Method for Increasing Apparent Amplitude Resolution, and Correcting Luminance Nonuniformity in Projected Displays

Robert Ulichney
Hewlett-Packard Co., Cambridge, MA (USA)
u@hp.com

Abstract

Projectors suffer from nonuniformity across the display, as well as spurious contours due to insufficient levels. This work offers an efficient and accurate solution to both problems. A camera-based calibration scheme both measures the spatial luminance nonuniformity and detects the projected levels that are unique. Using this data, an attenuation array is built using reverse-3D-table lookup. The method provides an accurate correction without needing to know or measure the camera-to-projector response (gamma). Knowing the unique levels that the projector can display, the system can work with the actual reduced number of levels. Spurious contouring or banding distortion is corrected by a mean-preserving multilevel spatial dither system.

1. Introduction

The display from projectors suffer two shortcomings:

(1) The luminance is not uniform across the display, and

(2) Insufficient number of unique displayable levels results in false contours or banding.

The first problem is particularly pronounced when a set of projector displays are concatenated in a wall display, as in the system described in [1]. Figure 1 plots the luminance of a typical projected display in foot-lamberts as measured with a spot photometer. Note that there is about a two-to-one divergence between the brightest and dimmest part of the display.

The second problem is seen in slowly varying regions of an image or video –most notably in highlights or shadows—where spurious contours can be seen due to insufficient levels.

Very little work has been done in the area of correcting projector luminance. One recent study [2] measures a single sample screen of the projector assuming a calibrated camera for the purpose of correcting non-uniform luminance, but does not address the problem of false contours. With one measurement, the response of the projector across all image levels cannot be known or properly accounted for. The problem of color calibration is addressed in [3], but again, neither the response of the camera or the projector is taken into account.

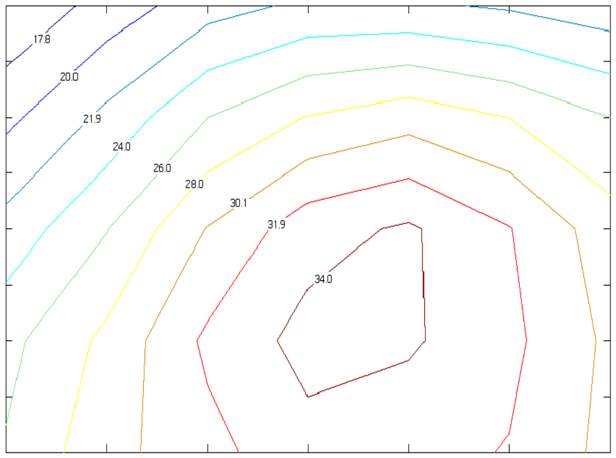

Figure 1. Typical luminance in foot-lamberts of a projected "white" image.

Temporal dithering is commonly used to achieve the perception of multiple luminance levels in projectors that used micro-mirrors such as Texas Instruments DLP. Temporal dithering, which achieves the illusion of varying brightness levels by changing the number of flashes per second from each pixel, can suffer from visible flicker when the number of levels increases.

The system described in this paper addresses both shortcomings.

The following symbols will be used:

N_g	Number of **G**iven, or stated, projector Levels. Typically 256.
N_o	Number of **O**utput levels; levels that the projector is actually capable of producing. This is also the number of output levels from the dither system.
N_r	Number of **R**aw input levels; this number can be greater than actually used. This is the number of input levels to the final system, and number of levels that the system can render.

