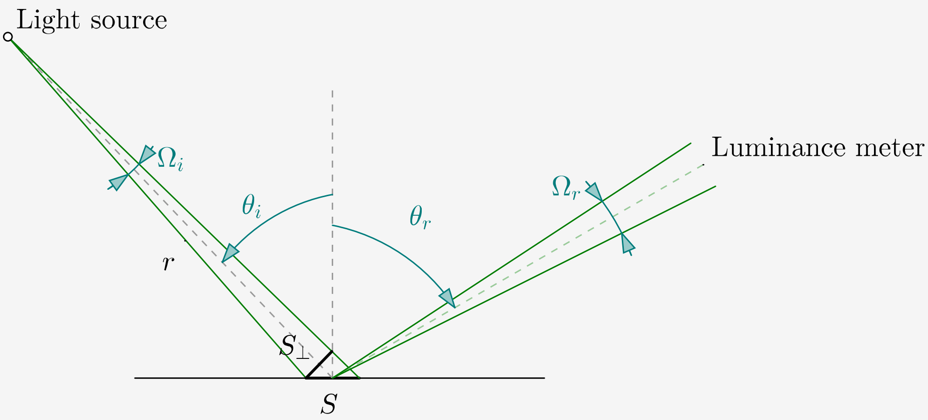
1 Intro: basic definitions and Oren-Nayar result



Flux ($\approx$ power, watt) from a point source on area S: $\Phi_i = I_i\Omega_i$ (lumen), I_i = luminous intensity of the source, cd=lm/sr. Assuming S, and solid angles Ω are small: $\Omega_i = S_\perp/r^2 = S\cos\theta_i/r^2$.

Illuminance (lm/m^2 = lux) *from a point-like source* at distance r:

$$E_0 \equiv \frac{I_i}{r^2} = \frac{\Phi_i}{S_\perp} \tag{1}$$

Lambertian diffusion, by definition, is when the luminance (cd/m^2) *of an extended light source* (i.e., of the illuminated surface)

$$L_r = \frac{\Phi_r}{S\cos\theta_r \cdot \Omega_r}$$

is not a function of θ_r. Total reflected (diffused) flux in such case $\Phi_r^T = L_r S\int\cos\theta_r d\Omega_r = L_r S\pi$. ($d\Omega = \sin\theta d\theta d\varphi$, $\int_0^{\pi/2}\cos\theta\sin\theta d\theta = 1/2$; and there will be no more integrals :)). Also, $\Phi_r^T = \rho\Phi_i$, ρ = reflectance. So

$$L_r = \frac{\rho\Phi_i}{\pi S} = \frac{\rho}{\pi}\cdot\frac{I_i}{r^2}\cdot\cos\theta_i = \frac{\rho}{\pi}E_0\cos\theta_i$$

This result can be found in many physics textbooks.

Real surfaces are not lambertian, but many matte ones can be approximately described with Oren-Nayar multiplication term (see Eq. 30 in [1])

$$L_r = \frac{\rho}{\pi}E_0\cos\theta_i\Big\{A + B\max[0,\cos(\varphi_i-\varphi_r)]\sin\bar\alpha\tan\bar\beta\Big\} \tag{2}$$

$$A = 1 - 0.5\frac{\sigma^2}{\sigma^2+0.33},\ B = 0.45\frac{\sigma^2}{\sigma^2+0.09},\ \bar\alpha = \max(\theta_i,\theta_r),\ \bar\beta = \min(\theta_i,\theta_r)$$

σ is "roughness" of the surface, φ are azimuthal angles – in surface plane (not shown on figure above). So, there are two parameters that characterize a matte surface — ρ and σ. σ can be taken from Oren-Nayar study on their web site. For example, plaster ceiling has $\sigma = 0.4$.

The $\{\cdots\}$ term is actually a simplification of a more accurate term given in Eq. 27 in [1] ("more accurate" – in angular dependence; still 2 parameters, ρ and σ).

The Oren-Nayar result (2) is the key because it allows to calculate E_0 if L_r is measured and geometry angles are known. (whether it gives adequate model for all range of angles, or what are the advancements on this topic – I didn't investigate)

2 Calibration and The Two Main Equations

Camera signal is proportional to luminance [1] of the object: $V = kL_r$, here k – some constant depending on the camera (its settings).

