High Dynamic Range Photography

Computational Photography (CSCI 3240U)

Faisal Z. Qureshi
http://vclab.science.ontariotechu.ca
OntarioTech UNIVERSITY

High Dynamic Range (HDR) Photography

- Human eyes have high visual range
  - It can differentiate and see structure between very bright and very dark regions in a scene
- An image is taken at a particular exposure setting, which determines whether it contains more visible structure in brighter or darker regions

Images taken at various exposure settings

High Dynamic Range (HDR) Photography

- Human eyes have high visual range
  - It can differentiate and see structure between very bright and very dark regions in a scene
- An image is taken at a particular exposure setting, which determines whether it contains more visible structure in brighter or darker regions

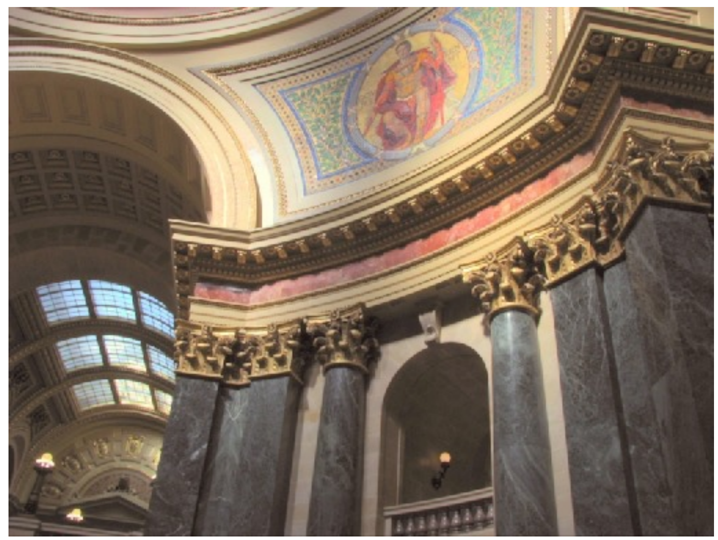
HDR techniques combine multiple images of the same scene to emulate the high visual range of human eyes

Images taken at various exposure settings

High Dynamic Range (HDR) Photography

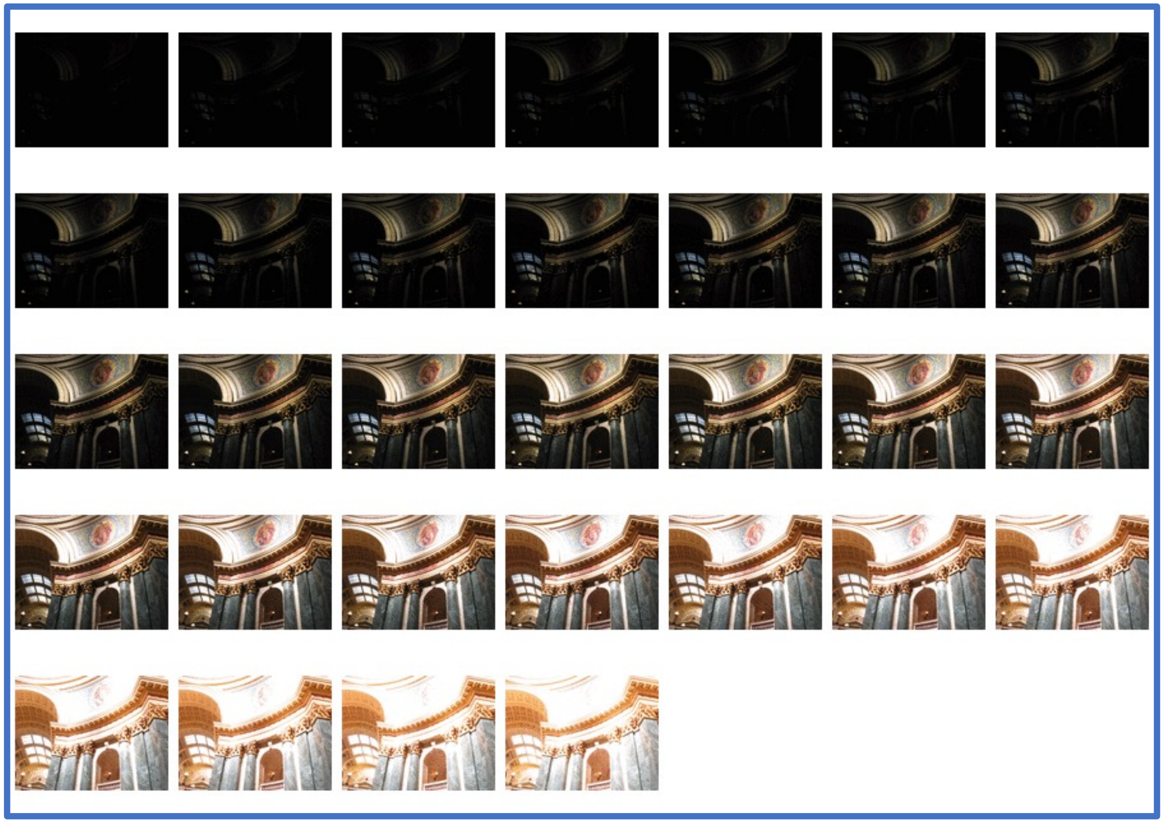
Images taken at various exposure settings

