

リスト 6.11	parallax.frag
<pre> varying vec3 P; //位置ベクトル varying vec3 N; //法線ベクトル varying vec3 T; //接線ベクトル varying vec3 B; //従法線ベクトル varying vec3 view; //接空間視線ベクトル uniform sampler2D normalMap; //法線マップ uniform float fShift; //視線ベクトルシフト量調整ファクタ void main(void) { view = normalize(view); //視線ベクトル正規化 //視線ベクトルによるシフト量 vec2 shift = view.xy * texture2D(normalMap, gl_TexCoord[0]).b; //修正されたテクスチャ座標 vec2 texCoord = gl_TexCoord[0].st + shift * fShift; //高度勾配を取得 vec3 grad = (texture2D(normalMap, texCoord) - 0.5) * 2.0; //修正された法線ベクトル N = normalize(N - grad.x * T - grad.y * B); //以下は通常のライティング vec3 L = normalize(gl_LightSource[0].position.xyz - P); vec4 ambient = gl_FrontLightProduct[0].ambient; float dotNL = dot(N, L); vec4 diffuse = gl_FrontLightProduct[0].diffuse * max(0.0, dotNL); vec3 V = normalize(-P); vec3 H = normalize(L + V); float powNH = pow(max(dot(N, H), 0.0), gl_FrontMaterial.shininess); if(dotNL <= 0.0) powNH = 0.0; vec4 specular = gl_FrontLightProduct[0].specular * powNH; gl_FragColor = ambient + diffuse + specular; } </pre>	