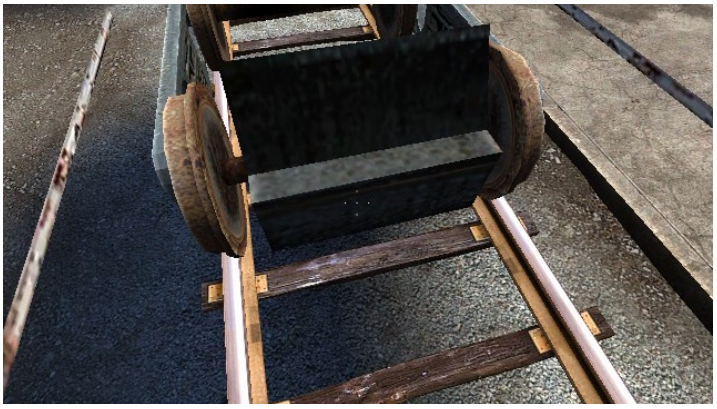
What you need:
- take a look on several maps and look for prop_statics that are dark even they are standin in a bright zone or bright models in dark areas (maybe some trees cutting into a building)



1. Why paying attention to vertex-lighting

All your models you export to the Source engine have to use Vertexlighting on their materials. There is no way to lightmap your models using the Vrad compiler.

I made some screenshots of de_train where you may see results of vertexlighting.

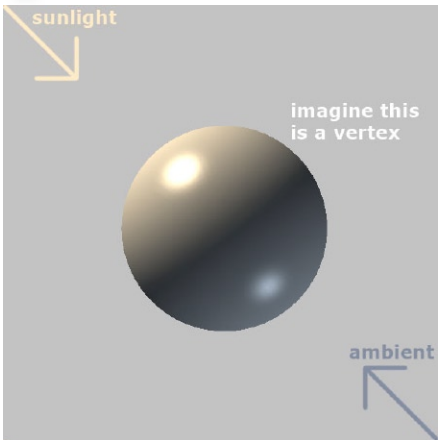
In the first picture you see train rails. they are both next to each other. the one is in the outside under influence of sunlight the other one is under a roof where artificial light is emitted. So you can see: there are two models and they don't look good due to that hard edge of lighting between them.

The second image shows you how vertexlighting is applied to models. Somehow upper vertices are bright and lower ones dark.

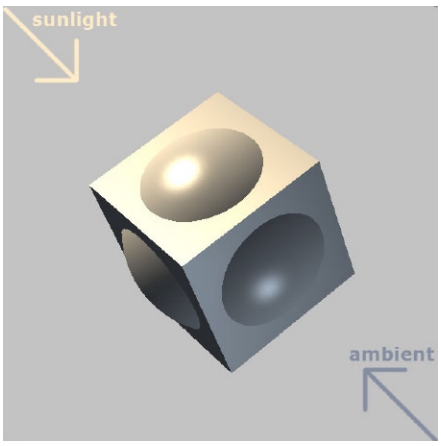
The third screenshot was made to show you that vertexlighting doesn't cast „selfshadows" on models.

You can see the same fact in the fourth picture too.

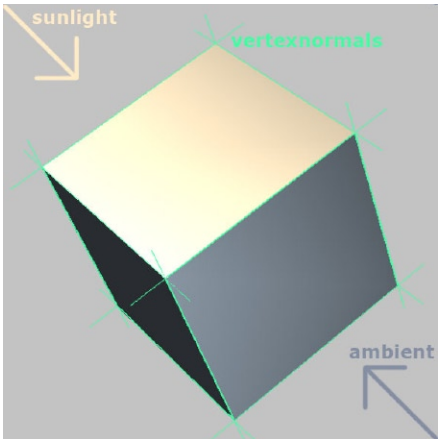
2. Lighting a single and multiple vertices



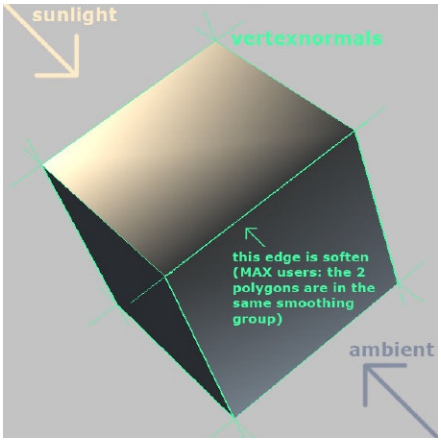
Too understand vertexlighting i created a simple scene in Maya with a sphere and 2 lights.
The sphere stands for a vertex or the info_lighting point entity you can place in the hammer editor.
The sunlight and ambient light cast different colors on the sphere. In the middle you should notice a dark stripe on the sphere because there is no light pointing to the sphere in the same direction the camera does.



I stuck a cube with harden edges in my sphere. As you can see it is lighted like the parts of the sphere you can still see. That's a result of the vertexnormals you have on every polygon mesh.



Here you see those vertexnormals. Each polygon has them on its vertices. The result: a vertex can have n>=1 vertexnormals (math N means integers bigger than 0; N = [1, 2, 3,...] while n is an element of N).
Take a look on the upper polygon. Its really bright yellow. All vertexnormals are orientied to the upside yellow sunlight. So they say „ok polygon has to be yellow at my side. -> yellow gradient on this polygon.
The same is valid for all other polygons and vertices. Imagine that each vertex of the cube is the sphere I showed in the first image. Place the vertexnormal on the spehere and place the lighting the normal points to on polygons vertex.
Talkin unprofessional: the sphere is a projection to all vertices telling the polygons vertexnormals how to be lighted.



Interesting: if you soften edges (or work with smoothing groups) your vertexnormals get merged to an average orientation. So they are no more like instances of the facenormals. Due to that the edge I softened has a vertexnormal pointing away from all lights in my scene. result: we got a yellow lighting at the top polygon that fades to a black lighting (no lighting) and fades to a blue lighting at the bottom (same as the „soft"/„smooth" sphere did).

3. Resumé

Knowing these basics you should understand why prop statics are dangerous to use in outdoor scenes.

Example:
Take a look on de_train. There train railings at the bottom have an extrem bright lighting that hardly cuts to a dark one.
That is the result of the fact that there are two models using 2 different lighting reference points (info_lighting) the one is in the sun the other one is in the dark area. All vertices of the prop_ static are referenced to this lighting point.

So take care of how you work with statics and props. There might be ugly bright or dark models in versa lighting situations because this info_lighting is in a different room or position your vertices are.