UNIVERSITÀ
DEGLI STUDI
FIRENZE

Elaborati Parallel Computing

Svolgimento

- È possibile svolgere gli elaborati in coppia
- Elaborato a 6 crediti: implementare un software con un solo approccio tra quelli visti a lezione + eventuale metodo non parallelo / due metodi diversi
- Elaborato a 9 crediti: implementare un software con due metodi diversi tra quelli visti a lezione.
- Relazione con analisi della performance

Thread pool

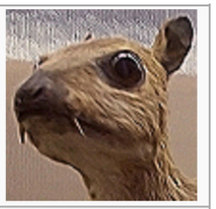
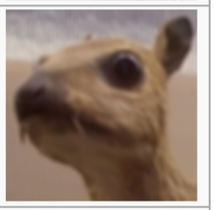
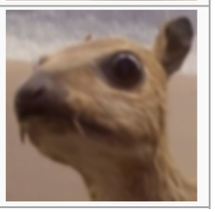
- Implementare Thread Pool in C/C++ usando Pthreads e/o C++11
- Ispirarsi a Java Thread Pool per API

Image reader

- Non blocking / multithread JPEG image reader
  - leggere X immagini da directory
  - se non-blocking leggere in background e poi rendere a richiesta le immagini, es. su base di nome di file o lista
  - usare due approcci diversi

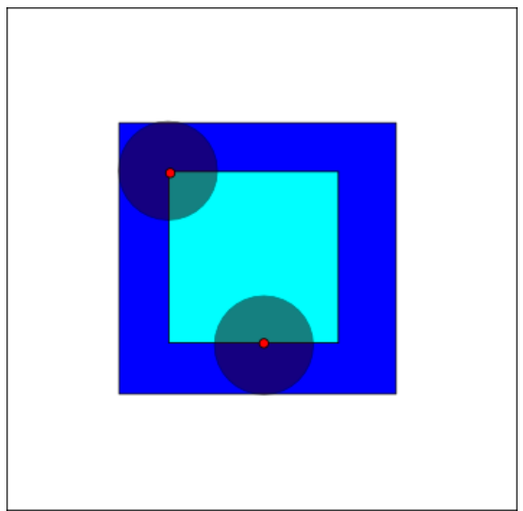
Kernel image processing

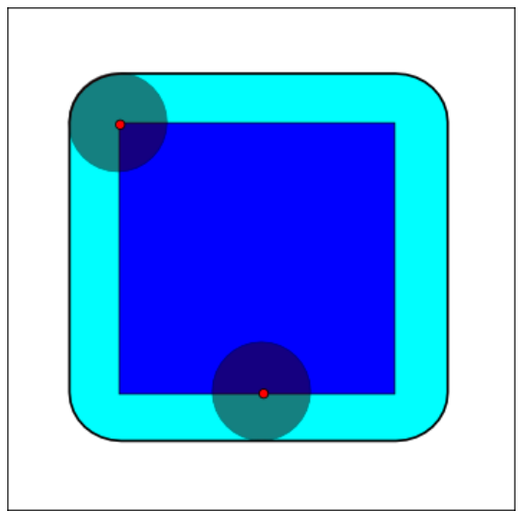
- Image processing
- usare due approcci diversi
- CUDA è ottimo candidato
- vedi https://en.wikipedia.org/wiki/Kernel_(image_processing)

Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Morphological image processing

- Image processing
  - usare due approcci diversi
  - CUDA è ottimo candidato
  - vedi https://en.wikipedia.org/wiki/Mathematical_morphology



