Moderovacie a renderovacie techniky

František Dráček
dracek1@uniba.sk

7. októbra 2025

https://github.com/frantisekdracek/Prezentacie/tree/main

Tone mapping

- HDR merging: LDR -> HDR e.g: combine several pictures with different exposure into HDR image
- tone mapping: HDR -> LDR , HDR image to display's dynamic range
- Paper
- Tutorial

Debevec algorithm

The exposure, or "amount of light" hitting camera can be assumed to follow relation

$$X_{ij} = E_i \delta t_j, \tag{1}$$

where E_i radiance, δt_j exposure time, i spatial index and j time index. Photographic film response is a function of exposure:

$$Z_{ij} = f(E_i \delta t_j). \tag{2}$$

We can invert the relation and simplify by taking logarithm:

$$g(Z_{ij}) = \ln(f^{-1}(Z_{ij}) = \ln(E_i) + \ln(\Delta t_j). \tag{3}$$

What are we trying to do?

Our goal is to recover original exposures $E_i j$, so we can construct HDR image. To achieve this, we need to know camera response curves $g(Z)$ for every channel.. Since Z takes on only integer values in range $\{0, \ldots, 255\}$, we can transform this to optimization problem.

Minimalization in least squares sense:

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} (g(Z_{ij}) - \ln(E_i) - \ln(\Delta t_j))^2, \qquad (4)$$

N number of pixels, P number of images.

Debevec algorithm

Regularization term to prevent excessive curliness:

$$\mathcal{O} = \sum_{i=1}^{N}\sum_{j=1}^{P}(g(Z_{ij}) - \ln(E_i) - \ln(\Delta t_j))^2 + \lambda \sum_{Z_{min}+1}^{Z_{max}-1} g''(x)^2 \quad (5)$$

Weight term ensuring better fit:

$$\mathcal{O} = \sum_{i=1}^{N}\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln(E_i) - \ln(\Delta t_j))^2 + \lambda \sum_{Z_{min}+1}^{Z_{max}-1} w(Z_{ij}) g''(x)^2 \quad (6)$$

$$w(z) = \begin{cases} z - Z_{min} & \text{if } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{if } z \geq \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \quad (7)$$

Sampling

We dont have to use all pixels:

$$NP \geq Z_{max} - Z_{min} + N \tag{8}$$

Sampling matrix:

$$S = 256 \left\{ \underbrace{\begin{bmatrix} \dots & \dots & \dots \\ \dots & S_{zj} & \dots \\ \dots & \dots & \dots \end{bmatrix}}_{P} \right. \tag{9}$$

- Select middle exposure image .
- for selected z localize position of all pixels with intensity z. $[(k_1, l_1,), (k_2, l_2), \dots]$
- select one random pair (k, l).
- find intenisty values on other images correspoding to (k, l)
- $S_{z,j} = Z_{i(k,l),j}$

Least squares

Standard linear regression

Linear regression problem:

$$\mathsf{A}\mathsf{x} = \mathsf{b}, \tag{10}$$

Least squares solution:

$$\mathcal{O} = \min ||\mathsf{A}\mathsf{x} - \mathsf{b}||^2 \tag{11}$$

By taking gradient of condition and setting it to zero $\nabla_{\mathsf{x}}\mathcal{O} \overset{!}{=} 0$:

$$\mathsf{x} = (\mathsf{A}^T\mathsf{A})^{-1}\mathsf{A}^T\mathsf{b} \tag{12}$$

Least squares

Standard linear regression - how to formulate optimization condition as matrix problem

We aim to formulate

$$\mathcal{O} = \sum_{i=1}^{N}\sum_{j=1}^{P}(g(Z_{ij}) - \ln(E_i) - \ln(\Delta t_j))^2 \quad (13)$$

as

$$\mathcal{O} = \min ||\mathbf{A}\mathbf{x} - \mathbf{b}||^2 = \min ||\sum_{l} A_{kl}x_l - b_k||^2 \quad (14)$$

.
This can be realized by choosing

$$x = \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(255) \\ \ln(E_1) \\ \vdots \ln(E_P) \end{bmatrix}, \quad (15) \qquad b = \begin{bmatrix} \ln(\Delta t_1) \\ \vdots \\ \ln(\Delta t_P) \\ \ln(\Delta t_1) \\ \vdots \\ \ln(\Delta t_P) \\ vdots \\ \ln(\Delta t_1) \\ \vdots \\ \ln(\Delta t_P) \end{bmatrix}. \quad (16)$$

Where x is a vector of size $256 + N$, with N is number of pixel locations.
Matrix A is has $(N * P)$ rows and $256 + N$ columns. It is zero everywhere, except a column indexed by Z_{ij}, where it is 1 and column indexed by $256 + i$, where it is -1. Rows of the matrix are index by i, j with rown indices $r = N * i + j$.
Vector b is just N times repeat exposure times $\ln(\delta t_j)$:

