
Politecnico di Torino

EURECOM
Sophia Antipolis


Double Master's Degree in Ingegneria del Cinema e dei Mezzi di Comunicazione and Data Science and Engineering

---

# INVESTIGATING TONE MAPPING SOLUTIONS WITHOUT AFFECTING COLORS

---

**Supervisors**

**Dr. Praveen CYRIAC**

**Prof. Andrea BOTTINO**

**Prof. Jean-Luc DUGELAY**

**Candidate**

**Simone MELCARNE**

**Academic Year 2023-2024**

*A mia nonna, Giovanna.*

*Per tutto ciò che ho perduto*

*e che non ritornerà.*

# Summary

Tone mapping is an essential step in the image processing pipeline to map the content of HDR input images to the reference display. The simplest approach to perform tone mapping is to apply a certain mathematical function to the luminance component of an image and then propagate the change to the triplet of R, G and B values. However, this operation might introduce some color distortion since the luminance and the chrominance are not always well separated and modifying one can affect the other. This project comes to life with the aim of identifying an alternative and more efficient solution to the tone mapping task especially when the application addressed requires to maintain color fidelity.

The core of this investigation is to test several tone mapping operators (TMOs) but in a different domain than the usual one. Specifically, instead of applying this mapping as described above, the objective is to first perform a decomposition of the image and then carry out the tone mapping operation exploiting the newly obtained intrinsic components, which offer unique and detailed information about the image itself.
Upon completing the decomposition phase, the primary aim is to evaluate the efficacy of this splitting process in mitigating color distortion. Furthermore, we seek to ascertain if it can play a constructive role in assisting tone mapping algorithms, ultimately elevating the overall quality of the image.
On the one hand, the decomposition task is mainly tackled by leveraging AI-based solutions. On the other hand, only traditional TMOs are tested. Starting to demonstrate the effectiveness of the methodology with simpler models, more complex ones can be gradually introduced.

The results obtained are evaluated by comparing them with those of other famous methodologies found in the literature, using quantitative metrics and carrying out psychophysical experiments.

# Résumé

Tone mapping est une étape essentielle dans le traitement d'images pour mapper le contenu des images HDR sur l'affichage de référence. L'approche la plus simple pour effectuer le tone mapping consiste à appliquer une certaine fonction mathématique au composant de luminance d'une image, puis à propager le changement aux valeurs des triplets RGB. Cependant, cette opération peut introduire des distorsions de couleur, car la luminance et la chrominance ne sont pas toujours bien séparées, et la modification de l'une peut affecter l'autre. Ce projet a pour but d'identifier une solution alternative et plus efficace pour réaliser la tâche de tone mapping, surtout lorsque le domaine d'utilisation nécessite de maintenir la fidélité des couleurs.

Le coeur de cette étude consiste à tester plusieurs opérateurs de tone mapping (TMOs), mais dans un domaine différent de celui habituel. Plus précisément, au lieu d'appliquer un mapping comme décrit précédemment, l'objectif est d'effectuer d'abord une décomposition de l'image, puis de réaliser l'opération de tone mapping en exploitant les composantes intrinsèques nouvellement obtenues, qui offrent des informations uniques et détaillées sur l'image elle-même.
Une fois la phase de décomposition terminée, l'objectif principal est d'évaluer l'efficacité de ce processus de séparation sur l'atténuation de la distorsion des couleurs. De plus, nous cherchons à déterminer s'il peut jouer un rôle constructif en aidant les algorithmes de tone mapping, élevant ainsi la qualité globale de l'image. D'une part, la tâche de décomposition est principalement abordée en exploitant des solutions basées sur l'intelligence artificielle. D'autre part, seuls les TMOs traditionnels sont testé. Commencer par démontrer l'efficacité de la méthodologie avec des modèles plus simples permet d'introduire progressivement des modèles plus complexes.

