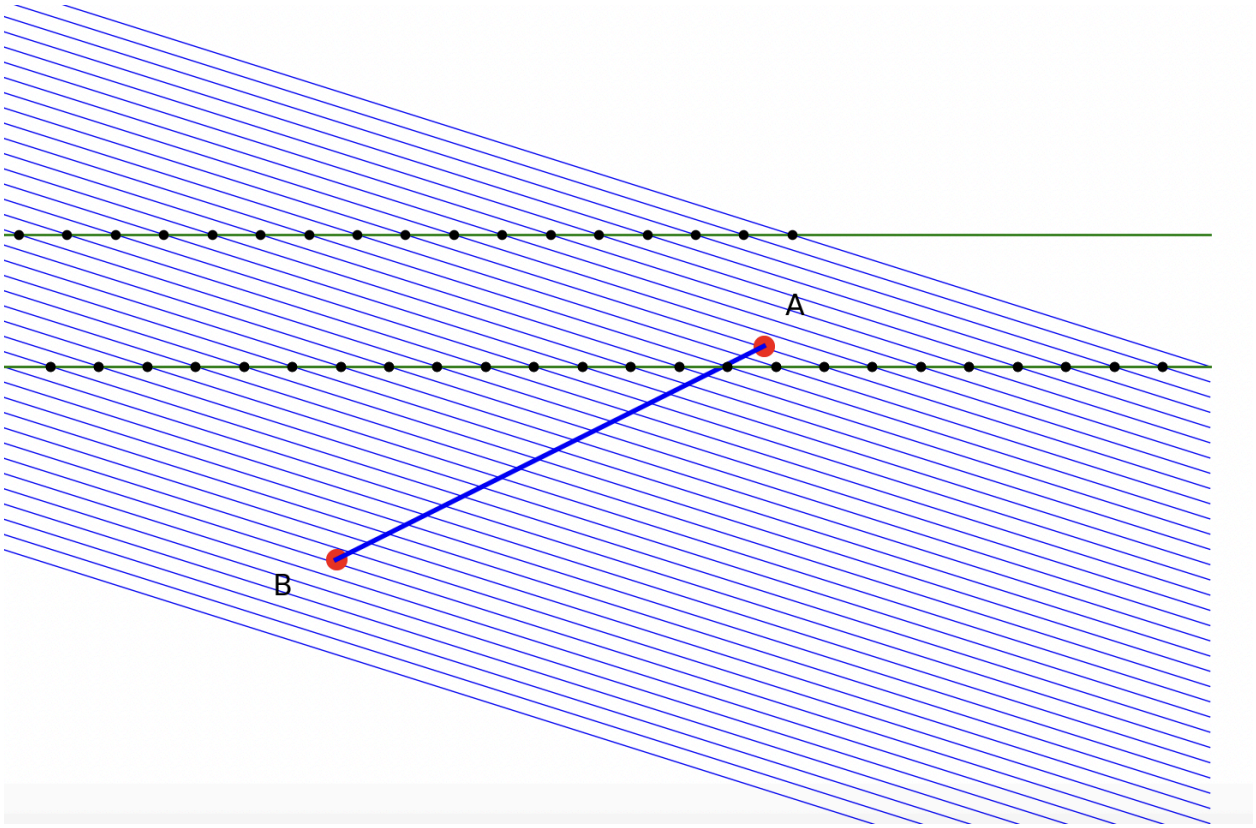
膜的说明

1.角度tan(150)
2.如图绿色线条的两个黑点之间为膜的宽度d的一半 halfD,分别代表左光柱和右光柱
3.膜所在的屏幕2688*1242和OpenGL的gl_FragCoord一致