HDR techniques combine multiple images of the same scene to emulate the high visual range of human eyes

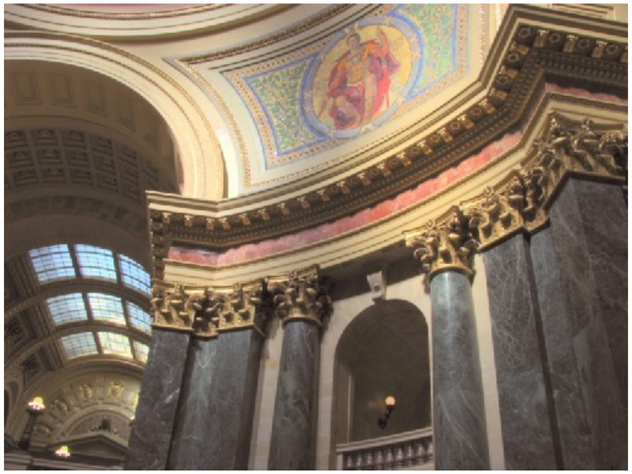


HDR image

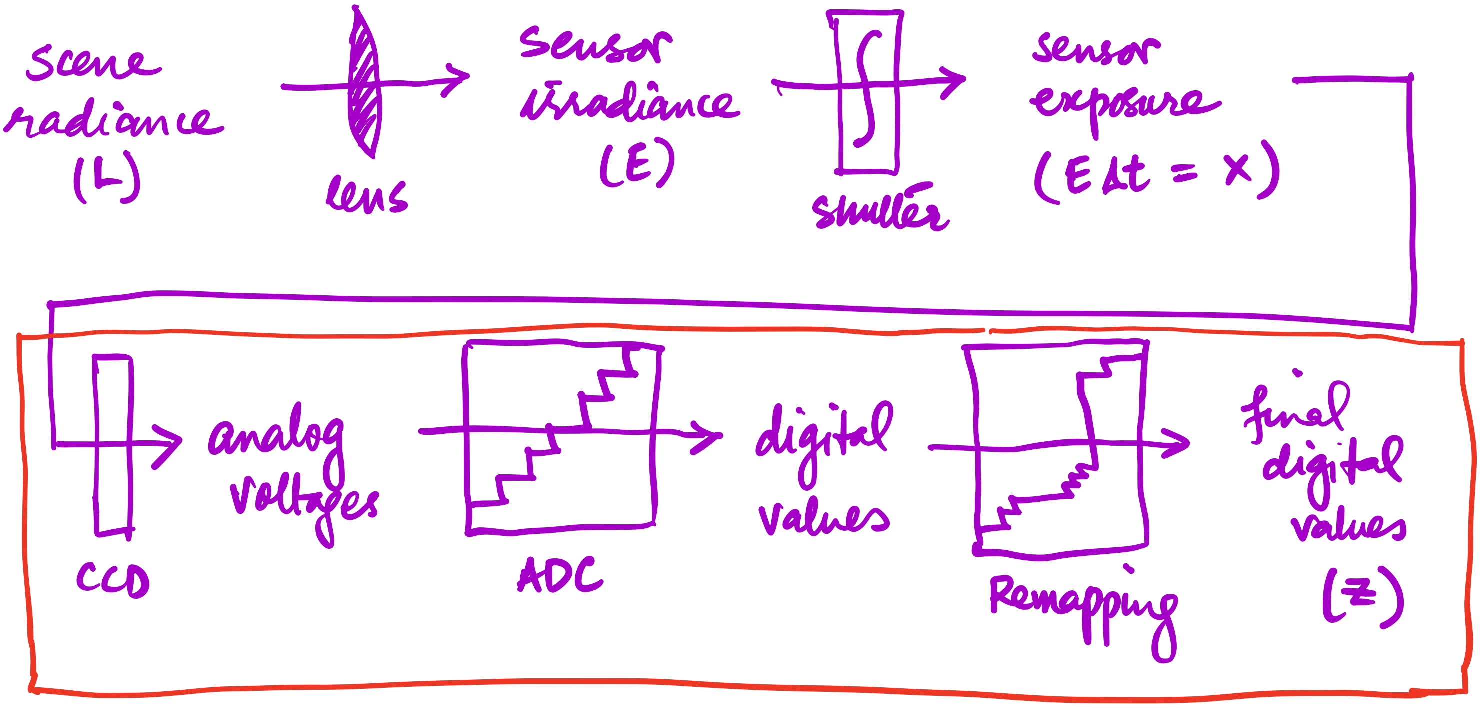
High Dynamic Range (HDR) Photography



Images taken at various exposure settings



HDR image



$$Z = f(E\Delta t)$$

observed pixels

camera response function

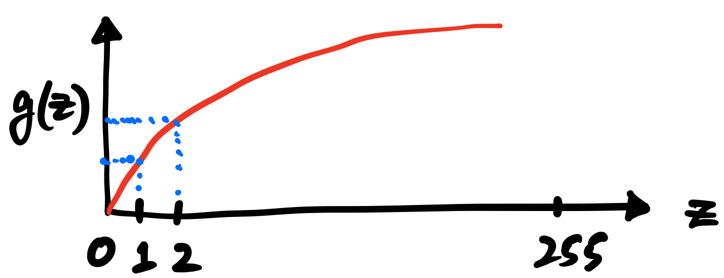
Goal: Compute the camera response function (f).

$$Z = f(E\Delta t)$$

$$f^{-1}(Z) = E\Delta t$$

$$\underbrace{\log f^{-1}(Z)} = \log E + \log \Delta t$$

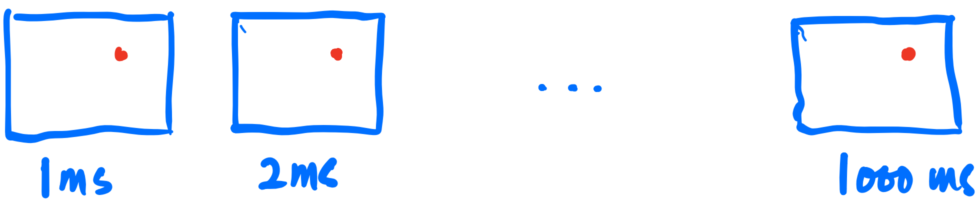
