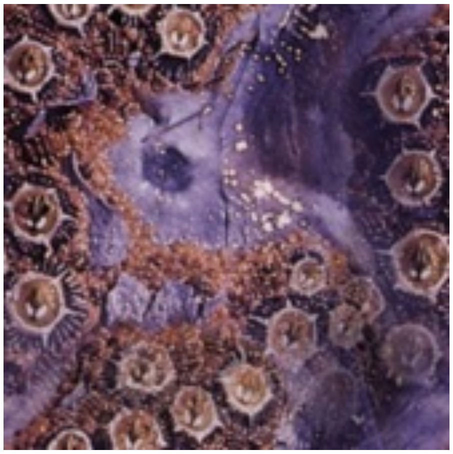
Understanding Gamma

J.P. Lewis

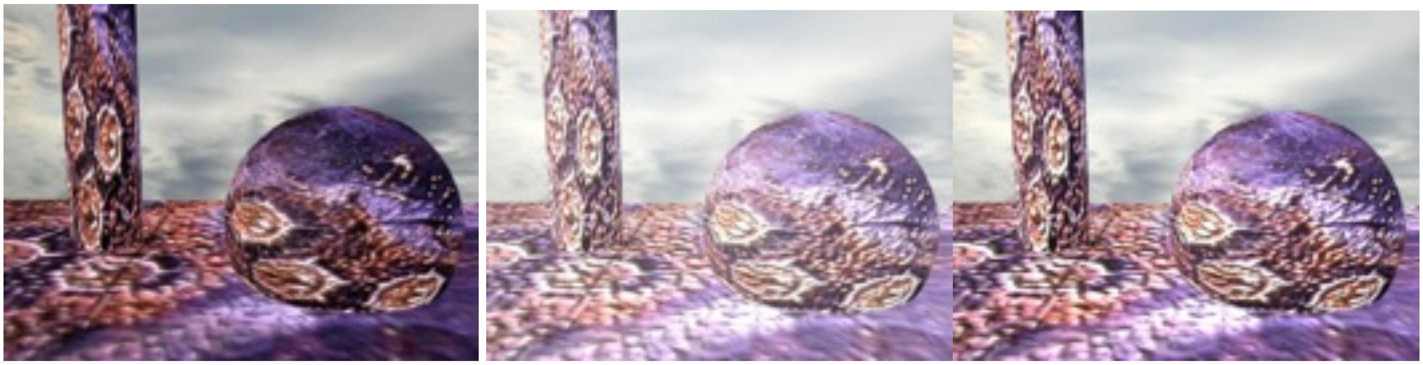
September 2009

Why it matters

If gamma is not considered the contrast in the final scene will be distorted, and the image can end up “blown out” with crushed dark areas, or washed out. Here’s a simple example, an original texture:

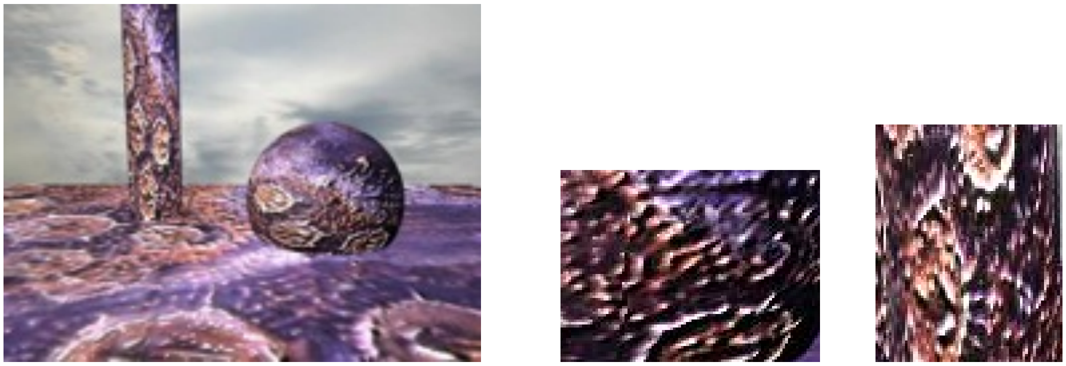


Here is the “correct” rendering on the left, and two renderings that do not consider gamma on the right:



