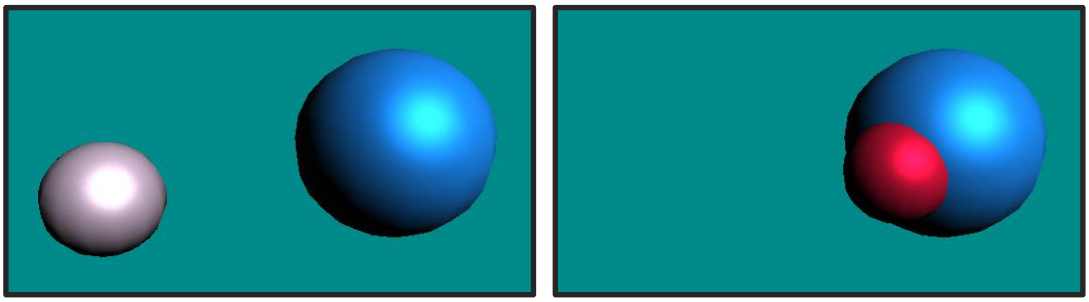
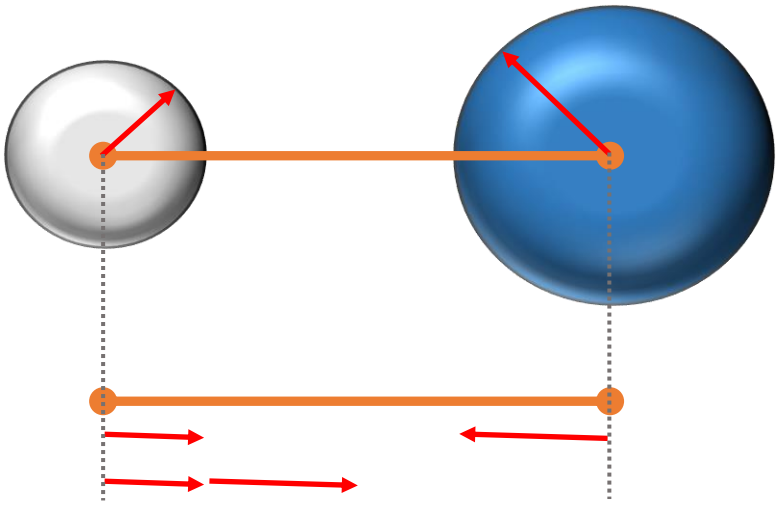
【衝突判定３Ｄ】球体と球体

３Ｄの衝突判定の中でも高速で、かつ、最も理解しやすい内容になっています。
それは、２Ｄの時と考え方や処理が、ほぼ同じだからです。

衝突していない　　衝突している



２つの円の中心座標の距離を計算して、



２つの円の半径距離の合計値より短ければ、衝突している、
ということになります。
（ 上図は、半径距離よりも短くないので、衝突していない ）

ただ、真面目に距離を測ろうとすると、
３平方の定理により、平方根を使う必要がありますが、
平方根は、他の演算に比べ、処理速度が遅いので極力使わないようにします。
（ 平方根を使わなくとも、大小関係は変わらないので使用しない
　本当に距離が必要であれば、平方根を使用する ）

考え方は、２Ｄと全く一緒ですね。

３Ｄになってもｚ軸が増えるだけになりますので、実装例を下記で紹介します。

```
bool AsoUtility::IsHitSpheres(
       const VECTOR& pos1, float radius1, const VECTOR& pos2, float radius2)
{
    // 球体同士の衝突判定
    bool ret = false;

    // お互いの半径の合計
    float radius = radius1 + radius2;

    // 座標の差からお互いの距離を取る
    VECTOR diff = VSub(pos2, pos1);

    // 三平方の定理で比較(SqrMagnitudeと同じ)
    float dis = (diff.x * diff.x) + (diff.y * diff.y) + (diff.z * diff.z);
    if (dis < (radius * radius))
    {
        ret = true;
    }

    return ret;
}
```