Least squares

Ridge regression

Linear regression problem:

$$\mathrm{A}\mathrm{x} = \mathrm{b}, \tag{17}$$

Least squares solution with Ridge regularization:

$$\mathcal{O} = \min ||\mathrm{A}\mathrm{x} - \mathrm{b}||^2 + \lambda||\mathrm{x}||^2 \tag{18}$$

Solution:

$$\mathrm{x} = (\mathrm{A}^T\mathrm{A} + \lambda\mathbb{I})^{-1}\mathrm{A}^T\mathrm{b} \tag{19}$$

Equivalent to standard linear regression with modified matrix $\tilde{\mathrm{A}}$.

$$\tilde{\mathrm{A}} = \begin{bmatrix} A \\ \sqrt{\lambda}\mathbb{I} \end{bmatrix}. \tag{20}$$

Least squares

Second derivative regularization

Linear regression problem:

$$\mathsf{Ax} = \mathsf{b}, \tag{21}$$

Least squares solution with second derivative regularization:

$$\mathcal{O} = \min ||\mathsf{Ag}(x) - \mathsf{b}||^2 + \lambda||\mathsf{g}''(x)||^2 \tag{22}$$

Equivalent to:

$$\tilde{\mathsf{A}} = \begin{bmatrix} A \\ \sqrt{\lambda}\mathsf{D} \end{bmatrix}, \tag{23}$$

where:

$$\mathsf{D} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & \cdots \\ 0 & 1 & -2 & 1 & \cdots & \cdots \\ 0 & 0 & 1 & -2 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \tag{24}$$

Debevec to simple least squares

transform problem to matrix

$$\begin{bmatrix} \mathrm{A} & \mathrm{E} \\ \mathrm{D}_w & 0 \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(255) \\ \ln(E_1) \\ \vdots \end{bmatrix} - \left[\ln(\nabla t_1)) \right], \tag{25}$$

where

$$\mathrm{D}_w = \begin{bmatrix} w(1) & -2w(1) & w(1) & 0 & \cdots & \cdots \\ 0 & w(2) & -2w(2) & w(2) & \cdots & \cdots \\ 0 & 0 & w(3) & -2w(3) & w(1) & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \tag{26}$$

Debevec to simple least squares

$$\mathrm{b} = \begin{bmatrix} w(Z_{1,1}) \ln(\nabla t_1) \\ w(Z_{1,2}) \ln(\nabla t_2) \\ \vdots \\ w(Z_{1,P}) \ln(\nabla t_1) \\ w(Z_{2,1}) \ln(\nabla t_1) \\ \vdots \\ w(Z_{2,P}) \ln(\nabla t_P) \\ \vdots \\ w(Z_{N,1}) \ln(\nabla t_1) \\ \vdots \\ w(Z_{N,P}) \ln(\nabla t_P) \end{bmatrix} \tag{27}$$

Debevec to simple least squares

Matrix A is zero matrix except for following terms:

$$A_{P*(i-1)+j,Z_{i,j}} = w(Z_{i,j}) \tag{28}$$

Matrix E is zero matrix except for following terms:

$$E_{P*(i-1)+j,i} = w(Z_{i,j}) \tag{29}$$

Algorithm

```
# 1. Add data-fitting constraints:
k = 0
for i in range(num_samples):
    for j in range(num_images):
        z_ij = intensity_samples[i, j]
        w_ij = weighting_function(z_ij)
        mat_A[k, z_ij] = w_ij
        mat_A[k, (intensity_range + 1) + i] = -w_ij
        mat_b[k, 0] = w_ij * log_exposures[j]
        k += 1

# 2. Add smoothing constraints:
for z_k in range(z_min + 1, z_max):
    w_k = weighting_function(z_k)
    mat_A[k, z_k - 1] = w_k * smoothing_lambda
    mat_A[k, z_k     ] = -2 * w_k * smoothing_lambda
    mat_A[k, z_k + 1] = w_k * smoothing_lambda
    k += 1

# 3. Add color curve centering constraint:
mat_A[k, (z_max - z_min) // 2] = 1
```

Reconstructing radiance map

$$\ln(E_i) = \frac{1}{P}\sum_{j=1}^{P}(g(Z_{ij}) - \ln(\nabla t_j)) \qquad (30)$$

With weights:

$$\ln(E_i) = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln(\nabla t_j))}{\sum_{j=1}^{P} w(Z_{ij})} \qquad (31)$$

Reconstructing radiance map

- Last step involves computing radiance map for every channel and combining them to a resulting HDR image.
- Follow with tone mapping techniques to map image to LDR for displaying purpose.

Thank you!