Dynamic Exposure and Gain Control in a Dynamic Environment

Demetris Demetriades

Abstract

Based on the noise-aware exposure control method in [1], we extend the algorithm to operate reliably in dynamic environments rather than in the static conditions assumed in the original paper. Our approach incorporates motion masking, temporal smoothing, and adaptive noise estimation to achieve stable exposure and gain updates under continuous scene changes.

1 Dynamic evolution of Nelder-Mead (DNM) optimization algorithm

1.1 Inputs of DNM

1. Read current Exposure
2. Read current Gain
3. Read Low-Res Gray-scale/Luminance image $\sim I_{grey}$
4. Read Low-Res color/Bayer image

We require only one of the last two inputs (3 or 4), depending on the capabilities of the Arducam SDK.

1.2 Parameters of DNM

- γ=0.6, Mapping threshold.
- λ=10, Mapping control.
- N_c=100, Grid cells.
- p=0.1, Homogenous threshold.
- τ_l=15, Saturation bound.
- τ_h=235, Saturation bound.

- K_g=2, Normalization factor.
- K_e=0.125, Normalization factor.
- $\alpha = \beta = 0.4$, Weighting factors.
- ε=1.7, Constant.

Caution! The parameters τ_l and τ_h are defined for an 8-bit image; for images with different bit depth, τ_l and τ_h must be adjusted accordingly.

1.3 DNM algorithm

$$f_{\mathrm{dyn}}(I) = \alpha\,\mathcal{L}^{\mathrm{smooth}}_{gradient}(t) + (1-\alpha)\,\mathcal{L}_{\mathrm{entropy}}(t) - \beta\,\sigma^{\mathrm{smooth}}_{\mathrm{noise}}(t), \quad t = \text{number of frame.} \tag{1}$$

1.4 Constructing I_{grey}

If the Arducam SDK cannot provide a **luminance image** and only supplies a **Bayer image**, we must construct I_{grey} ourselves. The dimensions of the I_{grey} matrix depend on the image resolution. Assume a 1080p resolution, corresponding to 1920 × 1080 pixels. For each pixel, the SDK provides three values corresponding to the colour channels [R_i, B_i, G_i].

$$I_{gray} = 0.299\,R_i + 0.587\,G_i + 0.114\,B_i, \quad \text{Matrix size } 1920 \times 1080. \tag{2}$$

2 Constructing the parameters of DNM

2.1 Constructing $\mathcal{L}^{\mathbf{smooth}}_{gradient}(t) \equiv \mathcal{L}^{\mathbf{s}}_{g}(t)$

Purpose of $\mathcal{L}^{\mathrm{s}}_g$: $\mathcal{L}^{\mathrm{s}}_g$ is used to create a stable measure of image quality based on gradients. The raw gradient value changes too quickly when the camera or objects in the scene move, which would make exposure control unstable. By smoothing $\mathcal{L}_g$, we keep a more consistent signal across frames, allowing the auto-exposure system to react smoothly instead of jumping.
Gradient-based quality: Gradient-based quality means evaluating how much structure, detail, and edge information the image contains. Strong and uniformly distributed gradients indicate a sharper and better-exposed image, while weak gradients usually mean blur, low contrast, or underexposure.
Constructing mapping function $\tilde{g}_i$:

$$\tilde{g}_i = \begin{cases} \frac{1}{N_g}\,\log(\lambda\,(g_i - \gamma) + 1)\,, & g_i \geq \gamma \\ 0, & g_i < \gamma \end{cases} \tag{3}$$

$$N_g = \log(\lambda(1-\gamma) + 1)$$

Meaning of $\tilde{g}_i$: $\tilde{g}_i$ is the mapped (compressed) gradient value obtained from the raw gradient g_i by applying three steps:

- **Thresholding:** ignore weak gradients (noise) when $g_i < \gamma$.
- **Logarithmic mapping:** compress very strong gradients so that edges do not dominate the metric.
- **Normalization:** scale the mapped values into a stable range.

Purpose:

- remove noise by discarding small gradients ($g_i < \gamma$),
- reduce the influence of very large gradients,
- make gradients more balanced across the entire image,
- improve stability of the overall image-quality metric used for exposure control.

To calculate mapping function we have first to calculate the gradient magnitude g_i:

$$g_i = \sqrt{G_x(i)^2 + G_y(i)^2}, \quad i = \text{num of pixel.} \tag{4}$$

$$G_x(i) = (I * K_x)(i), \qquad G_y(i) = (I * K_y)(i)$$

Where K_x, K_y are the Sobel horizontal and vertical kernels:

$$K_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

The dimension of $G_x, G_y, g_i, \tilde{g}_i \in \mathbb{R}^{H \times W}$.
The gradient-based image quality metric based on grid-level $\tilde{g}_i$ statistics order:

$$G_i = \sum_{j=1}^{N_c} \tilde{g}_i, \quad N_c = \text{numbers of grid cells in image.} \tag{5}$$

We introduce after the **Motion masking**:

$$V(i) = \begin{cases} 1, & \text{if } |I^t_{grey} - I^{t-1}_{grey}| < \tau_m, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Motion masking removes pixels that change significantly between consecutive frames, since these changes are caused by motion rather than true scene structure. By ignoring moving pixels, the algorithm prevents motion blur or object movement from corrupting the gradient and noise measurements used for exposure control. The global gradient-based image quality metric order:

$$G_i^{adj} = w_i \cdot \tilde{G}_i, \quad \tilde{G}_i = \sum_{j=1}^{N_c} \tilde{g}_i \cdot V(i). \tag{7}$$