$$g(Z) = \log E + \log \Delta t \quad \text{--- (1)}$$

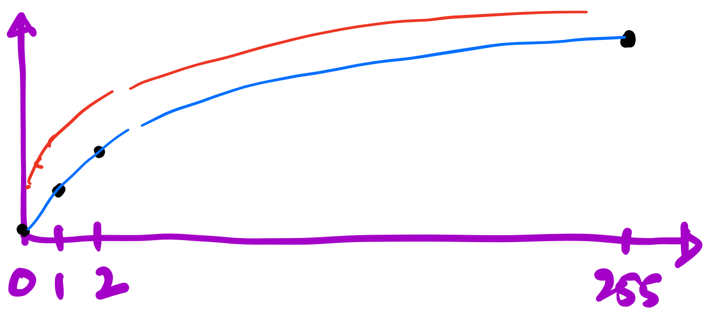


We need to compute $g(z)$: $\underbrace{g(0), g(1), g(2), \cdots, g(255)}_{256 \text{ values}}$.

Approach 1

- One pixel and many many images
- Images are taken by finely adjusting Δt
- Plot $\underline{\log \Delta t}$ as a function of pixel intensity.





Drawbacks: need to acquire too many images.
how to control Δt to hit all possible z values between 0 and 255.
what if you store 16-bit pixels. Here possible values are between 0 and $2^{16}-1$.
Does not scale!!