Les résultats obtenus sont évalués en les comparant à ceux d'autres approches réputées présentes dans la littérature, en utilisant des mesures quantitatives et en réalisant des expériences psychophysiques.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Huawei Technologies

Huawei Technologies Co. Ltd., established in 1987 in Shenzhen, China, has emerged as a global leader in Information and Communications Technology (ICT). The company's extensive operations span across various sectors including Telecommunications, Consumer Electronics, Semiconductors, and Cloud Computing. Renowned for its substantial investment in Research and Development, Huawei plays a pivotal role in advancing technological innovation. Serving over a third of the world's population, its international presence extends to more than 140 countries.

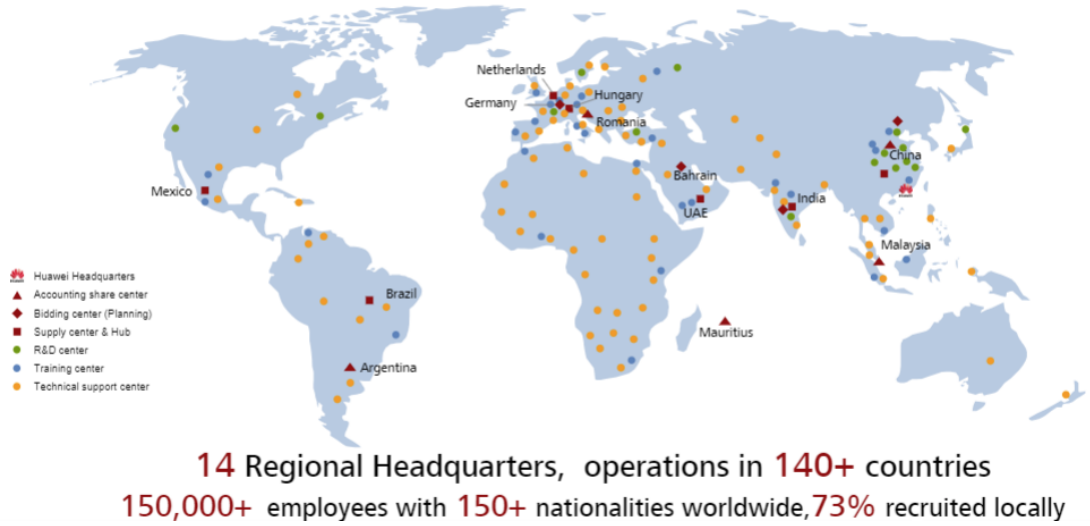


**Figure 1.1:** Huawei's localized operations.

Huawei's commitment to technological development is paralleled by its dedication to social responsibility, with initiatives supporting tech startups, environmental sustainability, and educational projects. These efforts underscore Huawei's vision of creating a fully connected, intelligent world, while promoting sustainable development and fostering young talent in the tech industry. Huawei's mission in France for the past 16 years has centered on fostering a lasting connection between the French population and cultural, innovative, and design initiatives.

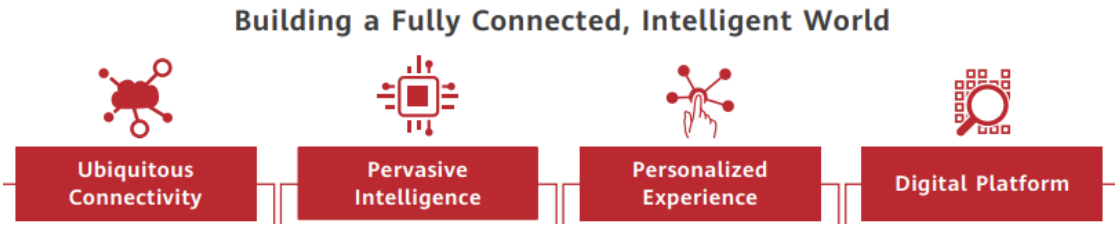


**Figure 1.2:** Huawei's mission.

In the domains of Research and Development, Huawei operates five centers in France with over 100 researchers focusing on 5G, AI, Photography, and Design. Academic collaboration includes partnerships with institutions like the Master DAST Paris-Sud and the Talent Numérique program (Seeds For the Future) with IMT Lille, Telecom Paris Tech, and INSA Lyon. Furthermore, Huawei France contributes to cultural initiatives by supporting institutions such as the Paris Opera and the Museum of Fine Arts in Lille. This multifaceted engagement underscores Huawei's dedication to forging meaningful connections within the French landscape.

