Shapoid

P. Baillehache

October 30, 2018

Contents

1 Definitions **1**
- 1.1 Transformation . . . 2
- 1.2 Shapoid's coordinate system . . . 2
- 1.3 Insideness . . . 3
- 1.4 Bounding box . . . 3
- 1.5 Depth and Center . . . 4
- 1.6 Iterator on Spheroid . . . 5
- 1.7 Collision detection of Spheroid . . . 6

2 Interface **7**

3 Code **7**
- 3.1 shapoid.c . . . 7
- 3.2 shapoid-inline.c . . . 7

4 Makefile **7**

5 Unit tests **7**

6 Unit tests output **7**

Introduction

Shapoid is a C library providing the `Shapoid` structure and its functions which can be used to manipulate Shapoid objects (see next section for details).

It also provides the `ShapoidIter` structure and its functions which can be used to sequantially loop through the surface/volume/... of a `Shapoid`.

It uses the `PBErr`, `PBMath` and `GSet` libraries.

1 Definitions

A Shapoid is a geometry defined by its dimension $D \in \mathbb{N}_+^*$ equals to the number of dimensions of the space it exists in, its position $\overrightarrow{P}$, and its axis $(\overrightarrow{A_0}, \overrightarrow{A_1}, ..., \overrightarrow{A_{D-1}})$. A_i and P are vectors of dimension D. In what follows I'll use I as notation for the interval $[0, D-1]$ for simplification.

Shapoids are classified in three groups: Facoid, Pyramidoid and Spheroid. The volume of a Shapoid is defined by, for a Facoid:

$$\left\{ \sum_{i \in I} v_i \overrightarrow{A_i} + \overrightarrow{P} \right\}, v_i \in [0.0, 1.0] \tag{1}$$

for a Pyramidoid:

$$\left\{ \sum_{i \in I} v_i \overrightarrow{A_i} + \overrightarrow{P} \right\}, v_i \in [0.0, 1.0], \sum_{i \in I} v_i \leq 1.0 \tag{2}$$

and for a Spheroid:

$$\begin{array}{c} \left\{ \sum_{i \in I} v_i \overrightarrow{A_i} + \overrightarrow{P} \right\}, \\ v_i \in [-0.5, 0.5], \quad \sum_{i \in I} v_i^2 \leq 0.25 \end{array} \tag{3}$$

1.1 Transformation

A translation of a Shapoid by $\overrightarrow{T}$ is obtained as follow:

$$\left(\overrightarrow{P}, \left\{ \overrightarrow{A_i} \right\}_{i \in I} \right) \mapsto \left(\overrightarrow{P} + \overrightarrow{T}, \left\{ \overrightarrow{A_i} \right\}_{i \in I} \right) \tag{4}$$

A scale of a Shapoid by $\overrightarrow{S}$ is obtained as follow:

$$\left(\overrightarrow{P}, \left\{ \overrightarrow{A_i} \right\}_{i \in I} \right) \mapsto \left(\overrightarrow{P}, \left\{ \overrightarrow{A'_i} \right\}_{i \in I} \right) \tag{5}$$

where

$$\overrightarrow{A'_i} = S_i \overrightarrow{A_i} \tag{6}$$

For Shapoid whose dimension D is equal to 2, a rotation by angle θ is obtained as follow:

$$\left(\overrightarrow{P}, \overrightarrow{A_0}, \overrightarrow{A_1}\right) \mapsto \left(\overrightarrow{P}, \overrightarrow{A_0'}, \overrightarrow{A_1'}\right) \tag{7}$$

where

$$\overrightarrow{A_i'} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \overrightarrow{A_i} \tag{8}$$

1.2 Shapoid's coordinate system

The Shapoid's coordinate system is the system having $\overrightarrow{P}$ as origin and $\overrightarrow{A_i}$ as axis. One can change from the Shapoid's coordinate system ($\overrightarrow{X^S}$) to the standard coordinate system ($\overrightarrow{X}$) as follow:

$$\overrightarrow{X} = \left[\left(\overrightarrow{A_0}\right)\left(\overrightarrow{A_1}\right)...\left(\overrightarrow{A_{D-1}}\right)\right]\overrightarrow{X^S} + \overrightarrow{P} \tag{9}$$

and reciprocally, from the standard coordinate system to the Shapoid's coordinate system:

$$\overrightarrow{X^S} = \left[\left(\overrightarrow{A_0}\right)\left(\overrightarrow{A_1}\right)...\left(\overrightarrow{A_{D-1}}\right)\right]^{-1}\left(\overrightarrow{X} - \overrightarrow{P}\right) \tag{10}$$