N_i	Number of **I**nput Levels to the Dither System.
N_t	Number of Dither **T**emplate levels.
n_g	An input level to the projector. $n_g = \{0, 1, \ldots, (N_g-1)\}$.
n_o	A unique projector input, or an output from the dither system. $n_o = \{0, 1, \ldots, (N_o-1)\}$.
n_r	A raw input level to final system. $n_r = \{0, 1, \ldots, (N_t-1)\}$.
n_i	An input level to the dither system. $n_i = \{0, 1, \ldots, (N_i-1)\}$.
n_t	A dither template value. $n_t = \{0, 1, \ldots, (N_t-1)\}$.
$c_g(x, y, n_g)$	Camera Capture Array. The image planes captured by the camera of a full screen displayed by a projector with constant input, n_g, at all pixel locations.
$c_o(x, y, n_o)$	Pruned Camera Capture Array.
$p(x, y, n_o)$	Projector Correction LUT. A look-up table that modifies an input i at a particular location (x, y) to a value that will appear as level i in the projected display.
$[X_c, Y_c]$	Number of x and y camera pixels. Typically values would be [640, 480].
(x_c, y_c)	Spatial address in terms of camera pixels.
$[X_s, Y_s]$	Number of x and y subsampled pixels. Typically values would be [16, 12].
(x_s, y_s)	Spatial address in terms of subsampled pixels.
$[X_p, Y_p]$	Number of x and y projected pixels. Typically values would be [1280, 768].
(x_p, y_p)	Spatial address in terms of projected pixels.
$[X_d, Y_d]$	Number of x and y elements in the dither array. Typically values would be [32, 32].
(x_d, y_d)	Spatial address into the dither array; the least significant bits of (x_p, y_p).
$a_c(x_c, y_c)$	Attenuation array for the camera. Values range from 0.0 to 1.0.
$a_p(x_p, y_p)$	Attenuation array for the projector. Values range from 0.0 to 1.0
$g(n_r)$	Front End LUT or Gain LUT.
$B(n_o)$	Back End LUT.
$T(x_d, y_d)$	Dither Template. Values are ordered from 0 to (N_t-1).
$d(x_d, y_d)$	Normalized Dither Array.

Our solution to the projector shortcomings in part C starts by using an ordinary video camera to measure the output from each input to the projector $n_g=\{0,1, \ldots, (N_g-1)\}$. We find the N_o levels that are unique. Through dithering we will be able to display all N_g levels, and in fact even more levels; so we define the number of desired display levels, N_r, that may be greater than N_g. We employ a fast mean-preserving run time dither system to comprising an efficient solution to both spurious contours and nonuniform luminance.

2. Flat-Fielding the Camera

Of course, the camera that we with to use has a nonuniform spatial response due to vignetting, the cosine-forth rule and other nonlinearities. In addition to this the response of the camera is not radiometricly linear. Along with spatial non-uniformity, the projector also suffers from a response that is not radiometricly linear.

The first step is to flatten the field of the camera. The problem defers to finding a flat field. The process of "flat fielding" a camera is well known in astronomy where a flat field –an averaged clear ski– is readily available at the focal length and aperture needed. For our purpose, an appropriate flat field is not obvious. One solution is to use a large white board in homogenous ambient illumination. We used a spot photometer as shown in Figure 2 to take several spot readings of the board at the location of the camera and interactively tilt the board in two dimensions until the readings are uniform. The video camera then captures this flat field at a focal length and aperture in the range of what would be used when viewing the projected display. Several frames are averaged to reduce sampling noise. The dimmest pixel in the averaged frame is found. A camera attenuation array, $a_c(x_c, y_c)$, is generated by taking the ratio of each pixel and the dimmest pixel, resulting is values at all locations between 0 and 1.

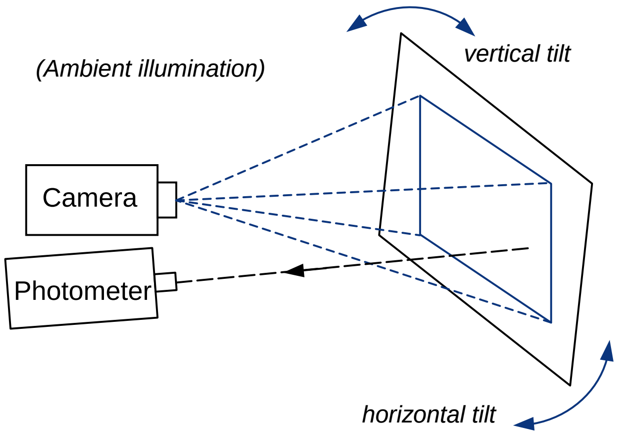

Figure 2. Flat fielding the camera

As camera capture sensitivity is quite stable, this attenuation array is considered permanent and need be measured only once. All subsequent measurements taken with this camera will now be modified by a point-wise multiply with $a_c(x_c, y_c)$. With this correction any 2 points seen by the camera with the same luminance will be captured with the same value.

3. Building the Camera Capture Array

Figure 3 illustrates the arrangement for capturing output from the projector. The projector sequentially displays each level for which it can display (0 through N_g-1) and the camera captures each, one at a time. The setting of the camera aperture must be appropriate to avoid clipping at both the bright end and dark end of the range of projected levels.