What happens $\log E$?

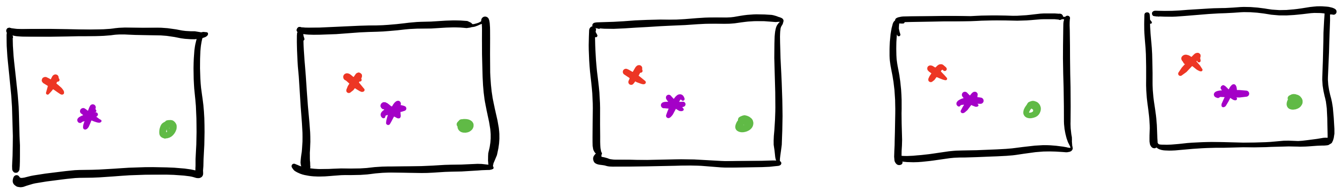
$$g(z) = \boxed{\log E} + \log \Delta t \rightsquigarrow g(z) \approx \log \Delta t$$

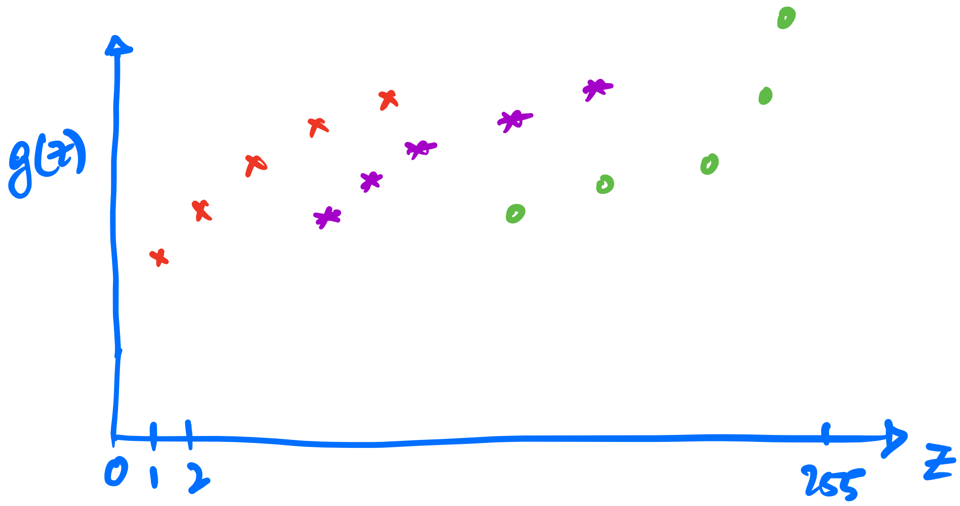
simplified

same value for all the images

Approach 2

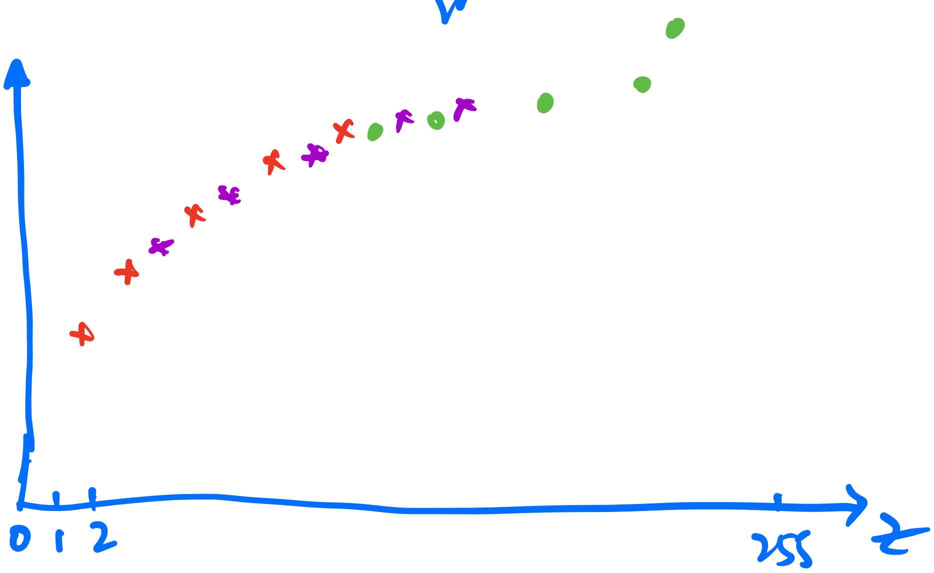
- Few pixels and a few images.





