KNN GPGPU Documentation

Includes

point.h Contains definitions of the different point struct data-types used by the knn gpgpu algorithms.

Members

void buildKdTree(struct Point *points, int n, struct Node* tree) Accepts a list of n PointS. Builds a balanced kd-tree from these points on the GPU, and writes this tree to the Point list tree.

void queryAll(struct Point *query_points, struct Node* tree, int n_qp, int n_tree, int k, int *result) Queries a previously built kd-tree of size n_tree for the k closest neighbors to the points specified in the query_points list of size n_qp. The index location of the k closest points are written to the result array. Uses a wrapper to partition the problem, in order to handle memory overflow.

void cuQueryAll(struct Point *query_points, struct Node* tree, int n_qp, int n_tree, int k, int *result) Queries a previously built kd-tree of size n_tree for the k closest neighbors to the points specified in the query_points list of size n_qp. The index location of the k closest points are written to the result array.

void mpQueryAll(struct Point *query_points, struct Node* tree, int n_qp, int n_tree, int k, int *result) Performes same operations as cuQueryAll, but is parallelized on the CPU using OpenMP instead of CUDA.

void knn_brute_force_garcia(float *ref_host, int ref_width, float* query_host, int query_width, int height, int k, float *dist_host, int* ind_host) Performs a brute force knn-search based on the code written by Garcia.

void knn_brute_force(float *ref_host, int ref_nb, float* query_host, int dim, int k, float *dist_host, int* ind_host) Performs a improved brute force knn-search.

Utils

size_t getFreeBytesOnGpu() Return the current amount of free memory on the GPU in bytes.

void cuSetDevice(int device) Sets device as the current device for the calling host thread.

int cuGetDevice() Returns the device on which the active host thread executes the device code.

int cuGetDeviceCount() Returns the number of devices accessible.

size_t getNeededBytesForBuildingKdTree(int n_tree) Returns needed bytes on GPU to build a tree of size n_tree.

size_t getTreeSize(int n_tree) Returns the size in bytes of a tree with length n_tree.

size_t getNeededBytesForQueryAll(int n_qp, int k, int n_tree) Returns needed bytes on GPU to perform a queryAll operation on CUDA.

size_t getNeededBytesInSearch(int n_qp, int k, int n_tree, int thread_num, int block_num) Returns needed bytes on GPU to perform a queryAll operation on CUDA without taking the tree size into account.