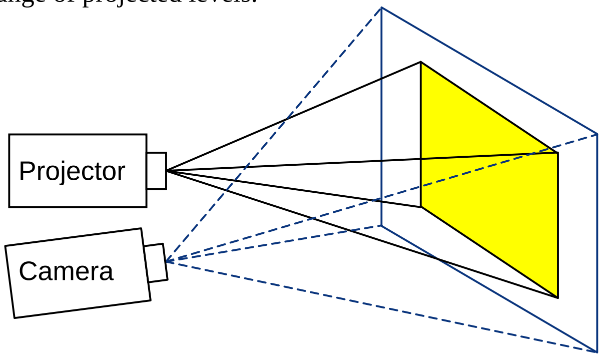

Figure 3. Generating the camera capture array

While the optical axis of the projector may be perpendicular to the screen resulting in a rectangular displayed image, the optical axis of the camera may not be perpendicular to the screen. If this is the case the camera will “see” a quadrilateral rather than a rectangle. We need to transform this quadrilateral into a rectangle. This transformation is the homography between the projector and the camera. The homography between two surfaces can be determined by finding four corresponding points in the two coordinate systems. We can easily do that in this case by finding the corners of the quadrilateral in the image grabbed by the camera. The details of this are found in [4]. Once we get the homograhy, we apply its inverse to any subsequent camera grabbed image to get the rectangular grabbed image in the projector frame of reference.

As in the camera flat-fielding stage, several frames are captured at each level and averaged to reduce video noise. These N_g planes are normalized with the camera attenuation array, and stacked to form a 3D camera capture array, $c_g(x_c, y_c, n_g)$, as shown in Figure 4. The horizontal and vertical samples (x_c, y_c) are in the spatial address of the camera, typically in the range $x_c = \{0, 1, \ldots, 639\}$, $y_c = \{0, 1, \ldots, 479\}$.

It is important to note that the “screen” captured can include nonuniformities beyond those inherent in the projector. This can include varying albedo due to differences in reflecting material, and even geometry. For example, the projector could display on an inside corner between a white wall and a gray cabinet. Another special case of an irregular screen is a flat surface that is not perpendicular to the optical axis of the projector. In such an oblique projection, the regions closer to the projector will brighter than the same projected regions would be otherwise.

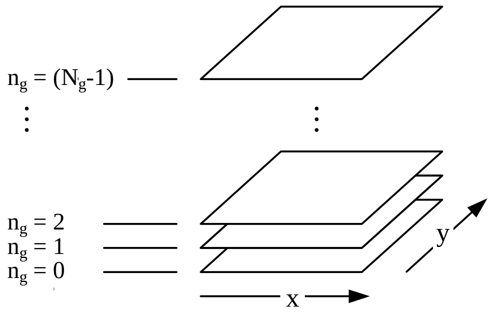

Figure 4. Camera capture array $c_g(x,y,n_g)$

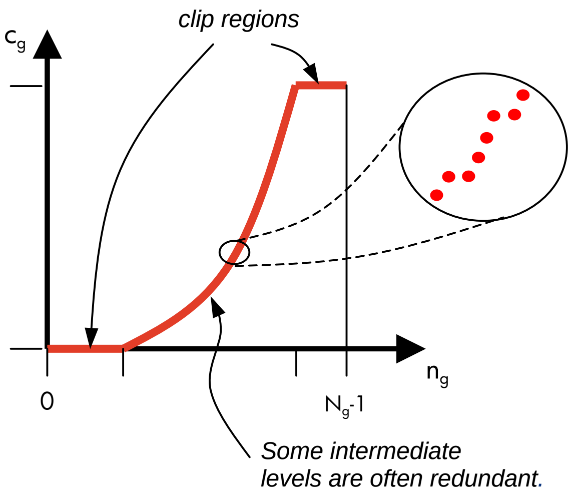

Figure 5. Relationship between input value, n_g, and the average value captured by a camera, c_g, of the projected input value.

3. Pruning the Camera Capture Array

We discovered that projectors often display redundant or non-unique levels that we prune from the camera capture array in the following way. Each plane of $c_g(x_c, y_c, n_g)$ is averaged and plotted as in Figure 5. This reveals two types of redundancy. The first is that the top and bottom of the range are clipped. More surprisingly the second type of redundancy is in the mid-level region where in many cases every third level is a copy of the level next to it. All of these redundant levels are pruned from the camera capture array, reducing the number of planes from N_g to N_o. Example values are $N_g = 256$ and $N_o = 131$. What is left is the pruned camera capture array $c_o(x_c, y_c, n_o)$, as shown in Figure 6. The mapping from $i_o = \{0,1,\ldots(N_o-1)\}$ to $i_g = \{0,1,\ldots,(N_g-1)\}$ is recorded in a look up table called the Back End LUT, $B(n_o)$.