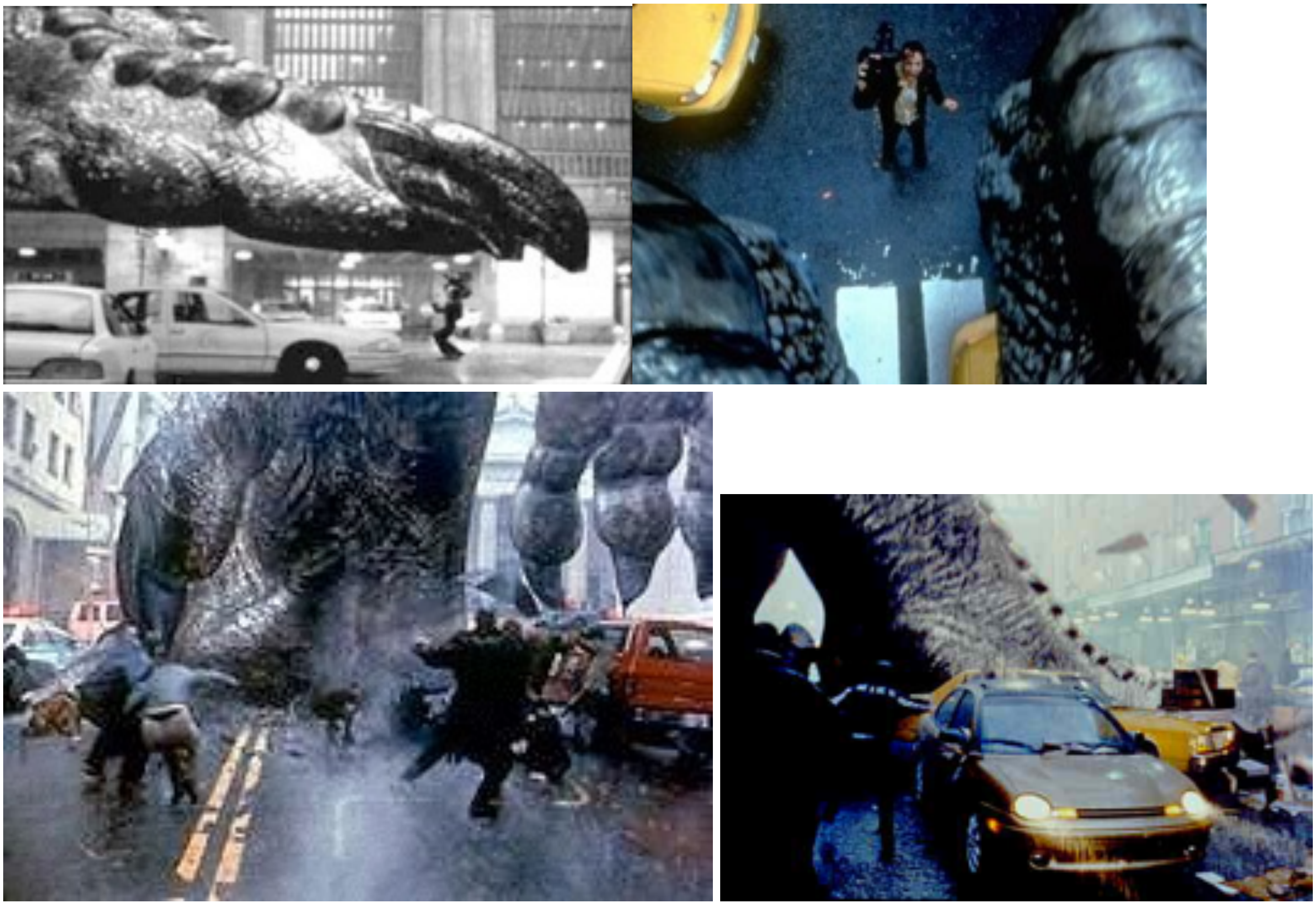
The one on the right looks OK at first, but look closer: the texture no longer looks like the original, it is washed out, has too much contrast, and we start to loose the sense of shape on the cylinder as a result.

