```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class VPMesh : MonoBehaviour
{
    // this is the camera that will be viewing the 3D mesh used as the display
    public Camera viewingCamera;

    // this is the rendertexture from the camera that is viewing the 'virtual' scene that will be displayed on the 3D mesh
    public RenderTexture renderTextureDisplayed;

    // this is the mesh renderer of the display which will be showing the rendertexture
    public MeshRenderer meshRendererToSetUVsOn;

    // Update is called once per frame
    void Update()
    {
        Mesh mesh = meshRendererToSetUVsOn.GetComponent<MeshFilter>().mesh;
        Vector3[] vertices = mesh.vertices;
        Vector2[] uvs = new Vector2[vertices.Length];

        // Loop through each vertex
        for (int i = 0; i < vertices.Length; i++)
        {
            // Convert each vertex position from local to world space
            Vector3 worldPos = meshRendererToSetUVsOn.transform.TransformPoint(vertices[i]);

            // Convert the world position to screen space
            Vector3 screenPos = viewingCamera.WorldToScreenPoint(worldPos);

            // Convert the world position to viewport space (which gives values between 0 and 1)
            Vector3 viewportPos = viewingCamera.WorldToViewportPoint(worldPos);

            // Use the viewport X and Y as UV coordinates (already normalized between 0 and 1)
            uvs[i] = new Vector2(viewportPos.x, viewportPos.y);
        }

        // Assign the new UVs to the mesh
        mesh.uv = uvs;
        // Update the mesh to apply the changes
        meshRendererToSetUVsOn.GetComponent<MeshFilter>().mesh = mesh;
    }
}
```