纹理说明

纹理是指sbs的video的每一个frame，大小和屏幕2688*1242一致
即一个屏幕坐标下面只有一个纹理坐标

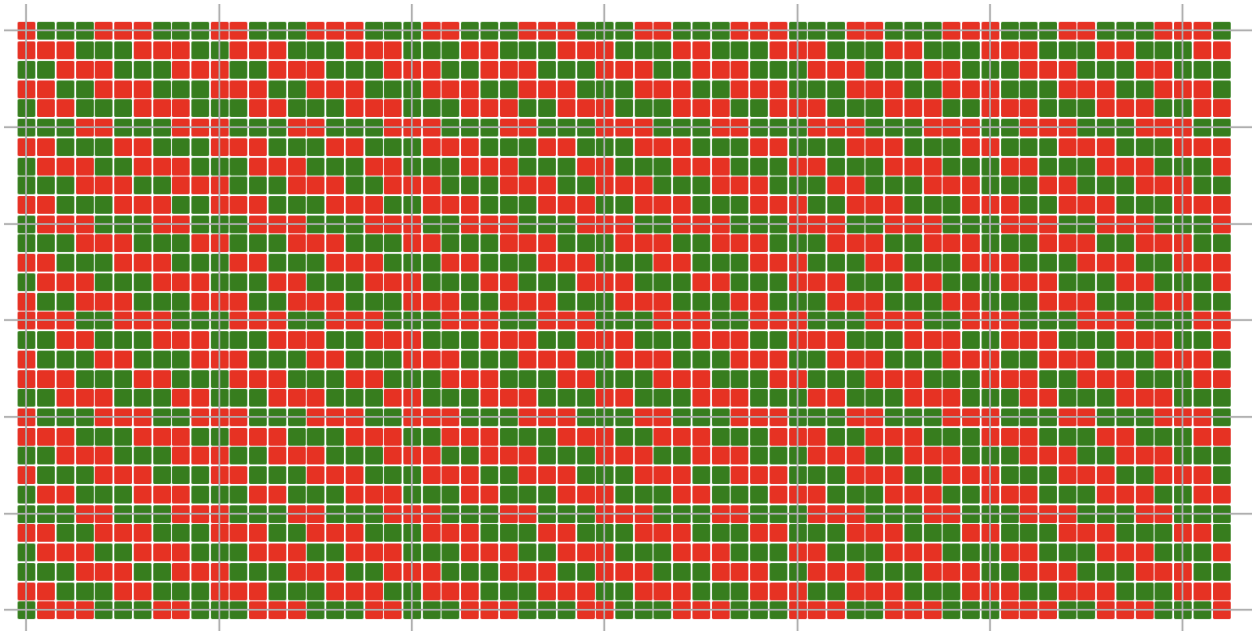
纹理算法说明

计算每个屏幕点对应的纹理坐标既可以显示video

w1 = mod((abs(uK * gl_FragCoord.x - gl_FragCoord.y + 5000.0 + uM) / sqrt(uK * uK + 1.0)),uD);

w1 是每个屏幕坐标对应的纹理坐标的值，如果w1<halfD则显示左图，反之显示右图 算法的结果就是每行基本都是3红3绿交错。

注意 此算法的结果就是保证了屏幕坐标和纹理坐标一一对应



如某一行前3个w1<halfD那他们都是显示左图

总之在此时，纹理坐标和屏幕坐标是绑定的，每个屏幕坐标对应的纹理坐标是固定的，该算法能把左右图准确分配到膜的左右。

为何要补偿值

在以上计算w1 = mod((abs(uK * gl_FragCoord.x - gl_FragCoord.y + 5000.0 + uM) / sqrt(uK * uK + 1.0)),uD);的过程中，默认uM=0

