MGCD Reference Manual

0.1

Generated by Doxygen 1.5.3

Thu Sep 6 00:44:55 2007

Contents

Chapter 1

libmgcd

1.1 Introduction

LIBMGCD is a C implementation of Multi Grid Contact Detection method (MGCD).The multigrid idea is integrated with contact detection problems. Both the time complexity and memory consumption of the MGCD are O(N). Unlike other methods, whose efficiencies are influenced strongly by the object size distribution, the performance of MGCD is insensitive to the object size distribution.

1.2 Introduction

When performing 2D detection, LIBMGCD needs :

```
sizeof(Int)*(AABBCount+nx*ny) for the finest grid using screening method,
sizeof(Int)*(AABBCount+2*nx*ny) bytes for other grids using screening method;
sizeof(Int)*(2*AABBCount+2*nx+ny)+nx+ny bytes for the finest grid using nbs method,
sizeof(Int)*(2*AABBCount+4*nx+ny)+nx+ny bytes for other grids using nbs method.
```

When performing 3D detection, LIBMGCD needs :

```
sizeof(Int)*(AABBCount+nx*ny*nz) bytes for the finest grid using screening method,
sizeof(Int)*(AABBCount+2*nx*ny*nz) bytes for other grids using screening method;
sizeof(Int)*(3*AABBCount+2*nx+2*ny+nz)+nx+ny+nz bytes for the finest grid using nbs method,
sizeof(Int)*(3*AABBCount+4*nx+2*ny+nz)+nx+ny+nz bytes for other grids using nbs method.
```

where nx, ny, nz is the cell count in x, y and z direction of coresponding grid.

1.3 Installation

Follow these steps to build and install the mgcd library, suppose you are in the root directory of this libmgcd distibution:

```
$./configure --prefix = path_to_install
$make
```

```
$make install
```

After these, the mgcd library and the sample program will be installed to a directory indicated by path_to_install.

1.4 Example

The following code demonstrate how to use the mgcd api, and it can be found in path_to_install/examples after installing libmgcd.

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "mgcd.h"

#define FRAND() (rand()%1000/1000.0)

void ramdomBoxBuilder3D(Float minx , Float miny, Float minz,
                        Float maxx , Float maxy, Float maxz,
                        Float dx, Float dy, Float dz,
                        struct AABB3D (p.11) * buf,
                        Int start, Int count
                        ){
    unsigned seed = (unsigned)time(NULL);

    Float rx ;
    Float ry = dy/2;
    Float rz = dz/2;
    Float wx = maxx - minx;
    Float wy = maxy - miny;
    Float wz = maxz - minz;
    Int i ;

    rx = dx/2;
    srand (seed);
    for(i= 0 ; i<count ; ++i){

        buf [start + i].x = FRAND() * wx + minx;
        buf [start + i].y = FRAND() * wy + miny;
        buf [start + i].z = FRAND() * wz + minz;
        buf [start + i].rx = rx;
        buf [start + i].ry = ry;
        buf [start + i].rz = rz;
    }
}

void test3D (){
    Int countNbs, countScr;
    int c1 = 10000;
    int c2 = 2560000;
    struct AABB3D (p. 11) * elems = malloc((c1+c2)*sizeof(*elems));
    Float grids [] = {0.1, 0.01};
    ramdomBoxBuilder3D(0, 0, 0,
                       5, 5, 5,
                       0.1, 0.1, 0.1,
                       elems, 0, c1);

    ramdomBoxBuilder3D(0, 0, 0,
                       5, 5, 5,
                       0.01, 0.01, 0.01,
                       elems, c1, c2);

    printf("mgcd 3D nbs ... \n");
    countNbs = mgcdContactDetect3D (grids, 2, elems, c1+c2, "NBS", NULL, NULL);

    printf("mgcd 3D scr ... \n");
    countScr = mgcdContactDetect3D (grids, 2, elems, c1+c2, "SCR", NULL, NULL);

    printf("nbs : %d\n", countNbs);
    printf("scr : %d\n", countNbs);