[1]See for example Peter Hiscocks (EE professor), Measuring Luminance with a Digital Camera, https://www.ee.ryerson.ca/∼phiscock/astronomy/astronomy.html (type the tilde manually)

See figure below. Let E_{0C} be illuminance at O_1 from some light at L (so that $\theta_i = 0$), and V_C be value of the pixels in the vicinity of O_1 in a photo shot, made with camera at C.

Using (2), they are related as:

$$V_C = kL_r = k\frac{\rho}{\pi}E_{0C}A \tag{3}$$

(A is all what's left of Oren-Nayar term). Express from here $k\rho$, and substitute it in the measurement of L_r at any other point on the screen ($\theta_i \neq 0$):

$$V = V_C\frac{E_0}{E_{0C}}\cos\theta_i\frac{\left\{\cdots\right\}}{A}, \qquad E_0 = E_{0C}\frac{V}{V_C}\frac{A}{\left\{A + B\max[0, \cos(\varphi_i - \varphi_r)]\sin\alpha\tan\beta\right\}\cos\theta_i} \tag{4}$$

This last equation is the main one.

If distance from light to that point O_1 was R_i, then for the $R_{10m} = 10m$-sphere (see below), the illuminance will be (follows from (1)):

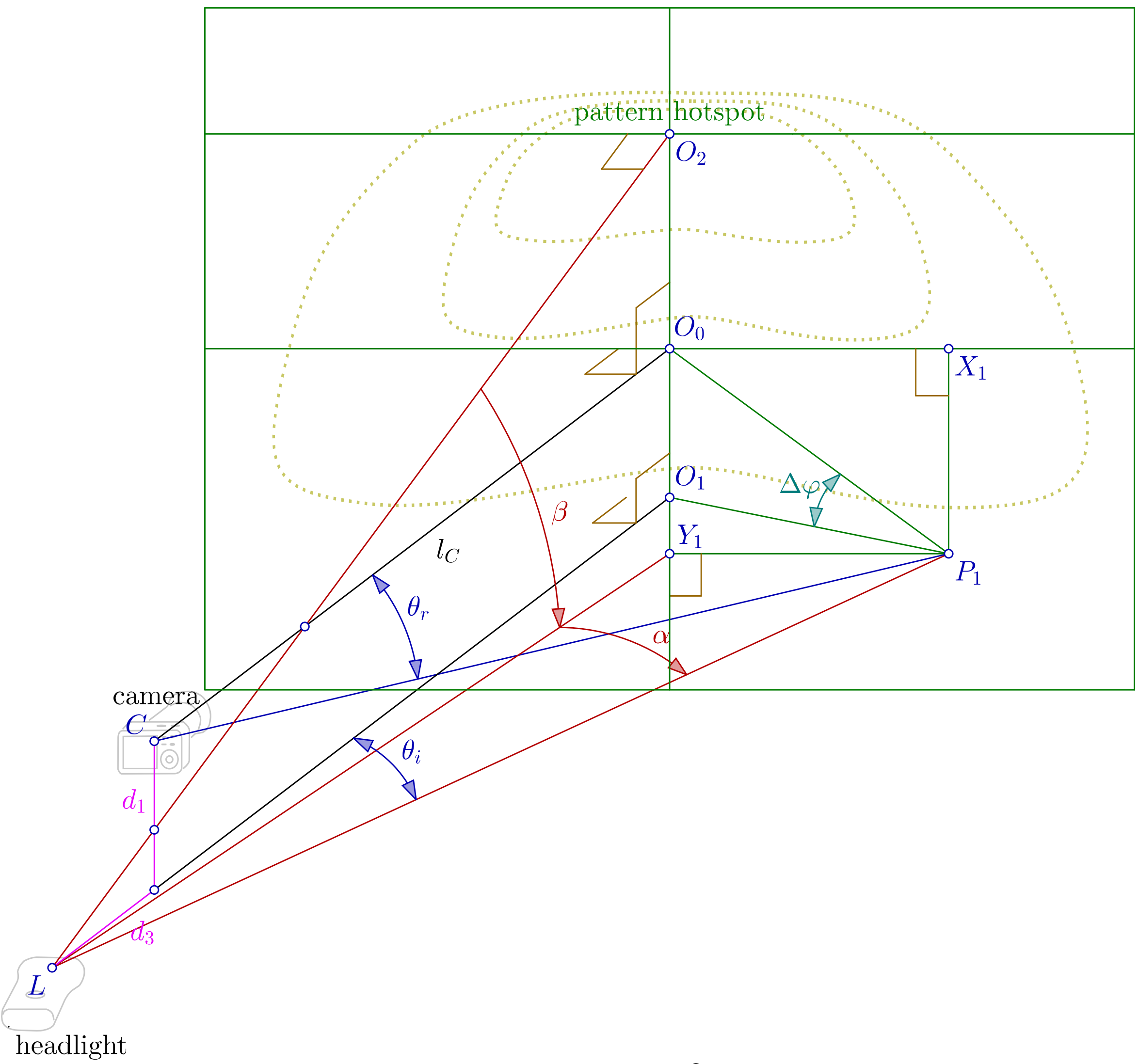
$$E_{10m} = E_0\frac{R_i^2}{R_{10m}^2} \tag{5}$$

