
Algorithm 1: Preprocessing of skeleton animation data

Input:

skeleton;

animations;

Output:

BonesAffineMatrixs ;

```

1 for animation = animations[animType][frameIndex] do
2   Update skeleton by animation;
3   for bone = skeleton[boneIndex] do
4     matrix = bone.affineMatrix
5     while bone have parent do
6       bone = bone.parentBone;
7       matrix = matrix × bone.affineMatrix
8     BonesAffineMatrixs[animType][frameIndex][boneIndex] = matrix;

```

Algorithm 2: Realization of diversity of crowd animation

Input:

vertexInf={ *position*, *boneIndex* ,*coordinateUV* }

avatarParameter={
 affineMatrix,
 animationType,
 animationSpeed,
 }

BonesAffineMatrixs ;

animationPlayTime ;

textureMapping ;

Output:

vextexScenePosition ;

```

1 if bones[boneIndex] have animation then
2   |   numberOfPlayedFrames=rounding( animationPlayTime*animationSpeed );
3   |   frameIndex = numberOfFramesPlayed mod frameIndexMax;
4 else
5   |   frameIndex = 0;
6 boneMatrix = BonesAffineMatrixs[animationType][frameIndex];
7 vertexScenePosition = affineMatrix × boneMatrix × position;

```

Algorithm 3: Partition mapping of avatar

Input:

vertexInf={ *position*, *coordinateUV*}

avatarParameter={

headTextureType,

upperBodyTextureType,

trousersTextureType,

neckHeight,

waistHeight,

}

textureMapping ;

Output:

texturePixel ;

1 **if** *position* < *waistHeight* **then**

2 | *textureType* = *headTextureType*;

3 **else if** *position* < *neckHeight* **then**

4 | *textureType* = *upperBodyTextureType*;

5 **else**

6 | *textureType* = *trousersBodyTextureType*;

7 *texturePixel* \leftarrow *textureMapping*[*textureType*][*coordinateUV*];