## 1.2 Nice Research Center

Situated in Mougins, the *Huawei Nice Research Center* is the second-biggest facility in France, surpassed only by the renowned Paris site. Established in 2013 under Mr. Stephen Busch's leadership, it started with only 9 members whose initial initial mission was to create a complete Image Signal Processor (ISP). The center's focus on ISP technology for smartphone cameras has been pivotal, with constant updates in image processing algorithms for better features. More recently, they've delved into AI domain, recognizing its potential, although mindful of the challenges like the hunger for data and energy. Facing these challenges, the center shifted gears to work on top-notch Computer Vision algorithms for ISPs. Over the years, the focus of this hub has transitioned from predominantly development to achieving a blend of development and research. It has evolved into a nexus where business and

university cultures converge, presenting prospects such as internships for Bachelor's or Master's students and research projects for those pursuing Doctorates.

## 1.3 Project Plan

In the realm of digital imaging and computer graphics, tone mapping stands as a pivotal technique, serving as a bridge between the dynamic range of real-world scenes and the limited capabilities of display devices. A critical aspect under scrutiny is the issue of color distortion encountered in tone mapping due to the potential interference between luminosity and chromatic information meaning that altering one of them may impact the other. This thesis explores the multifaceted landscape of tone mapping, delving into its theoretical foundations, practical applications and algorithmic intricacies with the clear aim of identifying an alternative solution to the aforementioned problem.
An in-depth investigation conducted into methods that can mitigate this type of color artifact suggested a change of the typical domain of action of a generic tone mapping operator (TMO) as a possible solution. The domain shift is achieved by performing a preliminary separation of the image into its two constituent components: *reflectance* and *shading*. This splitting method is known as Intrinsic Image Decomposition (IID) and it is used in computer vision and image processing in order to facilitate a deeper understanding of image content by disentangling intrinsic scene properties from lighting variations. As a matter of fact, the decomposition should be carried out such that the *reflectance* component delivers information about the inherent color and texture of the objects in a scene, while the *shading* component accounts for the illumination effects over those objects.
This project can be seen as broken down into two main steps: the first one dedicated to the study of the IID problem and the search for AI-based solutions and the second one which involves the use of the results obtained in the previous step to test mainly traditional TMOs but paving the way for the eventual examination of more sophisticated approaches that harness the capabilities of Convolutional Neural Networks (CNNs).

## 1.4 Resources and Development Tools

The workload has been spread across a local workstation and a remote server to leverage the computational power of an external Graphics Processing Unit (GPU). The local device is an HP EliteBook 840 G5 equipped with an Intel(R) Core(TM) i7 processor, 16.0 GB RAM, and running a 64-bit Windows 10 Pro system with an x64-based processor. The remote machine, also running a Windows operating system, boasts an NVIDIA GeForce RTX 2080 Ti with 16GB of memory. Python

3.11.5 version was chosen as the main programming language since it is very suitable for dealing with deep learning tasks. The code was written and executed in VS Code, a source-code editor developed by Microsoft.
Various libraries have been exploited throughout the process:

- **argparse** [1]: Argparse is a Python module for parsing command-line arguments, options, and sub-commands. It facilitates the creation of user-friendly command-line interfaces by defining the arguments a script can accept, automatically generating help messages, and handling user input.

- **cv2** [2]: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It contains a collection of tools and algorithms for image and video processing, computer vision tasks, and machine learning.

- **glob** [3]: The glob module provides a function that finds all pathnames matching a specified pattern according to the rules used by the Unix shell. This is useful for searching, matching, and processing file paths based on wildcards.