To fix the washed out problem, look what happens if we reduce the ambient to zero (new version like the picture on the right), with some details of the resulting texture rendering:



The CG has a very high contrast look that says “the artist didn’t pay attention to gamma!|”:

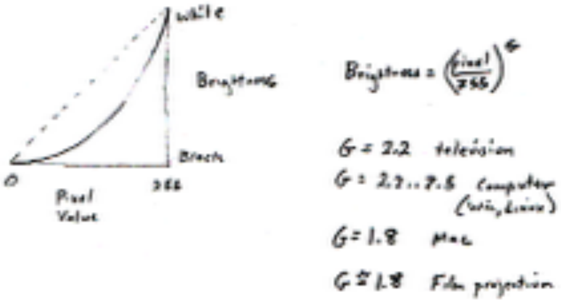
The first 1/3 of the movie Godzilla (1998) had gamma problems. Note the excessive contrast in these images:

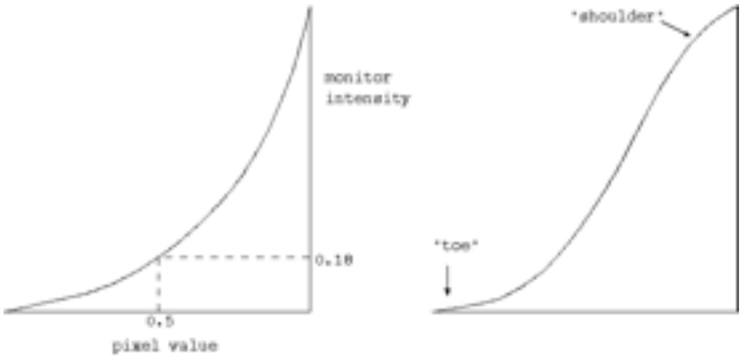


Here’s an example from later in the film, when the gamma treatment was corrected:

What’s Happening:

Monitors distort the brightness in an image: **they reduce the dark areas more than the ligh**t. Film also distorts brightness, but in a different way:





Left: shape of a gamma curve for a monitor. Right, film has a roughly similar shaped curve, with differences at top and bottom, exaggerated here for illustration (the classic toe and shoulder plot is the log-log plot of density versus log exposure).

.

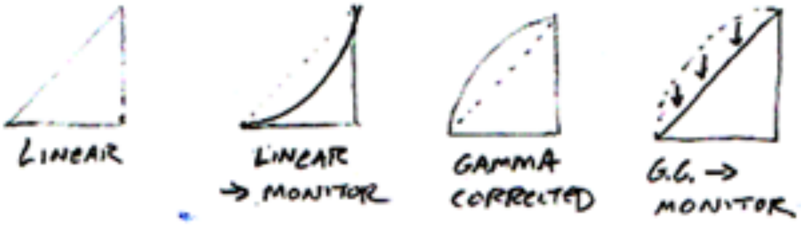
Definitions:

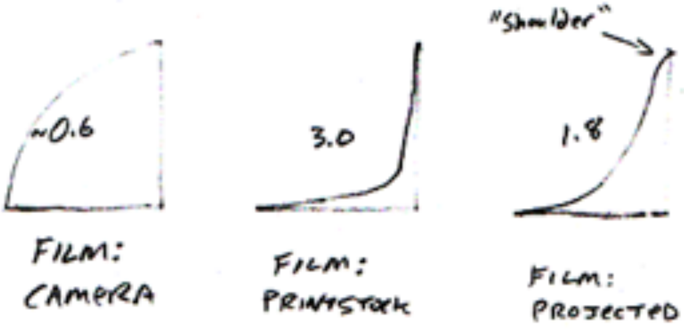
- Gamma is the exponent of the power curve, out=pow(in/255,G) "Colorspace" also means RGB vs HSB, etc.;

- Gamma is just one aspect. Film people say "colorspace" to mean "gammaspace".

