Algorithm 1: Preprocessing of skeleton animation data

Input:

skeleton;

animations;

Output:

BonesAffineMatrixs ;

```
1 for frame ← animations[animType][frameIndex] do
2     Update skeleton by frame;
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 crowd diversity animation

Input:

modelGeometry;

crowdParameter;

BonesAffineMatrixs ;

animationPlayTime ;

Output:

crowdGeometry ;

---

for *avatarParameter* $\in$ *crowdParameter* **do**
  GET {affineMatrix, animationType, animationSpeed }
  FROM avatarParameter.
  for *vertexInf* $\in$ *modelGeometry* **do**
    GET {position, boneIndex, coordinateUV }
    FROM vertexInf.
    if *bones*[*boneIndex*] *have animation* **then**
      numberOfPlayedFrames $\leftarrow$ rounding(animationPlayTime $\times$ animationSpeed);
      $frameIndex \leftarrow numberOfFramesPlayed \ mod \ frameIndexMax$;
    else
      $frameIndex \leftarrow 0$;
    $animationMatrix \leftarrow BonesAffineMatrixs[animationType][frameIndex][boneIndex]$;
    $vextexScenePosition \leftarrow avatarMatrix \times animationMatrix \times position$;
    $crowdGeometry.push(vextexScenePosition)$;

Algorithm 3: Partition binding of avatar texture map

Input:

modelGeometry;

crowdParameter;

textureMapping;

Output:

crowdMaterial ;

---

for $avatarParameter \in crowdParameter$ **do**
  GET {position, coordinateUV}
  FROM avatarParameter.
  for $vertexInf \in modelGeometry$ **do**
    GET { $textureType_{\{head,upperBody,trousers\}}$, $height_{\{neck,waist\}}$ }
    FROM vertexInf.
    if $position < height_{waist}$ **then**
      $type \leftarrow textureType_{head}$;
    else if $position < height_{neck}$ **then**
      $type \leftarrow textureType_{upperBody}$;
    else
      $type \leftarrow textureType_{trousers}$;
    $crowdMaterial.push(textureMapping[type][coordinateUV])$;