processors - A Hands-on Approach - Second Edition - David B. Kirk and Wen-mei W. Hwu'
- Epanechnikov, V.A. (1969). "Non-parametric estimation of a multivariate probability density". Theory of Probability and its Applications 14: 153–158.
- <http://www.cin.ufpe.br/~fatc/AM/kernel.pdf>

Perguntas ou Sugestões ?

- Raul Sena Ferreira
raulsenaferreira@gmail.com
- Juan Emanuel Hipólito Valenzuela
juan.emanuel@outlook.com
- Marcelo Zamith
zamith.marcelo@gmail.com
- Link da aplicação teste:
<http://107.170.124.51/wsage/>