Vol. 12, No. 1: 3–8

On Faster Sphere-Box Overlap Testing

Thomas Larsson
Mälardalen University

Tomas Akenine-Möller
Lund University

Eric Lengyel
Terathon Software

Abstract. We present faster overlap tests between spheres and either axis-aligned or oriented boxes. By utilizing quick rejection tests, faster execution times are observed compared to previous techniques. In addition, we present alternative vectorized overlap tests, which are compared to the sequential algorithms. Source code is available online.

1. Introduction

In this paper, we present faster sphere-AABB and sphere-OBB overlap tests by improving on Arvo's orignal method [Arvo 90]. Conservative versions of these overlap tests are also discussed. We note that when choosing a bounding volume (BV) type, there is frequently a trade-off between tightness and operation costs. Hence, the overlap tests we present can make it more attractive to use mixed types of BVs in, for example, collision detection.

2. Overlap Tests

We represent a sphere by its center point c and radius r and an AABB by two points *min* and *max* holding the minimal and maximal coordinate values of

1086-7651/07 $0.50 per page

the box, respectively. Arvo's original method minimizes the distance from c to the closest point on the AABB using the following compact algorithm:

```
    OverlapSphereAABB_Arvo($c$, $r$, $min$, $max$)
1.    $d \leftarrow 0$
2.    for each $i \in \{x, y, z\}$
3.        if $(c_i < min_i)$
4.            $e \leftarrow c_i - min_i$
5.            $d \leftarrow d + e^2$
6.        else if $(c_i > max_i)$
7.            $e \leftarrow c_i - max_i$
8.            $d \leftarrow d + e^2$
9.    if $(d <= r^2)$ return true
10.   return false
```

Surprisingly, since Arvo's paper, no one has attempted to improve his algorithm, and we have found that it can be made faster by incorporating simple rejection tests into Arvo's code. The idea behind the rejection tests is simply to enlarge the box extents by r and then check if c is inside the enlarged box or not. The improved code is shown below, where our rejection tests are on lines 4 and 7:

```
    OverlapSphereAABB_QRI($c$, $r$, $min$, $max$)
1.    $d \leftarrow 0$
2.    for each $i \in \{x, y, z\}$
3.        if $((e \leftarrow c_i - min_i) < 0)$
4.            if $(e < -r)$ return false
5.            $d \leftarrow d + e^2$
6.        else if $((e \leftarrow c_i - max_i) > 0)$
7.            if $(e > r)$ return false
8.            $d \leftarrow d + e^2$
9.    if $(d <= r^2)$ return true
10.   return false
```

We abbreviate this version of the algorithm as QRI (quick rejections intertwined). As another alternative, instead of trying quick rejections as the exact squared minimal distance calculations proceeds, all the rejection tests can be inserted first in Arvo's original method. We call this version of the algorithm QRF (quick rejections first).

Converting Arvo's method to the oriented box case is a simple matter of translating the distance calculations to take place in the box's own frame-of-reference. The OBB is described by a midpoint *mid*, three half side extents *ext*, and three orthogonal unit vectors *axes* defining the orientation. Then the vector $v = mid - c$ is projected onto the *axes* of the box in turn to obtain the distance between the center of the box and the center of the sphere along the oriented directions. We call the resulting algorithm G-Arvo, since it is generalized to the oriented case. Again, the same type of early rejection tests as described above can be performed by taking the rotated coordinate system into account, which again gives us a QRI and a QRF version. We refer to the source code available online at the address listed at the end of this paper for implementation details of these overlap tests.

Furthermore, we note that hierarchical approaches, which refine the search results in each level, can also possibly benefit from using conservative overlap tests. A conservative overlap test is such that sometimes a sphere-box overlap is reported when in fact there is no overlap. This is often referred to as a *false positive.* Note that such false positives do not affect the final search result, since this only means that testing will continue on child nodes, further down in the hierarchies, until a final correct result is obtained. When two objects are overlapping but the test returns that they are not, you have a *false negative.* This is not permitted in the conservative test, since then no guarantees can be given on the correctness of the search result.

The point behind using a conservative test is of course to speed up the overall search procedure. Therefore, it should be considerably faster, while still reporting few false positives. Good examples of this are the SAT lite test used previously for box-box overlap tests [van den Bergen 97, Larsson and Akenine-Möller 03] and the overlap test between k-DOPs [Klosowski et al. 98]. In our cases, i.e., for both the sphere-AABB and sphere-OBB tests, it seems most reasonable to simply perform the quick rejection tests mentioned above, and when a test fails to determine the overlap status, overlap is assumed.