Plot of g(z) from three images.

Computed g(z)



Benefit over approach 1:

we need far far fewer images.

Approach 2 (mathematics).

N pixels

P images.

Example:

$N = 3$

$P = 5$

$$g(z_{ij}) = \log E_i + \log \Delta t_j \qquad (2)$$

i^{th} pixel

in j^{th} image

irradiance for i^{th} pixel

exposure interval of j^{th} image.

Unknowns: $g(0), \dots, g(255)$ [256]

E_i [N]

$x_{ij} \in [0, 255]$

Given Δt_j

\# of unknowns = $256 + N$

\# of equations = NP

Simplified notation:

$$g(z_{ij}) = g_{ij}$$

$$\log E_i = e_i$$

$$\log \Delta t_j = \delta_j$$

Re-write (2)

$$g_{ij} = e_i + \delta_j$$

$$\Rightarrow \quad g_{ij} - e_i = s_j \quad \text{—③}$$

I want to express this as $Ax = b$

$$x = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{255} \\ e_1 \\ \vdots \\ e_N \end{bmatrix} \in \mathbb{R}^{(256+N)\times 1}$$

$g_0 = g(0)$

$\vdots$

$g_{255} = g(255)$

Express ③ as $Ax = b$

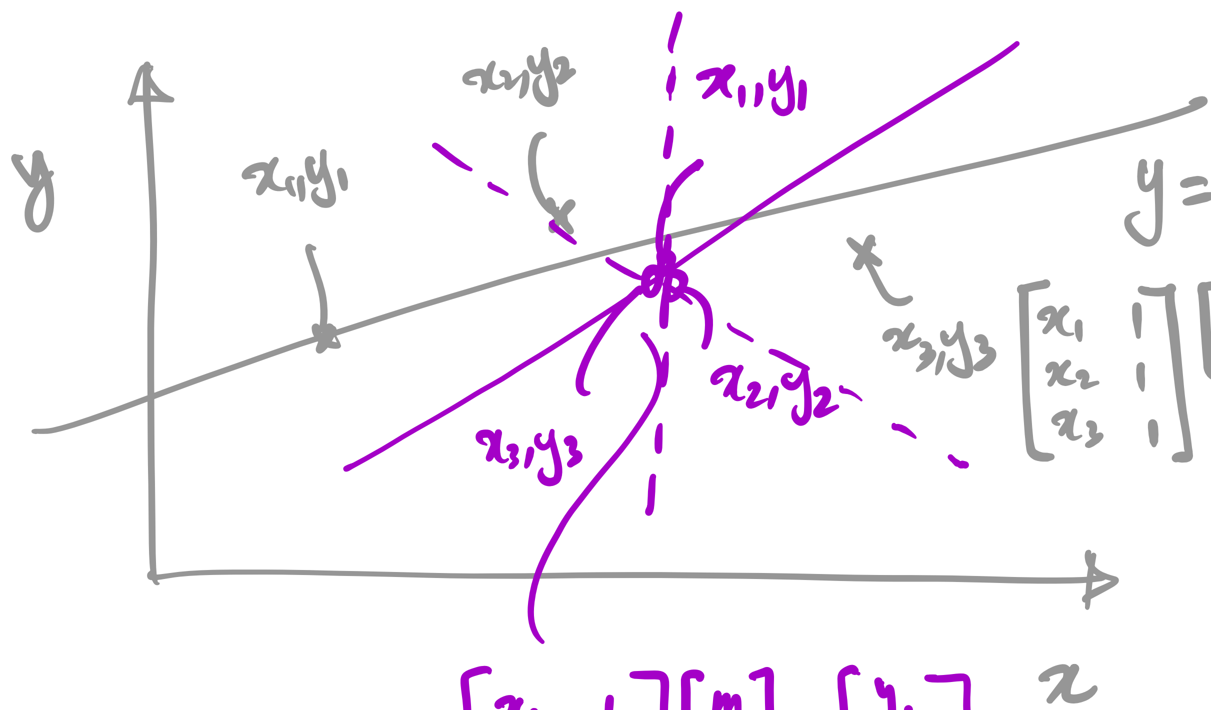
$$\begin{array}{c} \overset{256}{\longleftrightarrow} \qquad\qquad \overset{N}{\longleftrightarrow} \\ \left[\, 0 \;\cdots\; 0 \;\; \underset{z_{ij}^{\text{th}} \text{ column}}{1} \;\; 0 \;\cdots\; 0 \;\; 0 \cdots 0 \;\; \underset{256+i \text{ column}}{-1} \;\; 0 \;\cdots\; 0 \,\right] \end{array} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{255} \\ e_1 \\ \vdots \\ e_N \end{bmatrix} = [s_j]$$