- **h5py** [4]: The h5py module is a Python interface to the HDF5 (Hierarchical Data Format version 5) library. It allows to store and manipulate large datasets efficiently, providing a convenient way to work with structured data.

- **imageio** [5]: It is a Python library for reading and writing images in a variety of formats. It provides an easy-to-use interface for working with images and supports a wide range of image file formats.

- **matplotlib** [6]: Matplotlib is a 2D plotting library for Python. It produces high-quality visualizations, including charts, plots, histograms, and more. Matplotlib is widely used for creating static, animated, and interactive plots.

- **numpy** [7]: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

- **os** [8]: The os module provides a way to interact with the operating system. It includes functions for working with file systems, accessing environment variables, and executing system commands.

- **pandas** [9]: Pandas is a powerful data manipulation and analysis library. It provides data structures like DataFrames for efficient handling and analysis of structured data.

- **PIL** [10]: Pillow is a modern fork of the Python Imaging Library (PIL). It adds support for opening, manipulating, and saving many different image file formats.
- **random** [11]: The random module provides functions for generating pseudo-random numbers. It is commonly used for tasks involving randomness and probability
- **re** [12]: The re module is Python's regular expression module. It provides functions for working with regular expressions, allowing pattern matching and manipulation of strings based on specific patterns.
- **scipy** [13]: SciPy is an open-source library for mathematics, science, and engineering. It builds on NumPy and provides additional functionality, including optimization, signal and image processing, statistical functions, and more.
- **shutil** [14]: The shutil module provides a higher-level interface for file operations, including file copying, removal, and archiving. It simplifies common file and directory manipulation tasks.
- **skimage** [15]: scikit-image is an image processing library that is part of the scikit-learn ecosystem. It provides algorithms and tools for various image processing tasks.
- **time** [16]: The time module provides functions for working with time, including measuring and manipulating time, as well as dealing with timestamps.
- **torch** [17]: PyTorch is an open-source deep learning library. It provides a flexible and dynamic computational graph, making it suitable for research and production in machine learning tasks.
- **torchvision** [18]: torchvision is a package in PyTorch that provides utility functions and pre-trained models for computer vision tasks. It simplifies working with image datasets and models in the context of PyTorch.
- **tqdm** [19]: tqdm is a library for creating progress bars in the command line interface. It provides a simple way to visualize the progress of iterative tasks, enhancing the user experience during lengthy computations or loops.

In addition, several image processing algorithms have been implemented in **GNU Octave**, mainly exploiting the functions made available by the **HDR ToolBox**, a toolbox for processing High Dynamic Range content, from Banterle et al.[20].

# Chapter 2

# Foundational Concepts

## 2.1 HDR Imaging

HDR images encompass pixels that can portray a significantly broader spectrum of colors and luminosity than what conventional standard dynamic range images can achieve. These enhanced pixels contribute substantially to elevating the overall excellence of visual content, resulting in a heightened sense of realism and visual allure for the audience.
With the following paragraphs the intent is to review the main notions in the context of HDR imaging, moving from the meaning of dynamic range to revealing why it is so important and necessary to carry out the tone mapping operation.

### 2.1.1 Dynamic Range

In engineering, the concept of dynamic range delineates the ratio between the highest and lowest magnitudes within consideration. When applied to images, the observed quantities are the luminance levels which represent a photometric amount that characterizes the perceived intensity of light per surface area. Various dynamic range measures exist depending on the context.
One such metric is the *contrast ratio*, specifically employed in display systems. It characterizes the relationship between the luminance of the brightest color a display can produce (white) and the darkest (black). Normalization is often applied, ensuring the second value is consistently one.

$$\mathrm{CR} = \frac{Y_{\mathrm{peak}}}{Y_{\mathrm{noise}}} : 1 \tag{2.1}$$

Another prevalent measure in high dynamic range imaging is the *log exposure range*. It quantifies the dynamic range of scenes by calculating the base-10 logarithm of the difference between the brightest and darkest luminance. Logarithmic values

are favored for their ability to better represent perceived differences in dynamic range compared to the contrast ratio.

