

开发环境:

CentOS7 x64, maya2015x64, CMake, CodeBlocks

命令算法说明:

```
void peltOverlap::numOverlapUVFaces(const MString& shadingGroup, MStringArray&
flattenFaces)
```

```
{
    MFloatArray face1Orig, face1Vec, face2Orig, face2Vec, center, radius;

    // Loop through face i
    unsigned int numOverlap = 0;
    createBoundingCircle(flattenFaces, center, radius);

    for(unsigned int i = 0; i < flattenFaces.length() && numOverlap < fNthPairs; i++)
    {
        // for face[i]
        //construct point and edge(line section) for face[i]
        if(!createRayGivenFace(flattenFaces[i], face1Orig, face1Vec))
            continue;

        // center point of face[i]
        const float cui = center[2*i];
        const float cvi = center[2*i+1];

        //radius of face[i]
        const float ri = radius[i];

        //test face[i] and other faces
        for(unsigned int j = i+1; j < flattenFaces.length() && numOverlap < fNthPairs; j++)
        {
            // center point of face[j]
            const float &cuj = center[2*j];
            const float &cvj = center[2*j+1];

            //radius of face[j]
            const float &rj = radius[j];

            // Quick rejection if bounding circles don't overlap
            // 如果两个线段的中心点的距离大于它们的外接圆的半径和, 则这两个线段不可能
            相交

            float du = cuj - cui;
            float dv = cvj - cvi;
            float dsqr = du*du + dv*dv;
            if (dsqr >= (ri+rj)*(ri+rj))
                continue;

            //construct point and edge(line section) for face[j]
            if(!createRayGivenFace(flattenFaces[j], face2Orig, face2Vec))
```

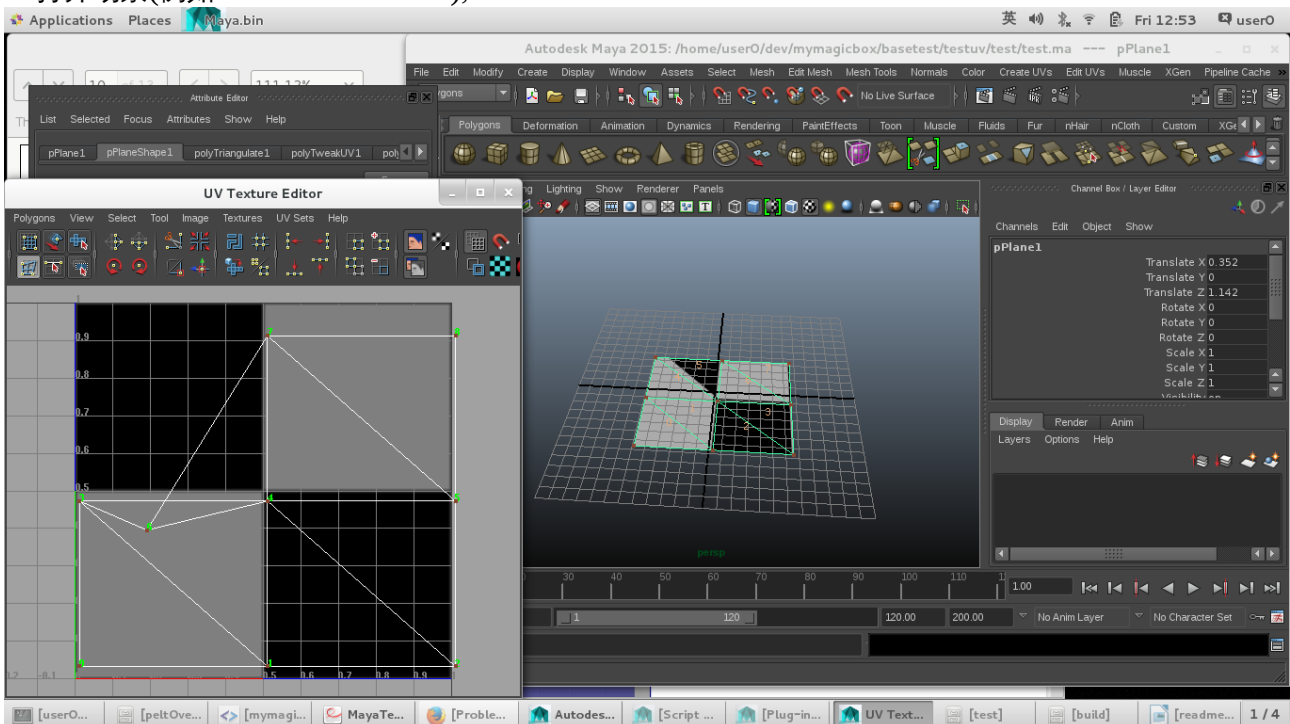
continue;

```
// test if the two line sections cross with each other
```