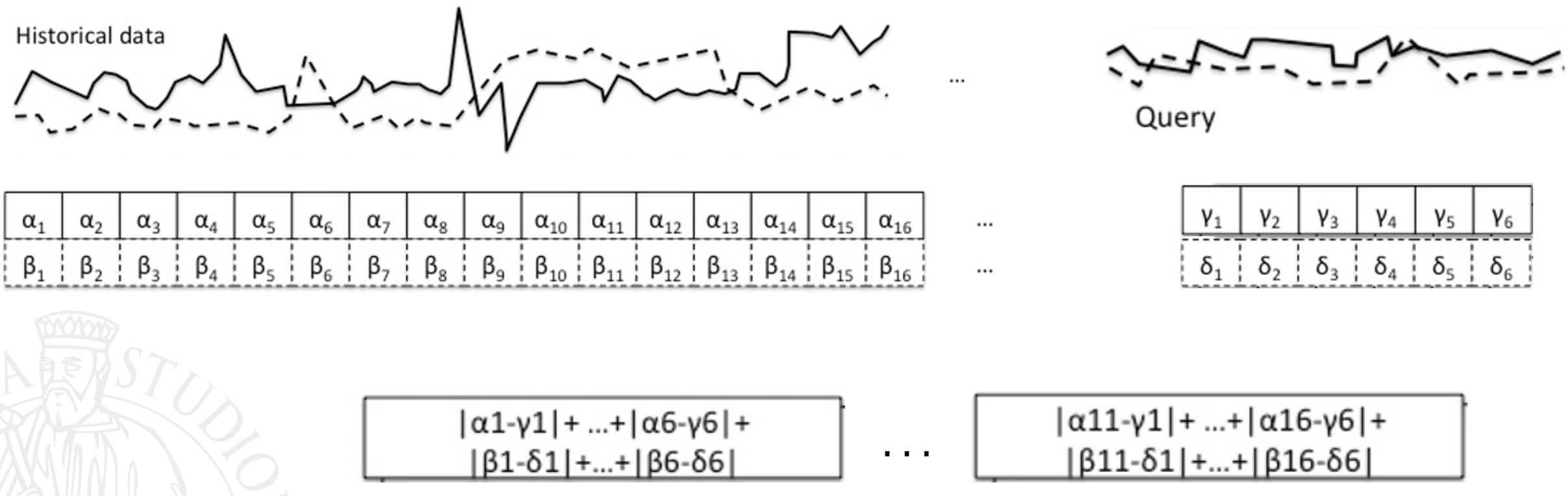


Bigrammi

- Calcolare bigrammi sul testo di Wikipedia
- sia bigrammi di parole che di lettere
- usare due approcci diversi
- Hadoop e MPI sono ottimi candidati

Pattern recognition

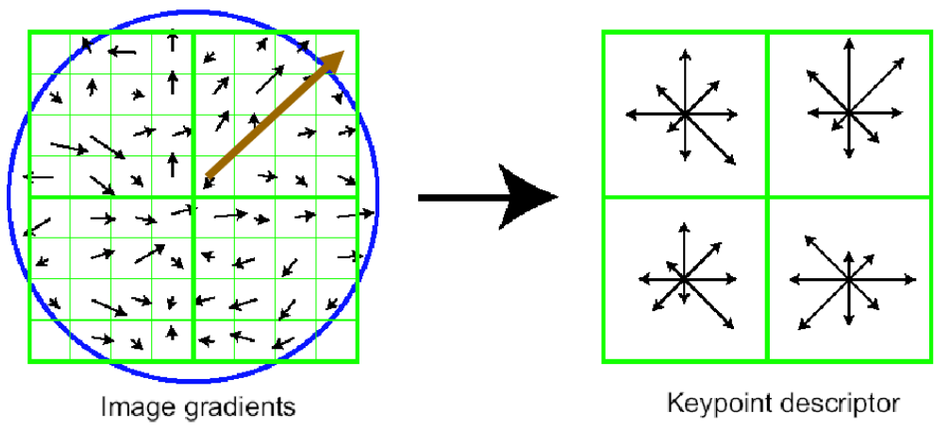
- Cercare all'interno di serie di valori un certo andamento
  - usare due approcci diversi
  - usare SAD come metrica per matching