    fflush(stdout);
    free (elems);
}

int main (){
    test3D();
    return 0;
}
```

Chapter 2

MGCD Class Index

2.1 MGCD Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

MGCD File Index

3.1 MGCD File List

Here is a list of all documented files with brief descriptions:

Chapter 4

MGCD Class Documentation

4.1 AABB2D Struct Reference

A data structure describing a 2D box.

`#include <mgcd.h>`

Public Attributes

- Float **x**
- Float **y**
- Float **rx**
- Float **ry**

4.1.1 Detailed Description

A data structure describing a 2D box.

The MGCD 2D api treat all objects on which the contact detection perform as 2D AABBs. User should use this struct to describe the objects

See also:

AABB3D (p. 11)

4.1.2 Member Data Documentation

4.1.2.1 Float AABB2D::x

x coordinate of box center

4.1.2.2 Float AABB2D::y

y coordinate of box center

4.1.2.3 Float AABB2D::rx

half of the size of a box in x direction

4.1.2.4 Float AABB2D::ry

half of the size of a box in y direction

The documentation for this struct was generated from the following file:

- D:/workspaces/sourceforge/mgcd/src/**mgcd.h**

4.2 AABB3D Struct Reference

A data structure describing a 3D box.

`#include <mgcd.h>`

Public Attributes

- Float **x**
- Float **y**
- Float **z**
- Float **rx**
- Float **ry**
- Float **rz**

4.2.1 Detailed Description

A data structure describing a 3D box.

The MGCD 3D api treat all objects on which the contact detection perform as 3D AABBs. User should use this struct to describe the objects

See also:

AABB2D (p. 9)

4.2.2 Member Data Documentation

4.2.2.1 Float AABB3D::x

x coordinate of box center

4.2.2.2 Float AABB3D::y

y coordinate of box center

4.2.2.3 Float AABB3D::z

z coordinate of box center

4.2.2.4 Float AABB3D::rx

half of the size of a box in x direction

4.2.2.5 Float AABB3D::ry

half of the size of a box in y direction

4.2.2.6 Float AABB3D::rz

half of the size of a box in z direction

The documentation for this struct was generated from the following file:

- D:/workspaces/sourceforge/mgcd/src/**mgcd.h**

Chapter 5

MGCD File Documentation

5.1 D:/workspaces/sourceforge/mgcd/src/mgcd.h File Reference

API for MGCD library.

`#include <math.h>`

Include dependency graph for mgcd.h:

Classes

- struct **AABB2D**

  A data structure describing a 2D box.

- struct **AABB3D**

  A data structure describing a 3D box.

Typedefs

- typedef void(∗ **MGCDHandlePair2D**)(Int el1, Int el2, struct **AABB2D** ∗els, void ∗context)

  2DBox pair handler function type definition

- typedef void(∗ **MGCDHandlePair3D**)(Int el1, Int el2, struct **AABB3D** ∗els, void ∗context)

  3DBox pair handler function type definition

Functions

- Int **mgcdContactDetect2D** (Float ∗grids, Int gridCount, struct **AABB2D** ∗elem, Int elemCount, const char ∗mstr, **MGCDHandlePair2D** func, void ∗context)

  Perform 2D multi grid contact dection.

- Int **mgcdContactDetect3D** (Float *grids, Int gridCount, struct **AABB3D** *elem, Int elemCount, const char *mstr, **MGCDHandlePair3D** func, void *context)

  Perform 3D multi grid contact dection.

5.1.1 Detailed Description

API for MGCD library.

Author:

Jialin Li <lijl.scut@gmail.com>, Kejing He <kejinghe@gmail.com>

MGCD stands for Multi Grid Contact Detection. It implements two well known algorithms for 2D and 3D contact detection, NBS and Screening.

5.1.2 Typedef Documentation

5.1.2.1 void(* MGCDHandlePair2D)(Int el1, Int el2, struct AABB2D *els, void *context)

2DBox pair handler function type definition

User can define a function of this type and provide the function pointer to MGCD 2D detection api to handle the contecting 2D box pairs when a contacting pair is detected

Parameters:

el1 the index of the first box in els

el2 the index of the second box in els

els a pointer to a struct **AABB2D** (p. 9) array containing all box infomation

context custom data struct used inside the function

See also:

MGCDPairHandler2D

5.1.2.2 void(* MGCDHandlePair3D)(Int el1, Int el2, struct AABB3D *els, void *context)

3DBox pair handler function type definition

User can define a function of this type and provide the function pointer to MGCD 3D detection api to handle the contecting 3D box pairs when a contacting pair is detected.

Parameters:

el1 The index of the first box of the contacting pair in els

el2 The index of the second box of the contacting pair in els