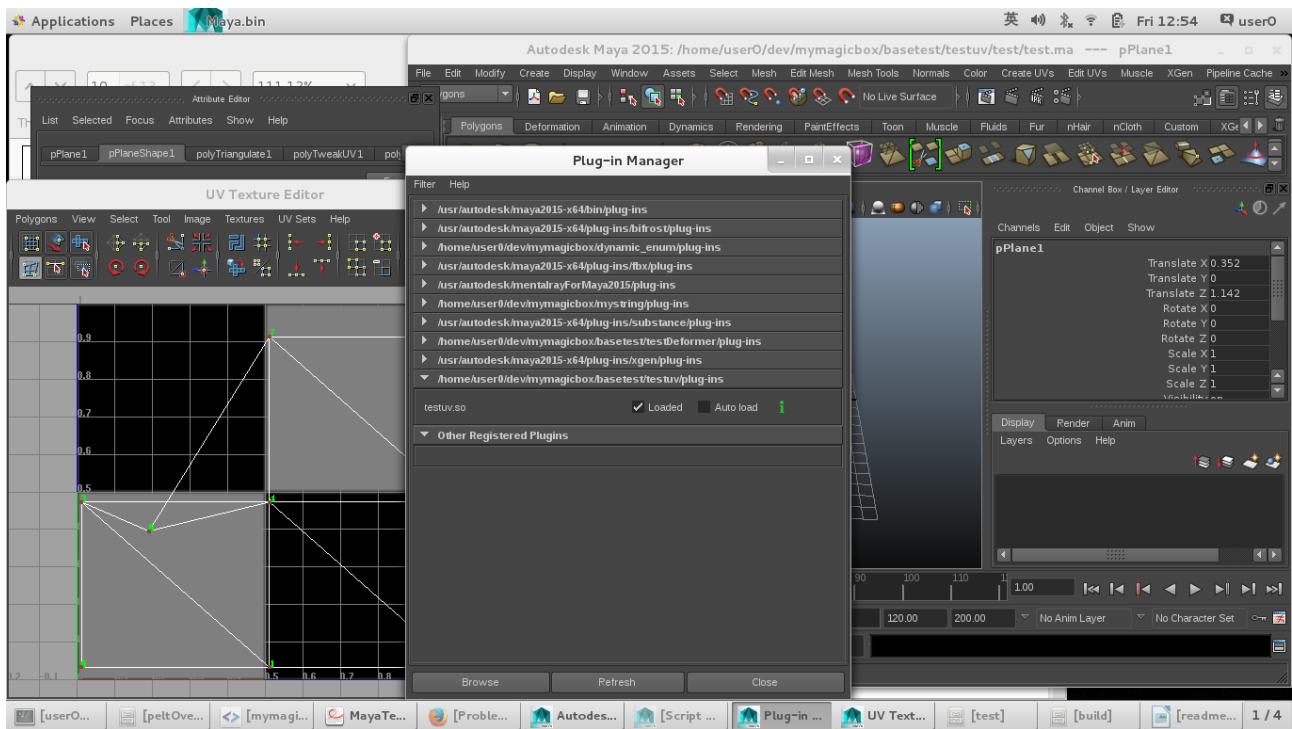
```
if (checkCrossingEdges(face1Orig, face1Vec, face2Orig, face2Vec)) {  
    numOverlap++;  
    appendToResult(flattenFaces[i]);  
    appendToResult(flattenFaces[j]);  
    continue;  
}  
}  
}  
}
```

使用说明:

1. 打开场景(例如 testuv/test/test.ma),



2. 加载插件 testuv.so



3. source 下面的 mel 函数(这些函数可以在 testuv/scripts/utility.mel 里找到)

```
proc string[] getSGsFromShape( string $shape )
{
    string $shadingEngines[];
    if ( `objExists $shape` )
    {
        string $dest[] = `listConnections -destination true -source false
        -plugins false -type "shadingEngine" $shape`;

        // listConnections can return duplicates within its list.
        // The select below is a quick trick to avoid dupes in the
        // returned array.
        if ( size( $dest ) )
        {
            string $select[] = `ls -sl`;
            select -r -ne $dest;
            $shadingEngines = `ls -sl`;
            select -r $select;
        }
    }
    return $shadingEngines;
}
```

```
proc selectTheOverlapFaces(string $sgName)
{
    select -cl;
    string $faces[] = `peltOverlapCmd $sgName`;
    select -cl;
    for($f in $faces)
    {
        select -tgl $f;
    }
}
```

```
}  
}
```

//Example:

```
//string $shadingEngines[] = getSGsFromShape("pPlaneShape1");
```

```
//selectTheOverlapFaces($shadingEngines[0]);
```

4. 输入参数是 mesh 名(比如"pPlaneShape1"):

```
string $shadingEngines[] = getSGsFromShape("pPlaneShape1");
```

```
selectTheOverlapFaces($shadingEngines[0]);
```

5. 结果是如果 uv 有穿插, 那些面片就会被选择

