Grupo de Investigación HDSP
Escuela de Ingeniería de Sistemas e Informática, Universidad Industrial de Santander

---

Taller de Laboratorio No. 2
Gamma Corrections, Geometric Transformations, and Representation in Another Basis

---

Workshop overview

Pipeline: `linear` → `exposure` → `gamma` → `warp` → `DCT` → `metrics`

Output: Notebook (`.ipynb`) + report (2–4 pages) with figures, a small metrics table, and discussion.

Checklist

- A personal image (not a standard course image).
- Comparison: **exposure in linear** vs **exposure in gamma**.
- A figure with histograms (two different gamma values).
- A small PSNR/SSIM table (interpolators or DCT).
- A short paragraph on "decisions and fixes": what was changed and why.

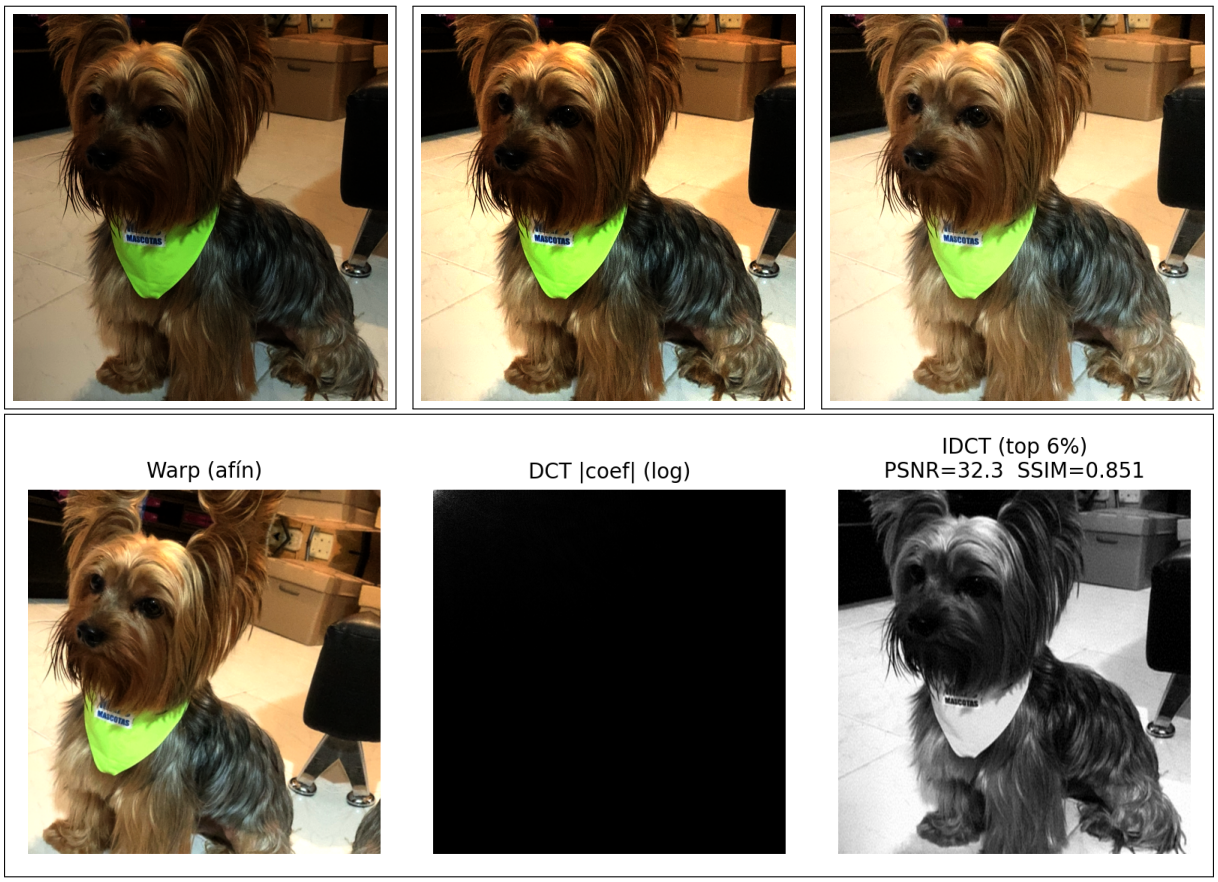
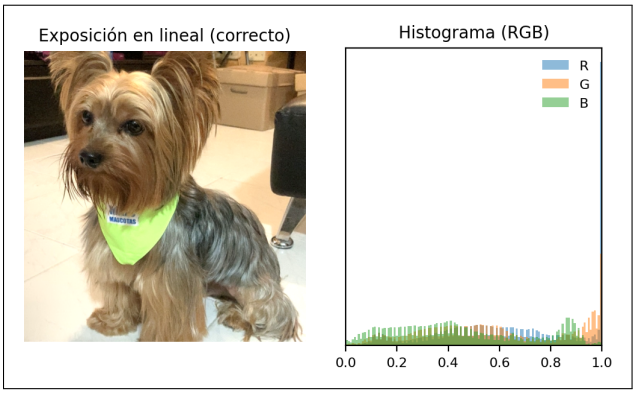