Common Colorspaces:

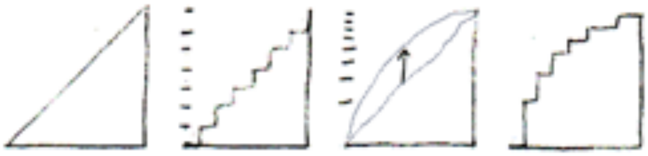
- Linear
- Log (approximately like Monitor, but not exactly)
- Monitor (or Video): out = pow(in,1/gamma), gamma = 1.8..2.5





8 bits, 10 bits, ...Gamma Conversion

- 8 bits/component (R,G,B) are enough to display an image
- 8 bits are not enough to convert between colorspaces

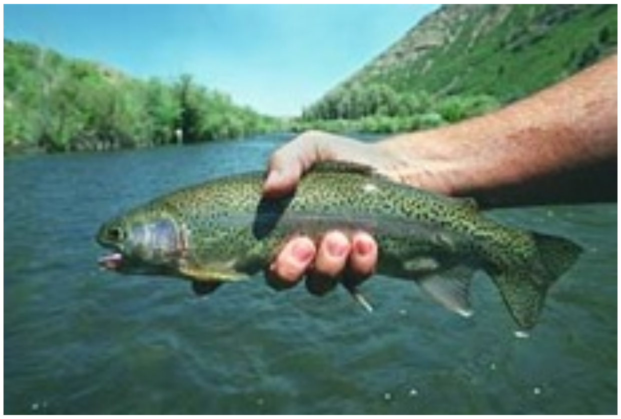


Solution: use more than 8 bits

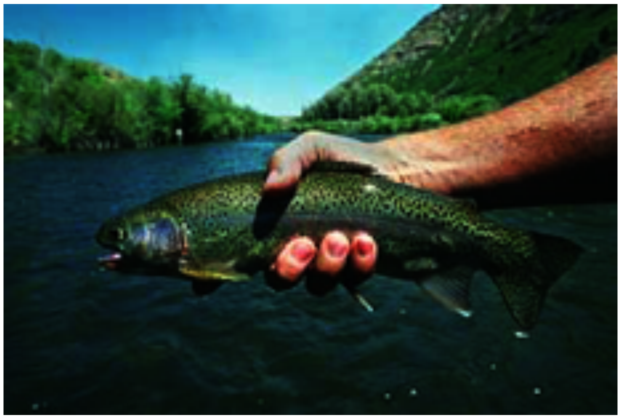
- Cineon format for film: 10 bits, log
- Convert to linear (16bits or float) in compositing program, Convert back to 10bit log for filming

Recognizing gamma

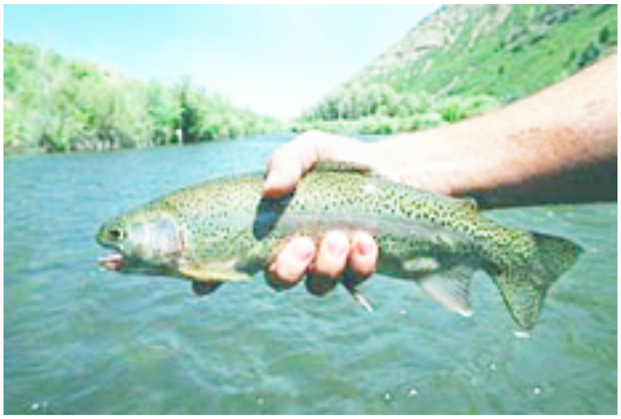
Original picture with monitor correction (as it would come from a digital camera). The data has already been distorted so it looks correct on a monitor: the dark areas have been brightened (“monitor correction”).



A linear picture (no monitor correction):



Result of one-too-many monitor correction or log-like steps: ("washed out").