1.3 Insideness

$\overrightarrow{X}$ is inside the Shapoid S if, for a Facoid:

$$\forall i \in I, 0.0 \leq X_i^S \leq 1.0 \tag{11}$$

for a Pyramidoid:

$$\begin{cases} \forall i \in I, 0.0 \leq X_i^S \leq 1.0 \\ \sum_{i \in I} X_i^S \leq 1.0 \end{cases} \tag{12}$$

for a Spheroid:

$$\left|\left|\overrightarrow{X^S}\right|\right| \leq 0.5 \tag{13}$$

1.4 Bounding box

A bounding box of a Shapoid is a Facoid whose axis are colinear to axis of the standard coordinate system, and including the Shapoid in its volume. While the smallest possible bounding box can be easily obtained for Facoid and Pyramidoid, it's more complicate for Spheroid. Then we

will consider for the Spheroid the bounding box of the equivalent Facoid $\left(\overrightarrow{P} - \sum_{i\in I}\left(0.5 * \overrightarrow{A_i}\right), \left\{\overrightarrow{A_i}\right\}_{i\in I}\right)$ which gives the smallest bounding box when axis of the Spheroid are colinear to axis of the standard coordinate system and a bounding box slightly too large when not colinear.
The bounding box is defined as follow, for a Facoid:

$$\left(\overrightarrow{P'}, \left\{\overrightarrow{A'_i}\right\}_{i\in I}\right) \tag{14}$$

where

$$\begin{cases} P'_i = P_i + \sum_{j\in I^-} A_{ji} \\ A'_{ij} = 0.0, i \neq j \\ A'_{ij} = \sum_{k\in I^+} A_{kj} - \sum_{k\in I^-} A_{kj}, i = j \end{cases} \tag{15}$$

and, I^+ and I^- are the subsets of I such as $\forall j \in I^+, A_{ij} \geq 0.0$ and $\forall j \in I^-, A_{ij} < 0.0$.

for a Pyramidoid:

$$\left(\overrightarrow{P'}, \left\{\overrightarrow{A'_i}\right\}_{i\in I}\right) \tag{16}$$

where

$$\begin{cases} P'_i = P_i + Min\left(Min_{j\in I}(A_{ji}), 0.0\right) \\ A'_{ij} = 0.0, i \neq j \\ A'_{ij} = Max_{k\in I}(A_{kj}) - Min_{k\in I}(A_{kj}), i = j \end{cases} \tag{17}$$

1.5 Depth and Center

Depth $\mathbf{D}_S(\overrightarrow{X})$ of position $\overrightarrow{X}$ a Shapoid S is a value ranging from 0.0 if $\overrightarrow{X}$ is on the surface of the Shapoid, to 1.0 if $\overrightarrow{X}$ is at the farthest location from the surface inside the Shapoid. Depth is by definition equal to 0.0 if $\overrightarrow{X}$ is outside the Shapoid. Depth is continuous and derivable on the volume of the Shapoid. It is defined by, for a Facoid:

$$\mathbf{D}_S(\overrightarrow{X}) = \prod_{i\in I}\left(1.0 - 4.0 * (0.5 - X_i^S)^2\right) \tag{18}$$

for a Pyramidoid:

$$\mathbf{D}_S(\overrightarrow{X}) = \prod_{i\in I}\left(1.0 - 4.0 * \left(0.5 - \frac{X_i^S}{1.0 - \sum_{j\in I-\{i\}} X_j^S}\right)^2\right) \tag{19}$$

and for a Spheroid:

$$\mathbf{D}_S(\overrightarrow{X}) = 1.0 - 2.0 * \left|\left|\overrightarrow{X^S}\right|\right| \tag{20}$$

The maximum depth is obtained at $\overrightarrow{C}$ such as, for a Facoid:

$$\forall i \in I, C_i^S = 0.5 \tag{21}$$

for a Pyramidoid:

$$\forall i \in I, C_i^S = \frac{1}{D+1} \tag{22}$$

for a Spheroid:

$$\forall i \in I, C_i^S = 0.0 \tag{23}$$

$\overrightarrow{C}$ is called the center of the Shapoid.

1.6 Iterator on Spheroid

While a sequential path through a Facoid and a Pyramidoid is obvious, path through a Spheroid is more complex. The solution implemented is described below.

Given a Spheroid of dimension N we start from an arbitrary position: $< 0, 0, ..., -0.5 >$. From there we step the axis starting from the first one. If we could step an axis the step algorithm stops and return the new position as it could successfully step. However, if we could step on an axis other than the first one, it means we have modified the constraint for previous axis, the constraint being "is inside the spheroid". Then we reposition the axis before the stepped one to its lower possible value. It will allow it to step again at the next iteration on a new boundary defined by other axis values, and this scale up naturally to any dimension. Care must be care to the case when an axis reaches its upper value: the delta given by the user and the influence of other axis value make it jumps "over" the boundary in most cases. To keep things neat and clean we recalculate the exact value of the axis for its last step instead of using the delta given by the user.