The first 34 grid cells have weight $w_i = 0.25$, the next 33 cells have $w_i = 0.5$, and the last 33 cells have $w_i = 0.25$. The upper cells receive lower weight because they contain sky with low useful detail, the middle cells receive higher weight because they capture the main scene structure, and the lower cells receive lower weight because they often contain ground regions with weak gradients.
Final gradient metric:

$$\mathcal{L}_g^{\text{final}}(t) = K_g \, \frac{E(G^{adj})}{s(G^{adj}) + \varepsilon}. \tag{8}$$

Where $E(G^{adj})$ and $s(G^{adj})$ are the mean value and standard deviation. Finally the $\mathrm{L}_g^{\mathrm{s}}(t)$:

$$\mathcal{L}_{gradient}^{\text{smooth}}(t) = \lambda_g \, \mathcal{L}_g^{\text{smooth}}(t-1) + (1-\lambda_g) \, \mathcal{L}_g^{\text{final}}(t). \tag{9}$$

Caution! The constants λ_g must define under experiments possible theoretical value in range (0.8-0.95).

2.2 Constructing $\mathcal{L}_{entrophy}(t)$

L_{entropy} measures how widely the pixel intensities are distributed in the image. Higher entropy indicates better contrast and exposure, while low entropy corresponds to flat or poorly exposed regions. It stabilizes the exposure metric in dynamic scenes because it is less sensitive to motion and local edges, providing a complementary global quality measure:

$$\mathcal{L}_{\text{entropy}}(t) = -K_e \sum_{k=0}^{255} P_I(k) \, \log_2 P_I(k). \tag{10}$$

Where for an 8-bit image, the intensity values range from 0 to 255. The $P_I(k)$ is the probability of intensity value on the $I_{grey}(t)$:

$$P_I(k) = \frac{N_k}{N}. \tag{11}$$

Where N_k is the occupation number of intensity value and N the number of pixels.

2.3 Constructing $\sigma_{\textbf{noise}}^{\textbf{smooth}}(t) \equiv \sigma_{\mathbf{n}}^{\mathbf{s}}(t)$

σ_{noise} provides a measure of the sensor noise level by averaging the absolute response of a noise–sensitive filter over homogeneous and unsaturated pixels, allowing the algorithm to estimate how much random variation (noise) is present in the image.
We introduce first the noise detection Kernel:

$$M = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}. \tag{12}$$

The kernel M fails in regions containing edges (i.e., object boundaries), textured areas, and saturated regions where $k \approx 0$ or $k \approx 255$. To ensure that the matrix M operates only on regions containing actual noise, we introduce two additional masks, $H(i)$ and $U(i)$.
The order of $H(i)$:

$$H(i) = \begin{cases} 1, & g_i \leq \delta \\ 0, & g_i > \delta. \end{cases} \tag{13}$$

Where δ corresponds to the value at the 10% position of the ordered set of g_i, sorted from smallest to largest. We take all the values of g_i , sort them from smallest to largest, and the value at the 10% position is taken as δ.With $H(i)$ we ensure that the matrix M does not operate on edges or textured regions. The order of binary mask $U(i)$:

$$U(i) = \begin{cases} 1, & \tau_l < I^t_{grey} < \tau_h \\ 0, & otherwise. \end{cases} \tag{14}$$

With $U(i)$ we ensure that the matrix M does not operate on saturated regions. Using the eq. (6) the σ^{final}_{noise} order:

$$\sigma^{\text{final}}_{\text{noise}}(t) = \sqrt{\frac{\pi}{2}} \frac{\sum_i H(i)\, U(i)\, V(i)\; \left|(I^t_{grey} * M)\right|(i)}{\sum_i H(i)\, U(i)\, V(i)}. \tag{15}$$

Finally the $\sigma^{\text{s}}_{\text{n}}(t)$ order:

$$\sigma^{\text{smooth}}_{\text{noise}}(t) = \lambda_n\, \sigma^{\text{smooth}}_{\text{noise}}(t-1) + (1-\lambda_n)\, \sigma^{\text{final}}_{\text{noise}}(t). \tag{16}$$

Caution! The constants λ_n must define under experiments possible theoretical value in range (0.8-0.95).

3 Application of DNM algorithm

Knowing how $f_{\text{dyn}}(t)$ is computed, we can now update the exposure and gain dynamically. The controller tries to keep the image quality close to a desired target value f_{target}.

$$e(t) = f_{\text{target}} - f_{\text{dyn}}(t). \tag{17}$$

The controller adjusts exposure and gain proportionally to the error. A positive error increases exposure/gain, while a negative error decreases them. The exposure is updated using a proportional step:

$$\text{Exposure}(t+1) = \text{Exposure}(t) + k_{\text{exp}}\, e(t). \tag{18}$$

The gain is updated in the same way:

$$\text{Gain}(t+1) = \text{Gain}(t) + k_{\text{gain}}\, e(t). \tag{19}$$

Values of k_{exp} and k_{gain} are determined empirically through experiments.

3.1 Output values of DNM

$$\boxed{\begin{gathered}\text{Exposure}(t+1) = \text{Exposure}(t) + k_{\text{exp}}\, e(t) \\ \text{Gain}(t+1) = \text{Gain}(t) + k_{\text{gain}}\, e(t)\end{gathered}}$$

References

[1] Ukcheol Shin, Jinsun Park, Gyumin Shim, Francois Rameau, and In So Kweon. Camera exposure control for robust robot vision with noise-aware image quality assessment. *arXiv preprint arXiv:1907.12646*, 2019.