Recommendations:

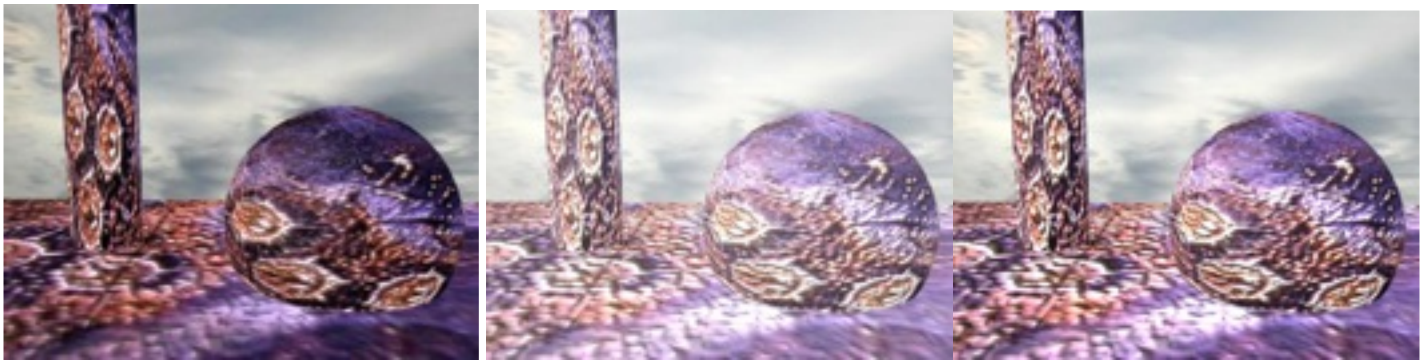
- Paint texture in monitor space, but convert to linear before rendering

- Output of the renderer (mental ray, Maya) is linear by default. Correct to monitor space in order to adjust lighting.

- Composite with linear CG over linear background! Set the shake viewer to compensate for the monitor.

We've already shown some examples of ignoring this advice. The early *Godzilla* shots composited linear CG over log (cineon) backgrounds: the CG had much more contrast than the background.

The example at the beginning showed the result of not converting textures to linear before rendering:



- The image on the left has the texture converted to linear, and the final render converted to monitor space.

- The image in the middle uses the original (monitor space) texture, and converts the final render to monitor space.

- The image on the right has no corrections: the texture is in monitor space, and the final render is a linear image (viewed on a monitor).

In the middle image the texture has been compensated for the monitor twice: once ("by eye") when it was originally painted (or obtained from a digital camera which also outputs monitor space images), and a second time when the final render was corrected. The texture is too washed out as a result.

In the image on the right, the texture has only been corrected once, when it was first painted. However, the CG shading has not been corrected for the monitor, so it has too much contrast. This is particularly visible in the bump mapping areas, which are practically black and white with no intermediate colors. This also distorts the colors in the texture, it no longer looks like the original.

Compositing: The alpha should be applied in the color space in which it was created

- 3D: alpha is calculated in linear, do composite in linear colorspace
- Video (Ultimatte-like) device: matte is calculated in video, apply it in video.

The basic 'over' operation in compositing is

a*F + (1-a)*B

where F,B are the foreground and background, and a is the alpha channel value rescaled as a fraction between 0 and 1. The image that comes out of a 3D renderer is "premultiplied", meaning that the image contains a*F rather than F. The use of premultiplied images was introduced in the compositing algebra paper by Duff and Porter in the Siggraph84 proceedings.

Consider trying to composite in any other color space. Try to convert everything to Log (for example), do the composite, and then convert back.

- p = a*F + (1-a)*B;   // the linear result
- Exp(a*Log(F) + (1-a)*Log(B)) = p?   usually not
- Also try converting the alpha: z = Log(a)
- Exp(z*Log(F) + (1-z)*Log(B)) != p also

The results will generally not be the same as what was obtained with the original alpha... and the original alpha is "correct" in the sense that it is the best that the 3D renderer or other software can do. The "matte lines" artifact may be visible, particularly if the alpha transition is wide, as with a translucent or motion-blurred object.

This is true for any other non linear mapping and its inverse.