The calculation of the lower and upper values of an axis given the values of other axis can be performed as follow:

Lets note $\overrightarrow{P} =< p_0, p_1, ..., p_{N-1} >$ the position in a Spheroid of dimension N. A position will be on the boundary of the Spheroid if and only if $||\overrightarrow{P}|| = 0.5$. We want to calculate α which bring a position $\overrightarrow{P'}$ inside the Spheroid to a position $\overrightarrow{P}$ on the boundary of the Spheroid by modifying the axis i (i.e. $p_i = p'_i + \alpha$ and $p_j = p'_j, j \neq i$). Lets note $n = ||\overrightarrow{P'}||$. We have:

$$p'^2_0 + p'^2_1 + ... + p'^2_i + ... + p'^2_{N-1} = n^2 \tag{24}$$

and

$$p^2_0 + p^2_1 + ... + p^2_i + ... + p^2_{N-1} = 0.25 \tag{25}$$

equivalent to

$$p'^2_0 + p'^2_1 + ... + (p'_i + \alpha)^2 + ... + p'^2_{N-1} = 0.25 \tag{26}$$

by substracting (24) and (26) we have

$$p'^2_i - (p'_i + \alpha)^2 = n^2 - 0.25 \tag{27}$$

equivalent to

$$p'^2_i - (p'^2_i + \alpha^2 + 2p'_i\alpha) = n^2 - 0.25 \tag{28}$$

equivalent to

$$-\alpha^2 - 2p'_i\alpha - (n^2 - 0.25) = 0 \tag{29}$$

simplified to

$$\alpha^2 + 2p'_i\alpha + (n^2 - 0.25) = 0 \tag{30}$$

This quadratic equation can be solved directly to obtain α:

$$\alpha = \frac{-2p'_i \pm \sqrt{4p'^2_i - 4(n^2 - 0.25)}}{2} \tag{31}$$

Which gives the two solutions defining the lower and upper boundaries of the Spheroid on the axis i.

This result can then be used to solve our problem with what I'll call the "Wormy Algorithm":

This algorithm step P to the next position in the path by *delta* and returns true if we haven't reached the end of the path, or false if we have reached the end of the path (i.e. if we have iterated through all the surface/volume/... of the Spheroid). Remember that P must be initialised to $< 0, 0, ..., -0.5 >$ as the beginning of the path, and that `ShapoidIter` iterates coordinates in the Shapoid's coordinate system.

1.7 Collision detection of Spheroid

The detection of collision between two Spheroids is done as follow. One of the spheroid is converted into the coordinates system of the other and checked against a circle of radius 0.5 centered at the origin of the system. By checking that the position of the converted Spheroid is less than its minor radius plus 0.5 or more than its major radius plus 0.5, the trivial cases of, respectively, interesection and non intersection can be performed. In other cases an incremental search from the center of the converted Spheroid toward the nearest point to the origin inside this Spheroid is performed. The intersection can then be checked by testing if the distance of this point to the origin is less or equal than 0.5.

2 Interface

3 Code

3.1 shapoid.c

3.2 shapoid-inline.c

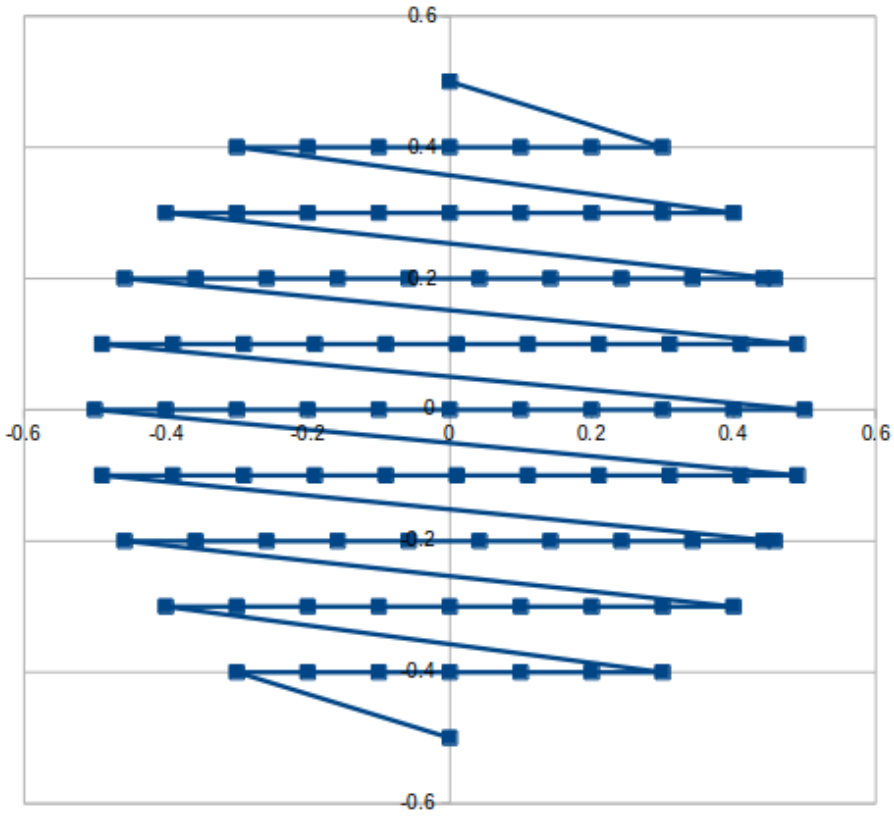
4 Makefile

5 Unit tests

6 Unit tests output

facoid.txt

Example of path on a 2D Spheroid using the ShapoidIterator:


0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.6
-0.4
-0.2
0
0.2
0.4
0.6