描边效果，使用Shader就可以轻松实现

这里先给出**效果图**（PS：这是一个3D球）



但是我并不熟悉shader这一块

这里直接给出**shader的代码**

**[cpp]** [view plaincopy](http://blog.csdn.net/xv_ly15/article/details/9038685)

1. Shader "Outlined/Silhouetted Diffuse" {
2. Properties {
3. \_Color ("Main Color", Color) = (1,1,1,1)
4. \_OutlineColor ("Outline Color", Color) = (0,0,0,1)  //改变这个能改变轮廓边的颜色
5. \_Outline ("Outline width", Range (0.0, 0.03)) = 0.008   //改变这个能改变轮廓边的粗细
6. \_MainTex ("Base (RGB)", 2D) = "white" { }
7. }
9. CGINCLUDE
10. #include "UnityCG.cginc"
12. **struct** appdata {
13. float4 vertex : POSITION;
14. float3 normal : NORMAL;
15. };
17. **struct** v2f {
18. float4 pos : POSITION;
19. float4 color : COLOR;
20. };
22. uniform **float** \_Outline;
23. uniform float4 \_OutlineColor;
25. v2f vert(appdata v) {
26. // just make a copy of incoming vertex data but scaled according to normal direction
27. v2f o;
28. o.pos = mul(UNITY\_MATRIX\_MVP, v.vertex);
30. float3 norm   = mul ((float3x3)UNITY\_MATRIX\_IT\_MV, v.normal);
31. float2 offset = TransformViewToProjection(norm.xy);
33. o.pos.xy += offset \* o.pos.z \* \_Outline;
34. o.color = \_OutlineColor;
35. **return** o;
36. }
37. ENDCG
39. SubShader {
40. Tags { "Queue" = "Transparent" }
42. // note that a vertex shader is specified here but its using the one above
43. Pass {
44. Name "OUTLINE"
45. Tags { "LightMode" = "Always" }
46. Cull Off
47. ZWrite Off
48. ZTest Always
49. ColorMask RGB // alpha not used
51. // you can choose what kind of blending mode you want for the outline
52. Blend SrcAlpha OneMinusSrcAlpha // Normal
53. //Blend One One // Additive
54. //Blend One OneMinusDstColor // Soft Additive
55. //Blend DstColor Zero // Multiplicative
56. //Blend DstColor SrcColor // 2x Multiplicative
58. CGPROGRAM
59. #pragma vertex vert
60. #pragma fragment frag
62. half4 frag(v2f i) :COLOR {
63. **return** i.color;
64. }
65. ENDCG
66. }
68. Pass {
69. Name "BASE"
70. ZWrite On
71. ZTest LEqual
72. Blend SrcAlpha OneMinusSrcAlpha
73. Material {
74. Diffuse [\_Color]
75. Ambient [\_Color]
76. }
77. Lighting On
78. SetTexture [\_MainTex] {
79. ConstantColor [\_Color]
80. Combine texture \* constant
81. }
82. SetTexture [\_MainTex] {
83. Combine previous \* primary DOUBLE
84. }
85. }
86. }
88. SubShader {
89. Tags { "Queue" = "Transparent" }
91. Pass {
92. Name "OUTLINE"
93. Tags { "LightMode" = "Always" }
94. Cull Front
95. ZWrite Off
96. ZTest Always
97. ColorMask RGB
99. // you can choose what kind of blending mode you want for the outline
100. Blend SrcAlpha OneMinusSrcAlpha // Normal
101. //Blend One One // Additive
102. //Blend One OneMinusDstColor // Soft Additive
103. //Blend DstColor Zero // Multiplicative
104. //Blend DstColor SrcColor // 2x Multiplicative
106. CGPROGRAM
107. #pragma vertex vert
108. #pragma exclude\_renderers gles xbox360 ps3
109. ENDCG
110. SetTexture [\_MainTex] { combine primary }
111. }
113. Pass {
114. Name "BASE"
115. ZWrite On
116. ZTest LEqual
117. Blend SrcAlpha OneMinusSrcAlpha
118. Material {
119. Diffuse [\_Color]
120. Ambient [\_Color]
121. }
122. Lighting On
123. SetTexture [\_MainTex] {
124. ConstantColor [\_Color]
125. Combine texture \* constant
126. }
127. SetTexture [\_MainTex] {
128. Combine previous \* primary DOUBLE
129. }
130. }
131. }
133. Fallback "Diffuse"
134. }

http://blog.csdn.net/xv\_ly15/article/details/9038685