$$D = \log_{10}(Y_{\text{peak}}) - \log_{10}(Y_{\text{noise}}) \tag{2.2}$$

*Exposure latitude*, expressed in logarithm base 2, accounts for the film's capturing capability relative to the photographed scene's luminance range. This measure, denoted in stop steps, is relevant in HDR photography, providing insight into the margin of error photographers can afford in setting exposure parameters while still achieving a satisfactory image.

$$L = \log_2(Y_{\text{peak}}) - \log_2(Y_{\text{noise}}) \tag{2.3}$$

On the digital camera front, the *signal-to-noise ratio (SNR)* is a prevalent metric for expressing dynamic range. Typically measured as the ratio between the intensity causing image sensor saturation and the minimum observable intensity above sensor noise, SNR is presented in decibels using a 20 times base-10 logarithm.

$$\text{SNR} = 20 \cdot \log_{10}\left(\frac{Y_{\text{peak}}}{Y_{\text{noise}}}\right) \tag{2.4}$$

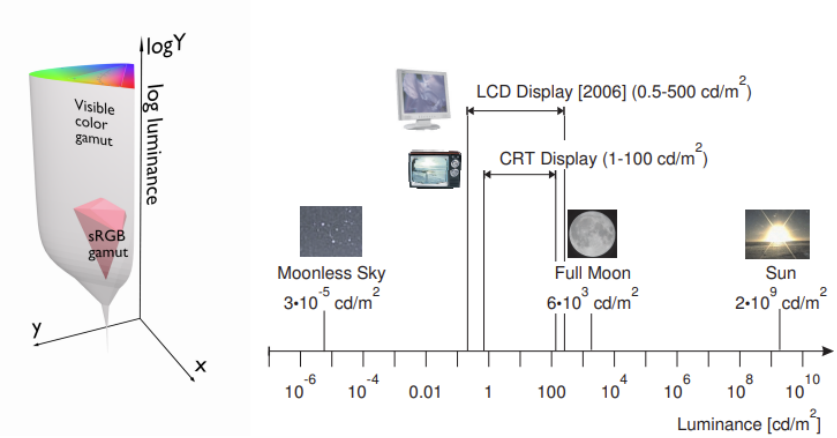


**Figure 2.1:** Luminance range. From [21].

Figure 2.1 depicts the relationship between color perception, sRGB color gamut, and luminance capabilities of traditional monitors. The translucent three-dimensional structure on the left side encapsulates the full spectrum of colors perceptible to the human eye. As it descends towards the base, it visually communicates the gradual reduction in color visibility under lower luminance conditions. In stark

contrast, the enclosed crimson solid represents the well-defined sRGB (Rec. 709) color gamut, a top-tier display known for its faithfully color reproduction. Shifting attention to the right side, the graphic juxtaposes actual luminance values against the luminance range achievable by Cathode Ray Tube (CRT) and Liquid Crystal Display (LCD) monitors. This visual depiction effectively showcases the real-world luminance levels and draws a parallel with the limited luminance capabilities of traditional CRT and LCD monitors. It's noteworthy that much of today's digital content adheres to a format that, at its peak, preserves the dynamic range akin to that of conventional displays.

### 2.1.2 Understanding the Differences: LDR and HDR

Low Dynamic Range (LDR) and High Dynamic Range (HDR) images represent two distinct approaches to capturing and displaying the range of luminance levels in a scene. LDR images are traditional images that have a limited dynamic range, meaning they depict a relatively narrow range of brightness levels from dark to light. Hence, these resulting images can exhibit extremes of brightness, leading to either excessive illumination (overexposed) or insufficient lighting (underexposed). In instances of excessive brightness, the camera captures solely the brighter subset of the infinite luminance range, while conversely, in the case of darker images, only the lower subset is captured.