Figure 1: Preview of the lab pipeline: linearization, exposure, gamma correction, geometric transformations, and DCT-domain representation.

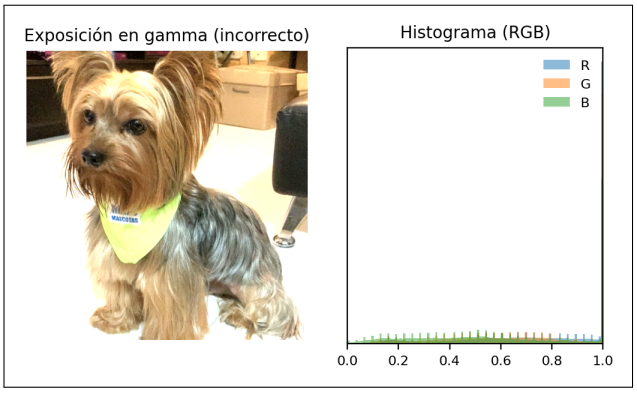
1 Theory

1.1 Motivation

Most common mistakes in image processing are not caused by "bad filters", but by applying operations in the **wrong domain**. In a real camera, the intensity measured by the sensor is approximately **linear** with the incoming light; however, what is typically displayed and stored (e.g., sRGB) is **encoded** using a nonlinearity (gamma). Therefore, adjustments such as exposure/illumination should be performed in the linear domain, while other visual effects are interpreted in the encoded domain. In this lab, a complete pipeline is implemented in Python to: (i) linearize and apply gamma correction, (ii) geometrically transform an image (understanding the role of resampling), and (iii) represent the image in a frequency basis (DCT) to analyze compression and loss of detail.



(a) Exposure applied in the **linear domain** (correct).



(b) Exposure applied in the **gamma domain** (incorrect).

Figure 2: Key comparison: the same operation yields different results depending on the working domain.

1.2 Gamma correction as a radiometric mapping

Let $r \in [0, 1]$ be a **linear** intensity proportional to radiance/irradiance (ideally, the sensor output after normalization). A power-law (gamma) transform is defined as

$$s \;=\; T_\gamma(r) \;=\; r^\gamma, \qquad r \in [0, 1],\ \gamma > 0, \tag{1}$$

and its inverse ("undo gamma") is

$$r \;=\; T_\gamma^{-1}(s) \;=\; s^{1/\gamma}. \tag{2}$$

Interpretation:

- $\gamma < 1$ expands shadows (increases detail in dark regions).
- $\gamma > 1$ compresses shadows (darkens), preserving more relative contrast in highlights.

Operations that model energy/illumination (e.g., exposure: multiplying by η) should be applied in the **linear** domain and not directly in the gamma domain.

1.3 Geometric transformations and sampling (interpolation)

A geometric transformation applies a coordinate mapping $x' = \varphi(x)$, with $x = (x, y)$, and requires **resampling** on a discrete grid. For affine transformations, a homogeneous form is

used:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{3}$$

Visual quality strongly depends on the interpolator: *nearest*, bilinear, bicubic. This impacts edge sharpness, aliasing, and ringing.

1.4 Representation in another basis: DCT as an example

A change of basis represents the image vector $x \in \mathbb{R}^N$ in a basis Φ:

$$\alpha = \Phi^\top x, \qquad x = \Phi\alpha. \tag{4}$$

A fundamental case for images is the **Discrete Cosine Transform (DCT)**, which tends to concentrate energy in low frequencies for natural images. In compression, *top-k* selection or truncation is applied to α and $\hat{x}$ is reconstructed, evaluating degradation with metrics (PSNR/SSIM) and visual evidence.