The luminance (candela), if needed, can then be obtained by multiplying by squared distance, i.e. $\times 100$ for the 10m-sphere.

These two simple equations, (4) and (5) is essentially all we need. All "physics" stops here. Really.

Note, this calibration is for the single "package": the camera (k) + the matte screen (ρ). With another screen, or another camera – need luxmeter again.

3 Shooting geometry and getting illuminance



Top of screen is the top of camera frame. l_C – distance from camera to screen. In `get_LID.m` all lengths are in centimeters, in `main_vis.m` — in meters. As shown, $d_1, d_3 > 0$ (either can be < 0). α, β increase as shown. $\Delta\varphi$ is the difference of r, i-azimuthal angles for Oren-Nayar term. To fully define the whole geometry we also need (in addition to the 3 distances mentioned) the ratio cm/pixels from a shot of the screen, `scale_cm_px`.

Such calculated angles α, β have the meaning of "pitch" and "yaw" correspondingly (no "roll" needed here) and are very convenient for road illumination calculations. They can also serve as axis on a wall shot. Another definition is used in automotive, see appropriate section below.

Direct axis x to the right from the center O_0, y down. These coordinates x, y of the points on screen (that linearly map to camera pixels) are stored in matrices `XX,YY` in the code.

Now straightforward trigonometry and pythagorean theorem give all what's needed for Oren-Nayar term – see `get_LID.m` file. (no equations because they are high-school-trivial). $\cos\Delta\varphi$ can be determined from scalar product of appropriate vectors. Then use (4) to get illuminance at each point on the screen, then (5).

To calculate β, we first need to determine the coordinates of the hotspot center (where light intensity is maximal). Then β is a sum of two angles (sharing the common ray LO_1 – light perpendicular). Again, straightforward trigonometry and pythagorean theorem.

The function `get_LID()` returns, in matrices of size $n_1 \times n_2$ (n_X=`crop_szX/shrink_fact`): α-s, β-s (in radians), and light intensities.

"Wall" visualization in `main_vis.m` is also straightforward. (except shifting the pattern in α, it's non-trivial: see appropriate subsection below).

3.1 HDR and image smoothing

The camera raw images are saved (with raw converter, like ufraw or rawtherapee) to standard 8-bit jpeg images, with *linear* value encoding (gamma=1). In `get_LID.m` these jpegs are first cropped to remove unnecessary objects. Then several images are combined into single high dynamics range "image" (a matrix of type `double`). This resulting image is unnecessary large, so we shrink it, with `shrink_fact` factor (hardcoded in `get_LID.m` to be 10).

The shrinking (in `matrix_scale_down.m`) is Fourier-based (zeroing large frequencies), so we simultaneously get rid of noise.

References

[1] M. Oren and S.K. Nayar, "Generalization of Lambert's Reflectance Model" *ACM 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 239-246, Jul. 1994, http://www.cs.columbia.edu/CAVE/projects/oren/