For computational ease, the spatial dimensions of the pruned camera capture array are reduced from the camera grid of X_c by Y_c to smaller subsampled grid of X_s by Y_s (typically of dimension 16 by 12) after some low pass filtering, resulting in a X_s by Y_s by N_o array. The spatial addresses of points on this subsampled grid are represented by values (x_s, y_s). In cases were the projection is known to take place on an irregular or highly varying screen, the subsample size X_s by Y_s should be larger to accommodate the details of the irregularity.

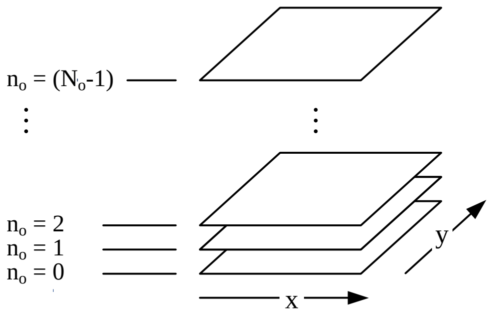

Figure 6. Pruned camera capture array $c_o(x,y,n_o)$

4. The Projector Correction LUT

From the pruned camera capture array $c_o(x_s, y_s, n_o)$, a projector correction LUT $p(x_s, y_s, n_o)$ of the same size is then generated. At each plane, n_o, a value of n_o is assigned to the lowest value, and all other values (x_s, y_s) are found by direct reverse look-up down the columns of the pruned camera capture array $c_o(x_s, y_s, n_o)$, matching the plane level closest to no. In most cases the search will yield a match between 2 levels; in such cases fractional values are interpolated between plane levels; the values of $p(x_s, y_s, n_o)$ are not restricted to integers. Then for any given value n_o, the projector correction LUT will provide an X_s by Y_s array that can be interpolated to the full size of the projector address space X_p by Y_p to project a uniform screen of the desired value. Typically a projector address space is 1024 by 768.

It is important to note that this mapping directly accounts for both the nonlinear luminance response of both the camera and the projector without have to measure the absolute response of either!

5. Projector Attenuation Array and Front End LUT

A straightforward implementation would be to directly use this giant 3D LUT $p(x_p, y_p, n_o)$. With 16 bits per entry this table could require over 200 Mbytes. We have found that almost identical results can be achieved be breaking this LUT into two parts: a 2D attenuation array and a 1D Front-End LUT.

The attenuation array for the projector $a_p(x, y)$ is found by first normalizing each plane of $p(x, y, n_o)$ by dividing each value by the maximum value in each plane. Then all planes are averaged.

$$a'_p(x,y) = \underset{n_o}{\text{avg}}\left[\frac{p(x,y,n_o)}{\underset{(x,y)}{\max}\left[p(x,y,n_o)\right]}\right]$$

Since the location of the maximum value of $a'_p(x,y)$ may not be the same for all no, the array will need to be further normalized to arrive at the final attenuation array

$$a_p(x,y) = \frac{a'_p(x,y)}{\underset{(x,y)}{\max}\left[a'_p(x,y)\right]}$$

The Front-End LUT $g(n_o)$ can be found for each value n_o by first dividing $p(x, y, n_o)$ by $a_p(x, y)$ at each n_o, then averaging over (x ,y):

$$g(n_o) = \underset{(x,y)}{\text{avg}}\left[\frac{p(x,y,n_o)}{a_p(x,y)}\right]$$

Note that the above two steps can be done at the projector resolutions (x_p, y_p), but it is more efficient to perform the operations in subsampled space (x_s, y_s). The attenuation array for the projector would then be interpolated from $a_p(x_s, y_s)$ to $a_p(x_p, y_p)$.

At this point it is important to introduce a new variable, Nr, the “raw” input levels to the system. This value can be equal to the stated or intended number of projector levels, N_g, or it be a value greater than that, made possible by dithering. The first element of our run time system in Figure 7 is a pass through the Front End LUT. Thus, $g(n_o)$ must be scaled through interpolation to accept this wider range of input values, resulting in $g(n_r)$.

6. Mean-preserving Multilevel Dither

This section details a means for spatial dithering with minimum hardware or software, yet guarantees output that preserves the mean of the input. It is described in [5]. It allows dithering from any number of input levels N_i to any number of output levels N_o, provided $N_i \geq N_o$. Note that N_i and N_o are not restricted to be powers of two.