2 Objectives

Learning objectives

1. Implement and analyze gamma correction and its inverse for linear and non-linear images.
2. Show how gamma modifies histograms and local/global contrast.
3. Implement affine geometric transformations (rotation, scaling, shear, translation) and compare interpolators.
4. Represent images in the DCT domain and study compression/reconstruction via sparsification.
5. Connect acquisition (exposure/linearity) with visualization (gamma) and processing (warps + basis change).

3 Tools (Python)

Use `Python 3.10+` and libraries: `numpy`, `matplotlib`, `scikit-image`, `scipy`. `opencv-python` is optional.

Note

Export figures from the notebook (e.g., `plt.savefig(...)`) to easily include them in the report.

4 Experiments

4.1 Experiment 1: linear → gamma → display pipeline

Note

Mission: recover shadow detail without destroying highlights.
Deliverable: comparative figure + histograms + a 5-line conclusion.

Purpose: distinguish between linear intensities (sensor-like) and encoded intensities (display-like).

1. Load an RGB image and convert it to float in $[0, 1]$.
2. Simulate exposure by multiplying by a factor η and clipping: $r_\eta = \text{clip}(\eta r, 0, 1)$.
3. Apply gamma $s = T_\gamma(r_\eta)$ for $\gamma \in \{0.4, 0.7, 1.0, 1.6, 2.2\}$.
4. Plot: resulting image and its histogram for each γ.
5. **Key comparison:** repeat exposure directly in the gamma domain (incorrect) and compare against the linear-domain case.

Listing 1: Gamma and exposure (implementation hint).

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, img_as_float

def gamma_map(r, gamma):
    r = np.clip(r, 0.0, 1.0)
    return np.power(r, gamma)

img = img_as_float(io.imread("input.jpg"))  # RGB in [0,1]
eta = 1.8                                   # exposure
r = np.clip(eta * img, 0, 1)

gammas = [0.4, 0.7, 1.0, 1.6, 2.2]
```

```python
for g in gammas:
    s = gamma_map(r, g)
    plt.figure(figsize=(10,3))
    plt.subplot(1,2,1); plt.imshow(s); plt.title(f"gamma={g}");
       plt.axis("off")
    plt.subplot(1,2,2)
    plt.hist(s[...,0].ravel(), bins=256, alpha=0.6, label="R")
    plt.hist(s[...,1].ravel(), bins=256, alpha=0.6, label="G")
    plt.hist(s[...,2].ravel(), bins=256, alpha=0.6, label="B")
    plt.legend(); plt.title("Histograms")
    plt.tight_layout()
    plt.show()
```

4.2 Experiment 2: geometric transformations + interpolation

Note

Mission: show that transforming implies resampling and that the interpolator matters.
Deliverable: interpolator comparison + PSNR/SSIM table + error map.

Purpose: demonstrate that geometric transforms imply resampling (and can introduce aliasing).

1. Implement an affine transform H (rotation + translation) and apply a *warp*.
2. Repeat with *nearest*, bilinear, and bicubic interpolation.
3. Apply the inverse H^{-1} to return to the original and measure error.
4. Compute PSNR/SSIM between the original and the reconstructed image.
5. **Question:** which interpolation best preserves fine edges? which introduces less ringing?

4.3 Experiment 3: DCT representation and sparsification-based compression

Note

Mission: achieve good quality while keeping few coefficients.
Deliverable: reconstructions for two percentages + PSNR/SSIM curve vs percentage.

Purpose: represent the image in another basis and analyze the quality/compression trade-off.

1. Convert the image to grayscale (or process channel-wise).
2. Apply block DCT (8×8 or 16×16) or global DCT.
3. Keep only a percentage $p\%$ of the largest-magnitude coefficients and zero the rest.
4. Reconstruct with IDCT and evaluate PSNR/SSIM vs. p.
5. **Question:** why does energy concentrate in low frequencies for natural images?

5 Deliverables

Deliverable	Minimum content
Notebook (`.ipynb`)	Executable code, intermediate figures, short conclusions per experiment, and a decisions/fixes block.
Report (PDF)	2–4 pages with 3 key figures (one per experiment), 1 metrics table, and answers to the analysis questions.
Results	`results/` folder with exported images (recommended).

6 Analysis questions (for the report)

1. Why is it conceptually incorrect to apply "physical" operations (e.g., exposure/linear mixtures) directly in the gamma domain without linearizing?
2. What relationship did you observe between gamma and the histogram distribution? Support your answer with evidence.
3. Explain, with examples, when a geometric transformation amplifies aliasing and what role interpolation plays.
4. In DCT compression: describe the trade-off between sparsity and perceptual quality. Which structures degrade first?