Other operations

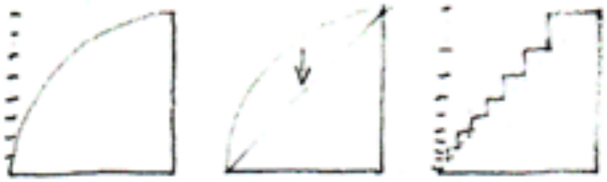
Some other operations should be done in linear gamma:

- Lighting: in linear gamma, increasing the CG lights by 2x is one "stop" in photographic terms. In monitor gamma this is not true --the bright areas increase by more than one stop, the dark areas by less.
- Antialiasing: antialiasing is calculated in linear and looks wrong in other gamma spaces. Example is a starfield - the stars "flicker" or "throb" during a slow pan if they are shown in an incorrect gamma.
- Similarly, blurs give different results according to the colorspace of the data.

Texture: if only 8bits (e.g. photoshop), paint in Monitor space

"Paint in monitor space" means paint on an uncorrected monitor, with a gamma of 2.2..2.5. The artist will make a picture that looks correct on the monitor, so they are unconsciously correcting for the monitor gamma.

Why: the human eye has a log-like response -- it is more responsive to the dark areas, and "crushes" the light areas. Thus, if you must quantize, you want the quantization to happen in the light areas where it won't be noticed.



Perception

Should we take the log-like effect of our perception into account? No (though exception is in considering texture quantization from an 8bit paint program). The reasons:

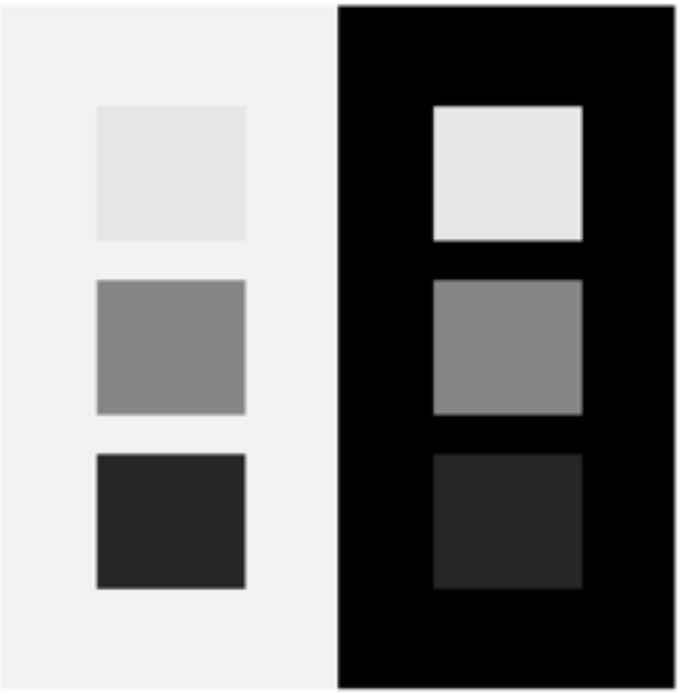
- It further confuses an already confusing issue, and it isn't necessary.
- Reality doesn't take our perception into acount, it shows us, or a camera, "linear light". The starting point for the behaviour of the virtual CG camera is as a simulation of the real camera, so it should also capture linear light, regardless of how our brain sees it. (We can depart from this engineering / "real point" of view for artist reasons, but we shouldn't depart from it simply by ignorance or mistakes!).

Why does projected film have a gamma of 1.7-1.8?

Because of the "dark surround" perceptual effect, an optical illusion that reduces apparent contrast when