els A pointer to a struct **AABB3D** (p. 11) array containing all box infomation

context Custom data struct used inside the function

See also:

MGCDPairHandler3D

5.1.3 Function Documentation

5.1.3.1 Int mgcdContactDetect2D (Float ∗ *grids*, Int *gridCount*, struct AABB2D ∗ *elem*, Int *elemCount*, const char ∗ *mstr*, MGCDHandlePair2D *func*, void ∗ *context*)

Perform 2D multi grid contact dection.

This function take the basic informations of grid configuration, information of all AABB2Ds the detection perform on and a custom contacting box pair handler as input parameter. It then perform the 2D Multi Grid Contact detection algorithm automatically and return the contacting pair count if no error occur, or error code otherwise.

Parameters:

grids A Float array indecating the cell size of each grid

gridCount An integer indecation the number of grids

elem A **AABB2D** (p. 9) array, describing the position and size of all objects

elemCount Number of AABB2Ds

mstr A string describing what detection algorithm each grid should use -"NBS" : Use NBS algorithm for each grid. -"SCR" : Use Screening algorithm for each grid. -"AUTO" : Default configuration, in this implementation, NBS will be used. -custom : User can provide a method configuration string to customize what algorithm each grid should use. The length of method cofiguration string should equals to gridCount, and each character should be 'a', 'n' or 's', for each representing a selection of algorithm:

1. 'a' : Automatic selection, in this implementaion NBS will be used
2. 's' : Screening
3. 'n' : NBS eg. "asn": this method configuration string tell the MGCD library automatically select a detection algorithm for the finest grid, use NBS for the coarsest grid, and use Screening for the middle grid.

func A custom contacting box pair handler, can be NULL

context Custom data structure used by handler

Returns:

This function return the contacting 2D box pair count if no error occur, or else return a error code who is a negative. -MGCD_METHOD_ERROR Invalid methods configuration, -MGCD_GRID_ERROR Invalid grid configuration, eg. invalid or duplicate cell size or can not find a grid to fit a box in -MGCD_BOX_ERROR invalid box information, eg. negative size of a box

See also:

AABB2D (p. 9), MGCDPairHandler2D

5.1.3.2 Int mgcdContactDetect3D (Float * *grids*, Int *gridCount*, struct AABB3D * *elem*, Int *elemCount*, const char * *mstr*, MGCDHandlePair3D *func*, void * *context*)

Perform 3D multi grid contact dection.

This function take the basic informations of grid configuration, information of all AABB3Ds the detection perform on and a custom contacting box pair handler as input parameter. It perform the 3D Multi Grid Contact detection algorithm automatically and return the contacting pair count if no error occur, or error code other wise.

Parameters:

grids A Float array indecating the cell size of each grid

gridCount An integer indecation the number of grids

elem A **AABB3D** (p. 11) array, describing the position and size of all objects

elemCount Number of AABB3Ds

mstr A string describing what detection algorithm each grid should use -"NBS" : Use NBS algorithm for each grid. -"SCR" : Use Screening algorithm for each grid. -"AUTO" : Default configuration, in this implementation, NBS will be used. -custom : User can provide a method configuration string to customize what algorithm each grid should use. The length of method cofiguration string should equals to gridCount, and each character should be 'a', 'n' or 's', for each representing a selection of algorithm:

1. 'a' : Automatic selection, in this implementaion NBS will be used
2. 's' : Screening
3. 'n' : NBS eg. "asn": this method configuration string tell the MGCD library automatically select a detection algorithm for the finest grid, use NBS for the coarsest grid, and use Screening for the middle grid.

func A custom contacting box pair handler, can be NULL

context Custom data structure used by handler

Returns:

This function return the contacting 3D box pair count if no error occur, or else return a error code who is a negative. -MGCD_METHOD_ERROR Invalid methods configuration -MGCD_GRID_ERROR Invalid grid configuration, eg. invalid or duplicate cell size or can not find a grid to fit a box in -MGCD_BOX_ERROR invalid box information, eg. negative size of a box

See also:

AABB3D (p. 11), MGCDPairHandler3D

Index