co-efficients of $g_0 \cdots g_{255}$

co-efficients of $e_1 \cdots e_N$

$$
\begin{array}{c|c}
 & \text{column } z_{ij} \qquad\qquad \text{column } 256+i \\
\hline
ij^{\text{th}} \text{ row} & \left[\begin{array}{cccccccccccc} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & -1 & 0 \;\cdots\; 0 \end{array}\right]
\end{array}
\begin{bmatrix} g_0 \\ \vdots \\ g_{255} \\ e_1 \\ \vdots \\ e_N \end{bmatrix}
=
\begin{bmatrix} \\ \\ \delta_j \\ \\ \\ \end{bmatrix}
$$

$$
A \in \mathbb{R}^{NP\times(256+N)} \qquad x \in \mathbb{R}^{256+N} = b \in \mathbb{R}^{NP}
$$



$y = mx + c$

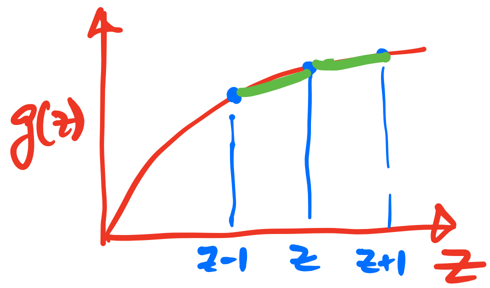
$$
\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}
\begin{bmatrix} m \\ c \end{bmatrix}
=
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}
$$

$$
\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}
\begin{bmatrix} m \\ c \end{bmatrix}
=
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}
$$

Consider $Ax = b$. What if A is ill-conditioned? How do we address this issue?

$\rightarrow$ "Regularize"

Real camera response curves (functions) varies smoothly.



Add more equations to enforce smoothness

$$g(z) - g(z-1) = g(z+1) - g(z) \quad *$$

Re-write this as

$$2g(z) - g(z-1) - g(z+1) = 0$$

$$\begin{array}{c} \left[\begin{array}{ccc|c|c|c|ccc} 0 & \cdots & 0 & -1 & 2 & -1 & 0 & \cdots & 0 \end{array}\right] \\ \begin{array}{ccc} \hspace{3em} z \text{ column} & z+1 \text{ column} & z+2 \text{ column} \hspace{3em} \end{array} \end{array} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{z-1} \\ g_z \\ g_{z+1} \\ \vdots \\ g_{255} \end{bmatrix} = [0]$$