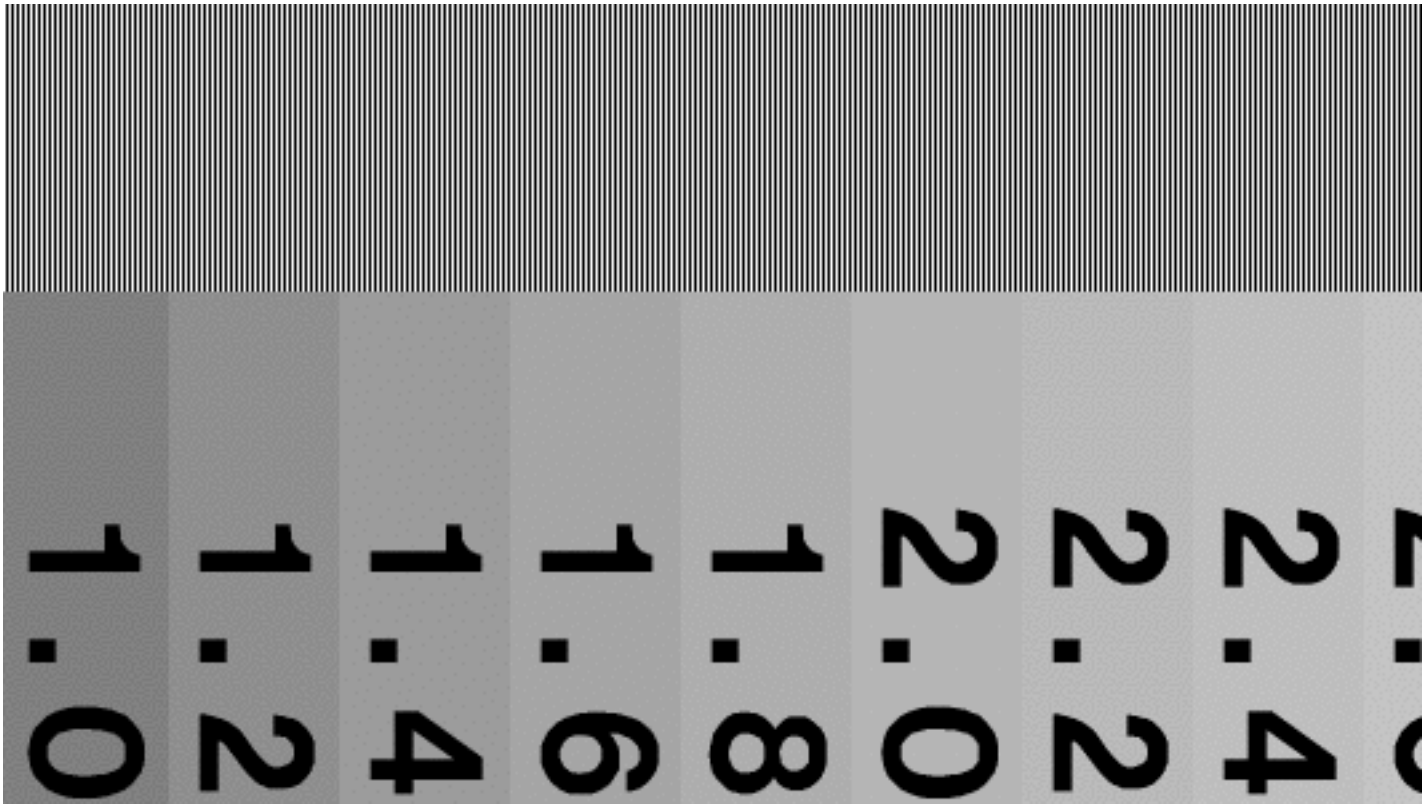
the peripherial regions are dark (as in a theater).

In CG film we can mosly ignore this: if we can make a monitor-compensated image look good in a not-too-dark room,then we just convert the frames to Cineon (Log) and the film-out process will do the rest.



Dark surround effect
from C. Poynton - gamma Faq

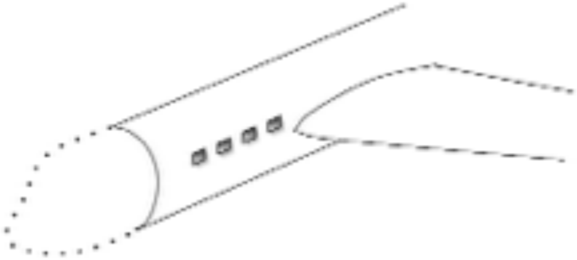
Identifying Gamma: halftoning charts

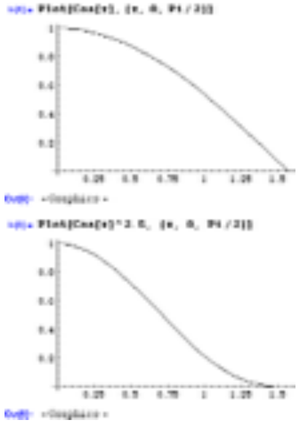


Color balance example

Imagine an object painted with a constant color (like the plane), but which has light and dark due to illumination. Because it is nonlinear, gamma changes the balance of r/g/b differently in the lights vs darks. The classic symptom is, the CG matches the image colour in the light regions but not in the dark, or vice versa.