Parallel SIFT

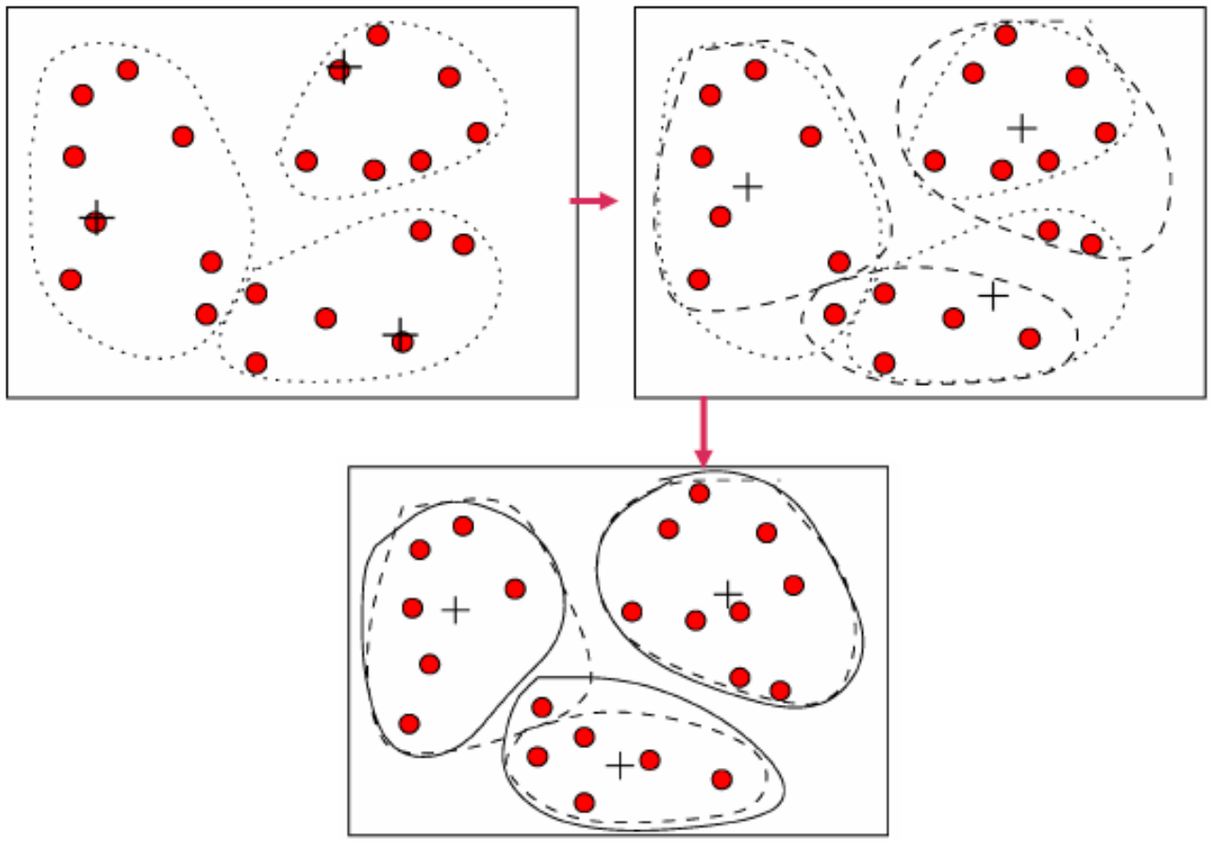
- Implementare il SIFT (versione open source in C di Rob Hess) e la versione in C++ di OpenCV con OpenMP

- vedi https://it.wikipedia.org/wiki/Scale-invariant_feature_transform



k-means

- Implementazione parallela del clustering k-means
  - OpenMP, MPI e CUDA sono ottimi candidati
- **Esempio**



Ricerca stringhe

- implementazione parallela edit distance (Levenshtein) + Bitap (ricerca approssimata, usata in agrep)

- OpenMP, MPI e CUDA sono ottimi candidati

- vedi https://en.wikipedia.org/wiki/Levenshtein_distance e https://en.wikipedia.org/wiki/Bitap_algorithm

MinHash

- implementazione MinHash parallela
- è un algoritmo per il calcolo della similarità tra documenti
- vedi http://matthewcasperson.blogspot.it/2013/11/minhash-for-dummies.html

Lambda architecture 1

- Implementare un sistema basato su Lambda architecture per
  - scaricare tweets geo-localizzati
  - dare statistiche su keyword e hashtags

Lambda architecture 2

- Implementare un sistema basato su Lambda architecture per prendere immagini da Google / Bing / Flickr relative a determinate keyword
  - il sistema fa crawling dei siti e scarica le immagini
  - il sistema fornisce ad un certo istante tutte le immagini associate ad una certa keyword che ha scaricato

Lambda architecture 3

- Implementare un sistema basato su Lambda architecture per dare istogrammi di frequenza di immagini simili su Twitter

  - il sistema scarica tweet associati a keyword o hashtags

  - compara le immagini associate per valutare quante immagini uguali sono presenti fino ad un certo istante

  - fornisce l'immagine più rappresentativa ad ogni istante