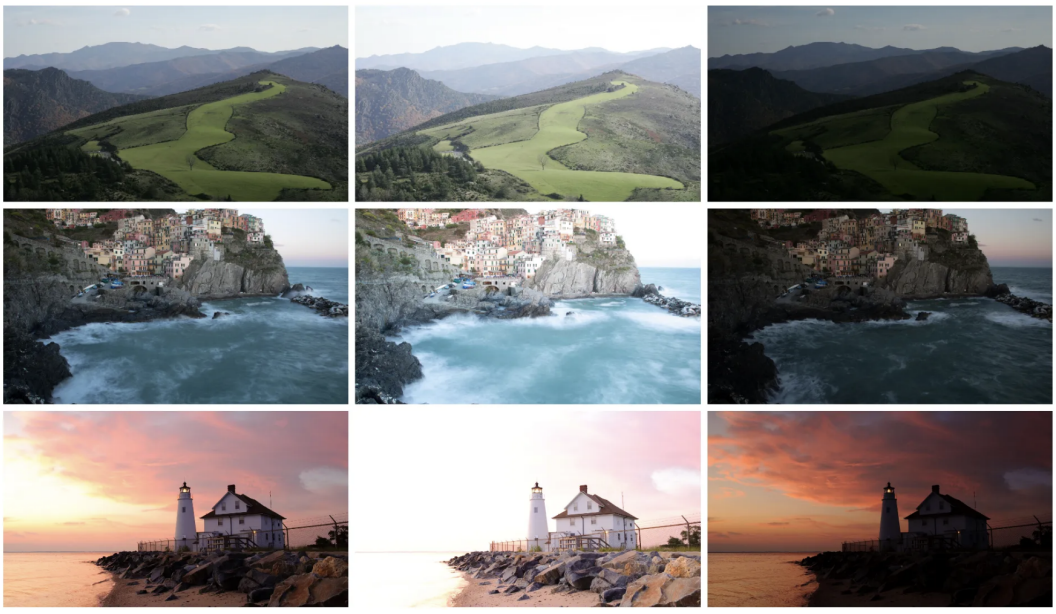

**Figure 2.2:** From left to right: true image, overexposed and underexposed.

In contrast, HDR images aim to capture a broader spectrum of luminance, including details in both the darkest shadows and the brightest highlights. Addressing this challenge involves the amalgamation of images acquired at various exposure levels. The benefit lies in the synergy between overexposed images, adept at capturing details in darker regions, and underexposed images, capable of mitigating intensity in overly bright areas. In this sense, distinct sections of the image are optimally captured at different exposure values. This process is also known as *image bracketing*. The overarching concept is to blend this array of images, enabling the retrieval of a composite image characterized by a heightened dynamic range.