根据以上的计算，每根光柱刚好基本能覆盖3根左图或3根右图，由于膜的特点，这些左右图可以完美反射到左右眼此时效果最好

旋转：
当手机旋转时，原来刚好覆盖3个左图的光柱 由于旋转变成1根右图2根左图，或2根右图1根左图，此时反射到眼睛的就是有gap的

为了解决这个问题，需要引入uM，uM的值在0-halfD之间，目的是改变w1的值，例如uM=halfD，则红绿色互换

在旋转的过程中，可能光柱刚好包含1根右图2根左图，或2根右图1根左图范围时，需要用uM来补偿

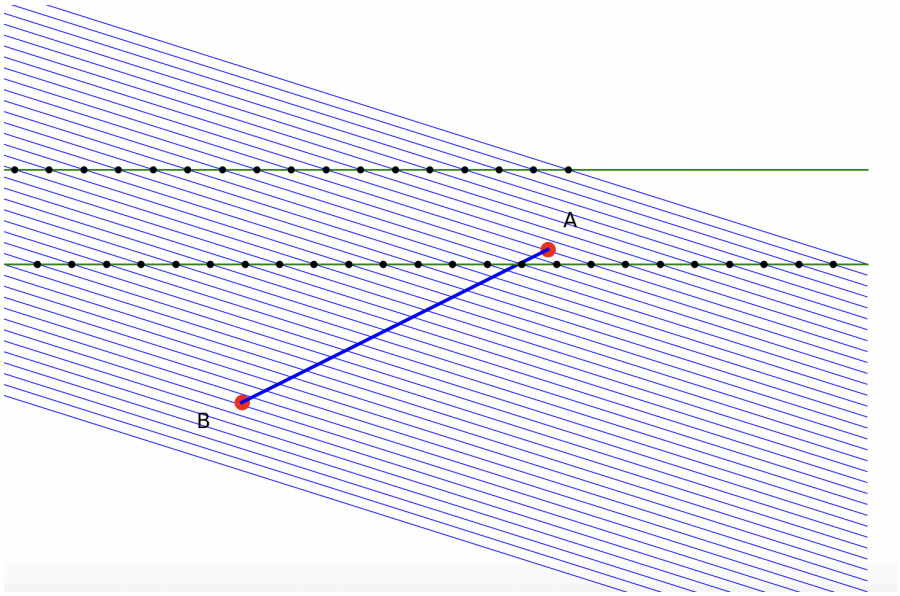
此时需要计算一个uM 使得w1变大或变小，从而改变左右图的位移，这就是目前要解决的问题

通过uM来改变w1的大小（是否大于halfD）来改变左右图的位移

w1 = mod((abs(uK * gl_FragCoord.x - gl_FragCoord.y + 5000.0 + uM) / sqrt(uK * uK + 1.0)),uD)

目前思路

1.先确定一个效果最佳的手机角度，此时uM为0，并保存该点的位置B
2.手机旋转时，确定另一个点的位置A
3.计算这2个点的位移，转换为uM




```
    float baseEyeDistance = hypot(_baseLeftPoint.x - _baseRightPoint.x,
_baseLeftPoint.y - _baseRightPoint.y);
    float currentEyeDistance = hypot(_currentLeftPoint.x - _currentRightPoint.x,
_currentLeftPoint.y - _currentRightPoint.y);

    _distanceRatio = currentEyeDistance / baseEyeDistance;

    float xMove = fabs((_currentEyePoint.x - _baseEyePoint.x) * _distanceRatio);
    float yMove = fabs((_baseEyePoint.y - _currentEyePoint.y) * _distanceRatio);

    float xRate = xMove/_myDelegate.uMRatio;
    float yRate = yMove/(-uK)/_myDelegate.uMRatio;

    float xShouldMove = xRate*halfD;
    float yShouldMove = yRate*halfD;
    float shouldMove = fmodf((xShouldMove+yShouldMove) , halfD);
```

```
    //  shouldMove = {0,halfD} 则 fmod(shouldMove,halfD) = shouldMove
    //    highp float w1 = mod((abs(uK * gl_FragCoord.x - gl_FragCoord.y + 5000.0
+ uM) / sqrt(uK * uK + 1.0)),uD);
    //    highp float w2 = uD - w1;

    NSLog(@"_baseEyePointX: %f,_baseEyePointY: %f,_currentEyePointX:
%f,_currentEyePointY: %f,eyeLength: %f,eyeCurrentDistance: %f,distanceRatio:
%f
",_baseEyePoint.x,_baseEyePoint.y,_currentEyePoint.x,_currentEyePoint.y,_bas
eEyeDistance,_currentEyeDistance,_distanceRatio);

    NSLog(@"X 轴上的移动: %f,Y 轴上的移动:%f,xRate: %f,yRate:%f,
xShouldMove: %f,yShouldMove:%f ,
shouldMove:%f,lastMove:%f",xMove,yMove,xRate,yRate,xShouldMove,yShould
Move,shouldMove,_lastshouldMov);

    if (fabs(shouldMove - _lastshouldMov)  < halfD*0.5){
        NSLog(@" < halfD*0.5)");
        return;
    }
```

现状:感觉还是没算准uM, 还是会有gap