Special Effects Example: Add a new nose special effects to an airplane:



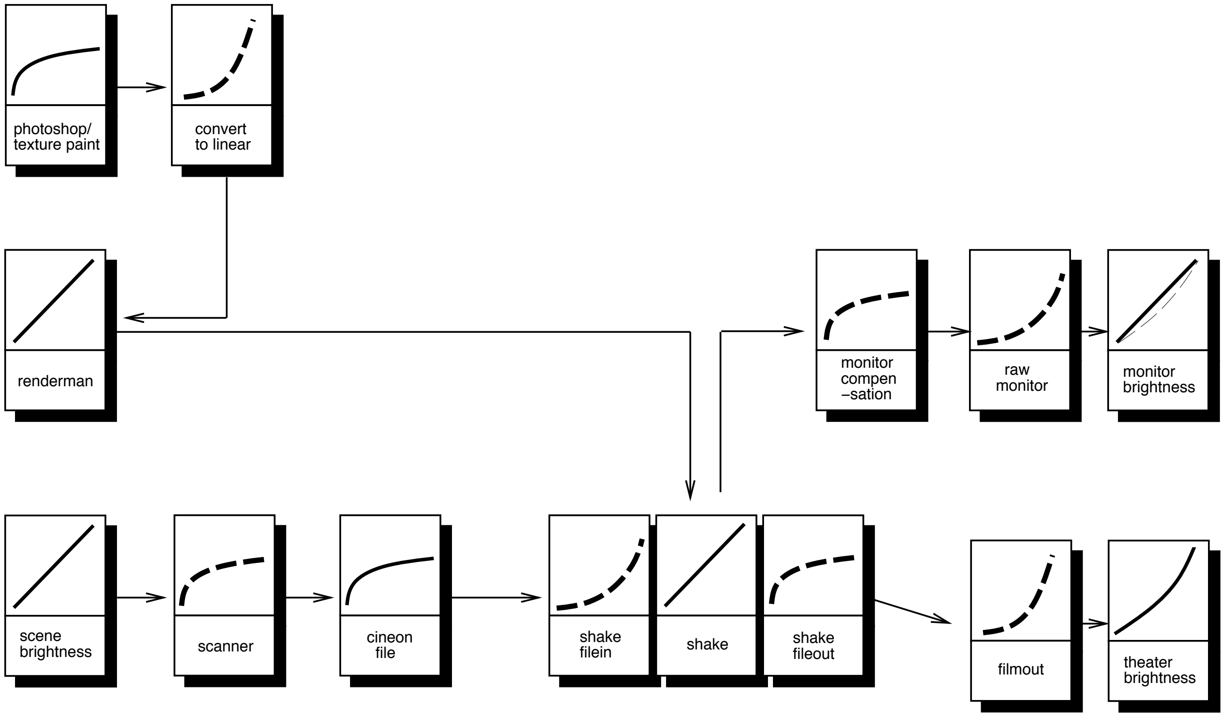


```
Plot[Cos[x], {x, 0, Pi / 2}]
-Graphics-
Plot[Cos[x]^2.0, {x, 0, Pi / 2}]
-Graphics-
```

Even if the renderer calculates the lighting on the new nose exactly, if the gamma is wrong it will not match. The lighting on the airplane body cylinder will fall off according to some curve (like the cosine at the top if it is purely diffuse). A gamma changes the shape of this brightness curve.

Example FX (not animation) Gamma Pipeline
(Rated R: Horror and Violence)

Suggested Deep–Bit Gamma Pipeline
J.P.Lewis



photoshop/ texture paint

convert to linear

renderman

monitor compen–sation

raw monitor

monitor brightness

scene brightness

scanner

cineon file

shake filein

shake

shake fileout

filmout

theater brightness

NOTE: ‘brightness’ in this chart means the
amount of physical light, not perceptual brightness.