The input image can have integer values between 0 and $(N_i - 1)$, and the output image can have integer values between 0 and $(N_o - 1)$. A deterministic dither array is used; to simplify addressing of this array its dimensions should each be a power of two. A dither template T(x', y'), such as that described in [6], defines the order in which dither values are array are arranged. The elements of the dither Template have integer values between 0 and $(N_t - 1)$, where N_t is the number of template levels, which represent the levels against which image input values are compared to determine their mapping to the output values. The dither template is central to determining the nature of the resulting dither patterns.

Figure 7 shows a dithering system that comprises a dither array, an adder, and a shift register. The system takes an input level n_i at image location (x_p,y_p) and produces output level n_o at the corresponding location in the dithered output image. The dither array is addressed by x_d and y_d, which represent the low-order bits of the image address (x_p,y_p). The selected dither value $d[x_d, y_d]$ is added to the input level to produce the sum that is quantized by simply removing the least significant R bits, as depicted in the "Shift R bits" block.

The trick to achieving mean-preserving dithering is to properly generate the LUT values. The dither array is a normalized version of the dither template specified as follows:

$$d[x_d,y_d] = \text{int}\{\Delta_d(T[x_d,y_d] + ½)\},$$

where int{} is integer truncation, Δ_d, the step size between normalized dither values, is defined as

$$\Delta_d = 2^R/N_t .$$

As detailed in [5] the following variables are defined:

$$R = \text{int}\left\{\log_2\left(\frac{2^b - 1}{N_o - 1}\right)\right\}$$

where in this case we assert that $2^b = 2N_r$ because N_r is a power of 2. Also, the number of input levels is defined as

$$N_i = (N_o - 1)2^R + 1.$$

For this simple system to be mean preserving we must impart an additional gain on the raw input n_r equal to

$$(N_i - 1) / (N_r - 1).$$

In our run time system in Figure 7 this gain can be rolled into the Front End LUT by simply modifying each element of $g(n_r)$ by multiplying it by $(N_i - 1) / (N_r - 1)$ before loading into the run time system.

7. The Run Time System

As an example, consider the case where N_o equals 131 (levels), N_t equals 1,024 (levels, for a 32-by-32 template), and N_r equals 512 (levels). Thus R equals 2, meaning that the R-bit shifter drops the least-significant 2 bits. N_i equals 521 (levels); the dither array is normalized by $d[x_d,y_d] = \text{int}\{\Delta_d(T[x_d,y_d] + ½)\}$ with $\Delta_d = 1/256$; and the gain factor to be included in the modified adjust LUT is 520/511. This data is loaded into the run time system in Figure 7 and uniformly maps input pixels across the 131 true output levels, giving the illusion of 512 levels. To better see the relative sizes of the various numbers of levels involved, a plot of this example is shown in Figure 8.

The goals of run-time rendering system is modify the input to the projector so that display will be uniform in a way that is computationally efficient, and elimination of the spurious contours that result because of the fact that there are only N_o uniquely displayed levels.

The Front End LUT accounts for nonlinear projector response, a multiply with the Spatial Attenuation Array fixes nonuniformity, the normalized Dither Array and Shift performs the mean-preserving dither and the Back End LUT maps the resulting N_o quantized levels to the actual unique values of the projector.

As the size of the attenuation array $a_p(x_p, y_p)$ in Figure 7 can be large, it can be subsampled by factors of two in one or both directions. For every halving in size in a given dimension, one less bit of pixel address is used to access it. This has the effect of approximating the attenuation as constant in blocks of size 2^m by 2^n, where m and n are the subsample rates in the x and y dimension. However, as the block sizes increase, the appearance of sharp discontinuities becomes more acute. It would be useful in future work to measure the tradeoff between visual quality and attenuation array compression.

8. Color spatial variation

For single panel projectors, the run-time system described would be replicated for the red, green, and blue channel. For multiple panel systems, it is possible to have independent variations in each separate color. In such cases, a separate camera capture array is built for each color, and the associated projector correction LUTs are generated separately. As with other color dithering systems, the dither arrays for each color plane would be different to reduce texture.

9. Conclusion

The reverse lookup method described here directly accounts for both the nonlinear luminance response of both the camera and the projector without needing to know or measure the absolute response of either! The run-time system is simple and very accurately corrects the nonuniformity. Also, because of the direct measurement approach, nonuniformities beyond those inherent in the projector can be accounted for. This includes irregularities in the screen and the case of oblique projection.

