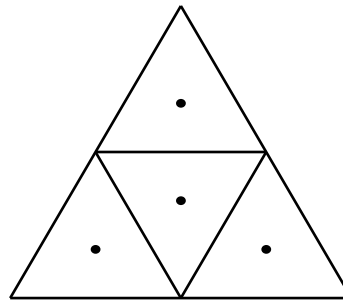
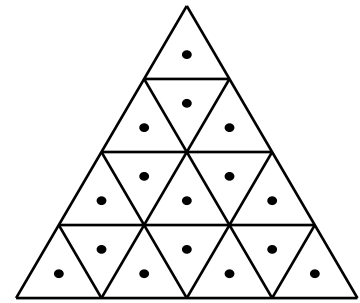


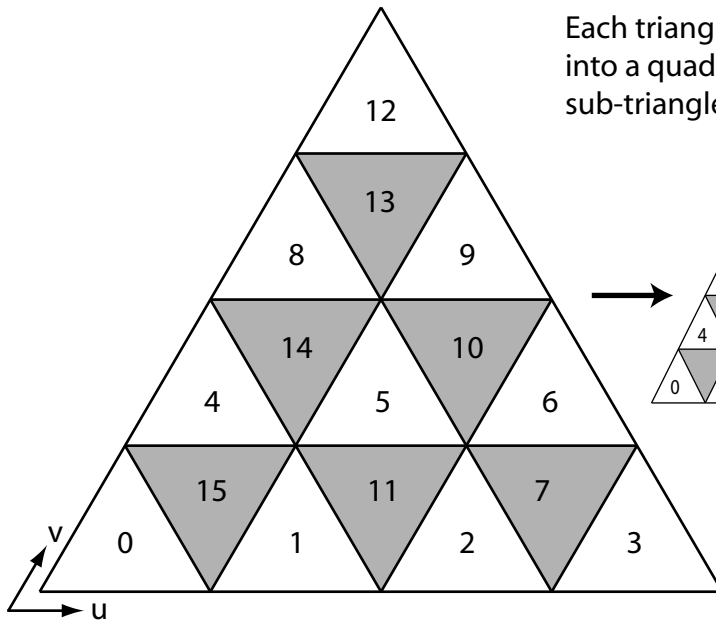
Level 0 = 1 texel



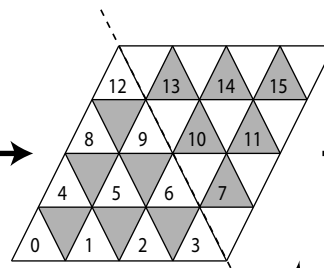
Level 1 = 4 texels



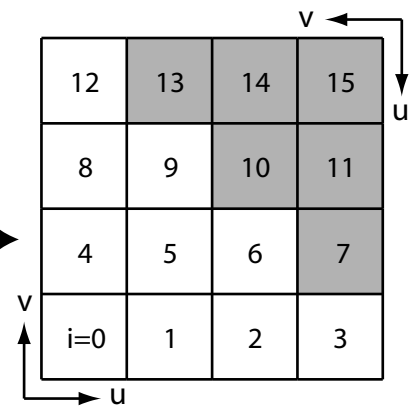
Level 2 = 16 texels  
Level  $n = 4^n$  texels  
( $res = 2^n$  texels)



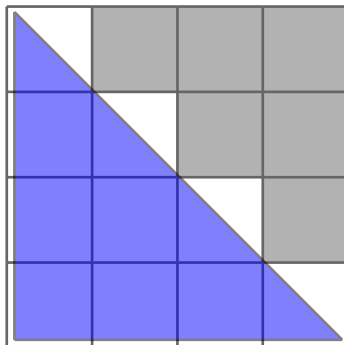
Each triangle texture is packed into a quad texture as two sub-triangles:



flip odd texels

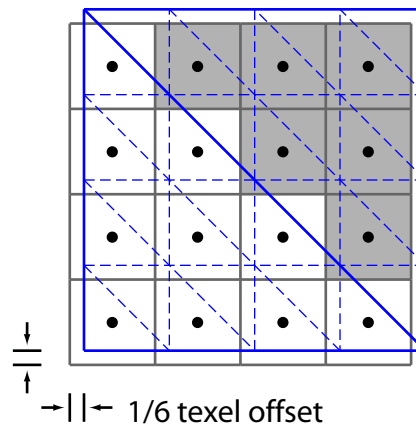


Indexing (computing  $i$  from  $u, v$ ):  
 $ut = u * res; vt = v * res;$   
 $ui = \text{floor}(ut); vi = \text{floor}(vt);$   
 $uf = ut - ui; vf = vt - vi;$   
 if  $uf + vf \leq 1: i = ui + vi * res$   
 else:  $i = (res^2 - 1) - (vi + ui * res).$



For GL display, the triangle can be rendered directly from the lower half-texture. A small epsilon should be used to keep the triangle inside the texture.

Alternately, a shader can point-sample the full texture using the above indexing method.



For paint projection (i.e. rasterizing triangles into the quad texture), two triangle projections are needed to cover the quad. A 1/6 texel offset is also required to align the triangle sample points with the quad texel centers.