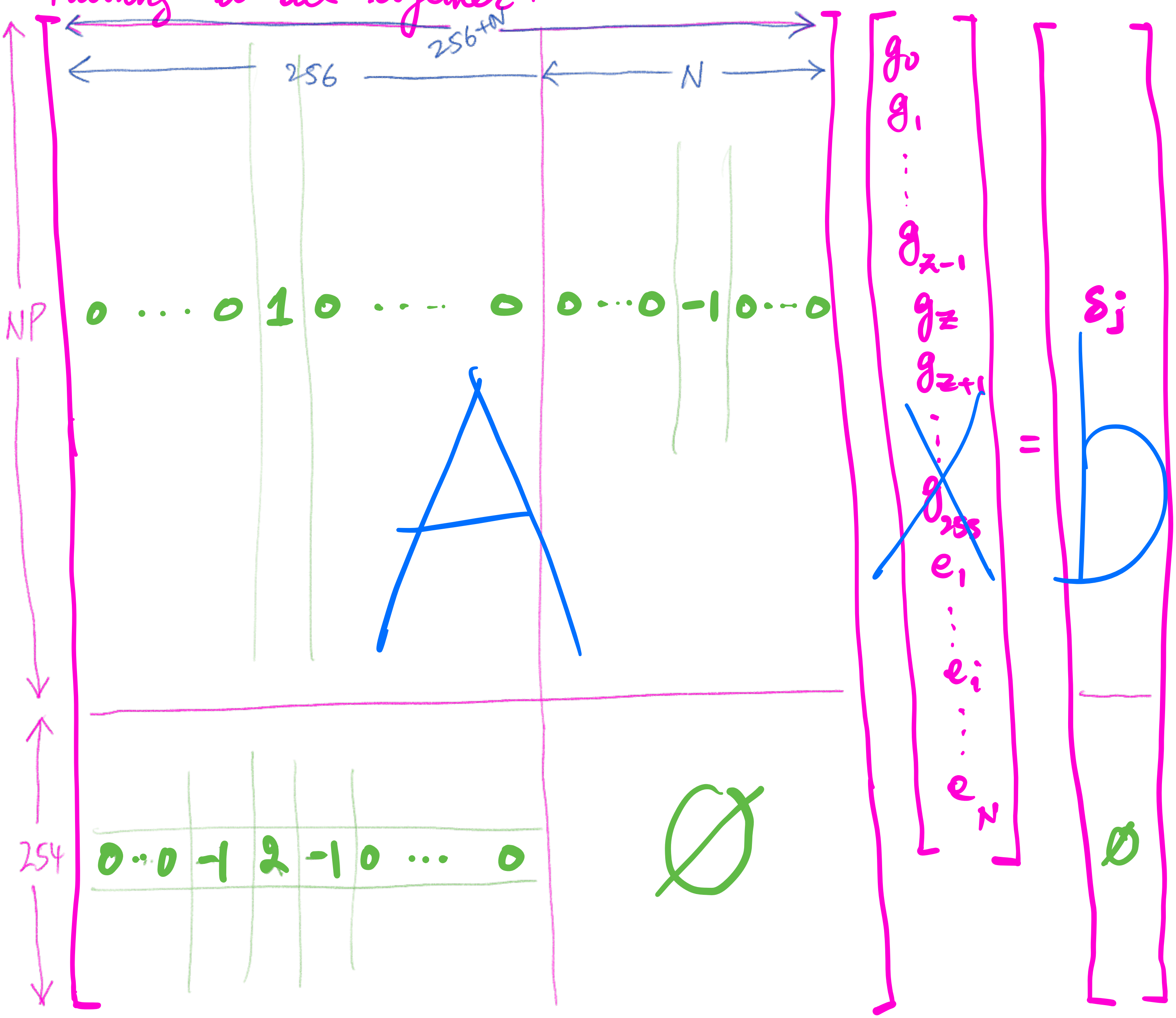
For a particular g_z, g_{z-1}, g_{z+1}

We have $g_0, \dots, g_{255}$. We extend the idea to all possible values of $z \in [0, 255]$.

We get 254 more equations:

$$\left.\begin{aligned}
2g_1 - g_0 - g_2 &= 0 \\
2g_2 - g_1 - g_3 &= 0 \\
&\vdots \\
2g_{254} - g_{253} - g_{255} &= 0
\end{aligned}\right\} 254$$

Putting it all together:



$$g(z) = \log E + \log \Delta t$$

$$\Rightarrow \log E = g(z) - \log \Delta t.$$

irradiance of that pixel.

shutter speed

While $z \in [0, 255]$, $E \in [0, \infty]$ (HDR)

$\boxed{\text{tone-mapping}}$ because irradiances can not be displayed on a standard display.

Process of compressing a range to the available range.

For HDR logarithmic tone-mapping:

- Given an image I.
- $I' = \log(1+\alpha I) / \log(1+\alpha)$
  - $\rightarrow \alpha = 0.1$, controls the strength of tone-mapping. smaller results in stronger compression.
- $I'' = [I' - \min(I')] / [\max(I') - \min(I')]$
  - $\rightarrow$ remapping pixel values between 0 and 1.
- $R = (I'' \times 255.0) \rightarrow$ int8

result.

High Dynamic Range (HDR) Photography

- Collect images at different exposure settings
  - Align images if needed
- Estimate response curve
  - Set up the system of linear equations and solve for unknowns
- Use response curve to compute pixel irradiance values
- Perform tone mapping to construct the final HDR image that can displayed on a device with limited dynamic range

Reference

- https://pages.cs.wisc.edu/~csverma/CS766_09/HDRI/hdr