While other projector display systems have employed dither, the method described in this paper detects the actual unique output levels and produces a mean-preserving dither specifically to those unique levels. This spatial dither system results in the perception of higher numbers of luminance levels without disturbing flicker as can occur in temporal dither approaches. This approach uses the full range available and results in a smoother transition between levels.

Acknowledgement

The author gratefully acknowledges the contributions of summer intern Himanshu Arora. The solutions presented in this paper were validated by his arduous photometric measurements, and camera-projector system implementations

References

[1] Chen, H., R. Sukthankar, G. Wallace, K. Li. Scalable Alignment of Large-Format Multi-Projector Displays Using Camera Homography Trees. *Proceedings of Visualization*, 2002.

[2] Majumder, A. and R. Stevens, LAM: Luminance Attenuation Map for Photometric Uniformity in Projection Based Displays, *ACM Symposium on Virtual Reality Software and Technology* (VRST), Nov. 11-13, 2002.

[3] Stone, M., Color Balancing Experimental Projection Displays, *9th IS&T/SID Color Imaging Conference*, Apr 1, 2001.

[4] Sukthankar, R. ,Tat-Jen Cham; G. Sukthankar, J. Rehg, D. Hsu, T. Leung, "Self-calibrating camera-projector systems for interactive displays and presentations", Computer Vision, and ICCV, 2001.

[5] Ulichney, R., "Halftoning", *Wiley Encyclopedia of Electrical and Electronics Engineering*, Vol. 8, pp. 588-600, John Wiley & Sons, Inc., 1999.

[6] Ulichney, R., "The Void-and-Cluster Method for Generating Dither Arrays," *IS&T/SPIE Symposium on Electronic Imaging Science & Technology*, San Jose, CA, vol. 1913, Feb. 1-5, 1993, pp. 332-343.

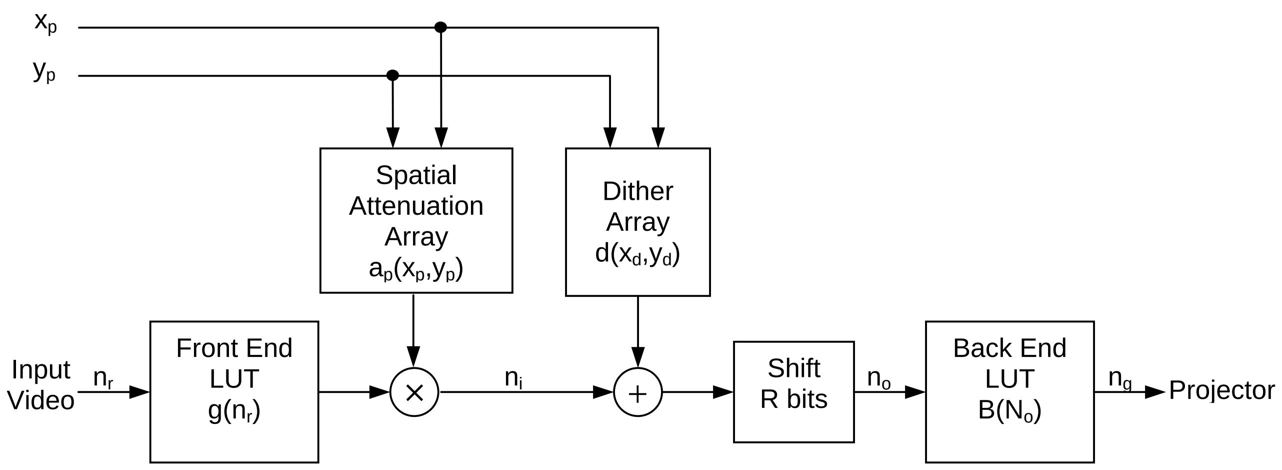

Figure 7. Run-time system

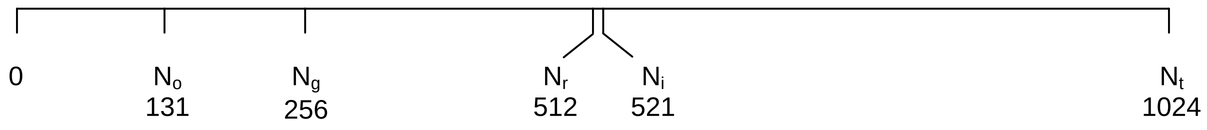

Figure 8. Example relative sizes of each of the level counts