3. Branch Elimination and Vectorization

Although the above presented solutions speed up the sphere-box overlap tests in most cases, today's hardware offers other alternative optimization possibilities. Since our early rejection tests add branches to an algorithm that is already branch-intensive, it might be advantageous to reformulate the algorithm slightly in order to eliminate branches. For example, the comparisons against zero in the QRI algorithm above can be eliminated as follows:

```
1.  $d \leftarrow 0$
2.  **for each** $i \in \{x, y, z\}$
3.      $e \leftarrow \max(min_i - c_i, 0) + \max(c_i - max_i, 0)$
4.      **if** $(e <= r)$ **return false**
5.      $d \leftarrow d + e^2$
6.  **if** $(d <= r^2)$ **return true**
7.  **return false**
```

Of course, this requires hardware that has a floating-point max instruction. Fortunately, such an instruction is widely available in vector instruction sets such as SSE and AltiVec. However, using a vectorized algorithm goes far beyond providing the max instruction. It can also be used to eliminate the loop over the three axes and perform all three iterations simultaneously. This being the case, an early rejection test is no longer needed in a vectorized version. In the accompanying source code, we provide such implementations using the vectorized max function, similar to the pseudocode above. Note that the primary calculation is performed for all three axes simultaneously and that there is only one branch at the very end of the function. A corresponding SSE version for the sphere-OBB case is also provided. An AltiVec implementation would be very similar.

4. Results

All benchmarks were run on several platforms, and we report the results from two of them, which we believe capture architectural performance differences on current hardware. The first platform, M_1, was a Pentium M, 1.4 GHz, laptop. The second platform, M_2, was a Pentium 830 Dual-Core, 3 GHz, PC. In Table 1 and Table 2, the obtained results are given for the different versions of the sphere-AABB and sphere-OBB overlap tests, respectively. For each measurement, two million overlap tests were executed. The columns describe repeated runs for varying numbers of overlapping pairs in the test data. The percentage of false positives among the reported overlaps for the conservative test is given in the last line.

As can be seen, the SSE version of the algorithm is clearly fastest for both the AABB and OBB cases regardless of the overlap frequency in the test data. Among the sequential algorithms, however, the best method depends on the frequency of overlap in the test data. The QRF method seems preferable with an overlap frequency of less than approximately 50%, and otherwise

	Overlap frequency									
	5%		25%		50%		75%		95%	
Method	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
Arvo	172	146	178	156	169	148	148	128	121	91
QRI	112	112	138	148	142	149	134	134	113	97
QRF	100	92	128	120	147	132	159	131	157	116
SSE	86	78	91	98	94	105	91	95	89	77
Cons	82	61	85	62	83	58	84	50	82	42
Error	17%		11%		4.3%		1.6%		0.3%	

Table 1. Sphere-AABB overlap test benchmark. Times are in ms.

	Overlap frequency									
	5%		25%		50%		75%		95%	
Method	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2	M_1	M_2
G-Arvo	246	226	253	242	247	236	227	206	199	153
QRI	201	166	221	214	224	221	215	204	196	154
QRF	196	145	218	181	235	198	241	198	236	180
SSE	191	105	182	134	184	145	181	133	178	105
Cons	177	126	179	134	177	123	173	116	172	113
Error	17%		11%		4.6%		1.8%		0.3%	

Table 2. Sphere-OBB overlap test benchmark. Times are in ms.

the QRI method seems a better choice. Furthermore, the conservative test seems promising for hierarchical searching, since it is very fast in all cases without reporting an excessive number of false positives. Also, note that the overlap test cost for the sphere-OBB test is less than twice as expensive as the sphere-AABB test for almost all corresponding versions for all test cases.

Finally, although not tested here, we note that some applications may benefit from trying to exploit vectorization more fully by performing several overlap tests simultaneously. For example, in collision detection queries, using a mix of bounding volume hierarchies of spheres and boxes, it may be advantageous to test one sphere against four boxes, and four boxes against one sphere, simultaneously.

References

[Arvo 90] J. Arvo. "A Simple Method for Box-Sphere Intersection Testing." In *Graphics Gems*, edited by Andrew Glassner, pp. 335–339. San Diego, CA: Academic Press Professional, Inc., 1990.

[Klosowski et al. 98] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. "Efficient Collision Detection Using Bounding Volume Hierarchies of *k*-DOPs." *IEEE Transactions on Visualization and Computer Graphics* 4:1 (1998), 21–36.

[Larsson and Akenine-Möller 03] Thomas Larsson and Tomas Akenine-Möller. "Efficient Collision Detection for Models Deformed by Morphing." *The Visual Computer* 19:2-3 (2003), 164–174.

[van den Bergen 97] Gino van den Bergen. "Efficient Collision Detection of Complex Deformable Models using AABB Trees." *journal of graphics tools* 2:4 (1997), 1–14.

Web Information:

The source code is available at http://jgt.akpeters.com/papers/LarssonEtAl07/.

Thomas Larsson, Mälardalen University, IDE, PO Box 883, S-721 23 Västerås, Sweden (thomas.larsson@mdh.se)

Tomas Akenine-Möller, Department of Computer Science, Lund University, Box 118, 221 00 Lund, Sweden (tam@cs.lth.se)

Eric Lengyel, Terathon Software, 1986 Esterel Way, Sacramento, CA 95832 (lengyel@terathon.com)

Received December 5, 2005; accepted in revised form September 29, 2006.