algorithm 1 Lightweight rendering of large crowd

InputData:

vertexInf={ $position$, UV, $skinIndex$ }//Information about the current vertex
instanceObjInf={ //Information about the instantiated objects
    $matrix$,
    $animationStyle$,//The type and speed of animation
    $textureStyle$,//Types and tones of maps
}
uniformInf={ //Data shared by all instantiated objects
    $skelontonDate$,//Bone data
    $time$,//Time is used to calculate the current frame number
}

if $bone(vertexInf.skinIndex) have\ animation$ **then**
    $frameIndex = time * speed\ mod(frameIndexMax + 1)$;
    $address0 = addressGet1(skinIndex, type, frameIndex)$;
else
    $address0 = addressGet2(skinIndex, type)$;
end if
$matrix1 \leftarrow skelontonDate\ at\ position\ address0$;
$matrix2 \leftarrow instanceObjInf.matrix$;
$glPosition = modelViewProjectionMatrix * matrix2 * matrix1 * position$;
$judgeArea()$; //Determine which part of the body the current vertex is in

OutData:

glPosition //The current vertex corresponds to the position on the screen
sendFragmentShader={ //Information sent to fragmentshader
    $areaType$,//The area where the current point is located
    UV, $textureType$,//Map type
    $color$,//For tone adjustment
}