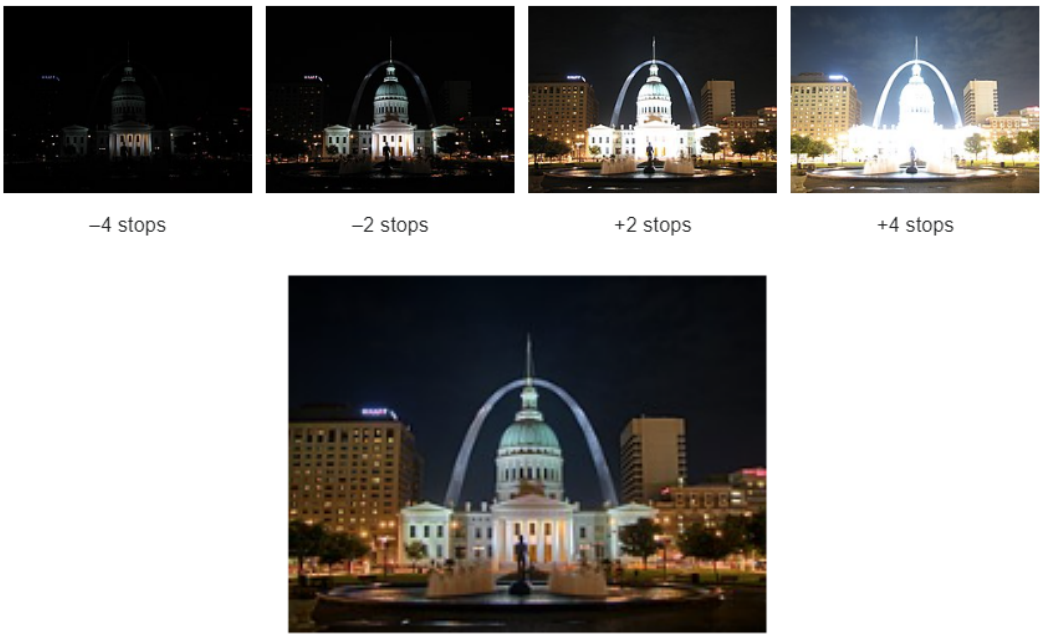


**Figure 2.3:** Multi-exposure HDR capture.

In addition to the differences in capturing dynamic range, it's essential to understand the nuances in how pixel values are represented in HDR and LDR images. In HDR images, pixel values maintain a linear relationship with luminance. This linear relationship allows HDR images to accurately portray the full range of brightness levels in a scene. Conversely, LDR images exhibit a non-linear relationship between pixel values and photometric values. As a matter of fact, in LDR images the brightness values are often adjusted to fit within a specific range. The key factor of this adjustment refers to how display devices function. In the past, when CRT displays were common, the technology involved firing electrons onto a phosphor surface. When the electrons hit the phosphor screen, it emitted photons, creating

the visible image. However, the brightness of the displayed image didn't change proportionally with the strength of the electron beam. To address this issue, the incoming image or video signals were modified in a non-linear way opposite to the display's non-linearity. This adjustment aimed to obtain a more accurate linear representation of the original scene brightness. Even though display technologies have evolved, many devices still exhibit non-linear behavior. To counteract this, the process of adjusting the source image non-linearly is known as *gamma correction*.

$$I_{\text{gamma}} = I^{1/\gamma} \tag{2.5}$$

Essentially, this encoding helps compensate for the non-linear characteristics of displays, ensuring that what is seen on the screen better reflects the true brightness of the captured scene.
Considered an input image denoted as $I$, a display device with a certain value of $\gamma$ shows the image as $I^{\gamma}$. To ensure that the displayed image accurately represents the original, the input image is encoded as $I^{1/\gamma}$.

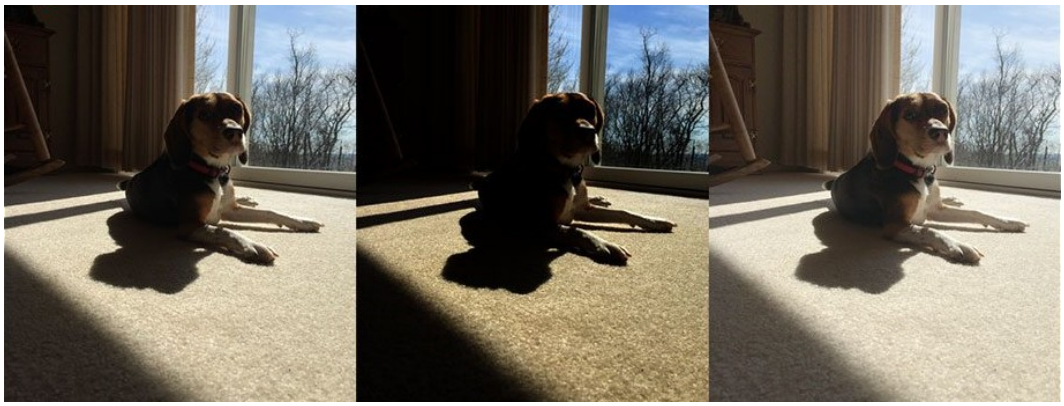

**Figure 2.4:** From left to right: image with $1/\gamma$ values equal to 1, 1.5 and 0.5.

Gamma correction and tone mapping are related concepts, even if they serve different purposes in the context of image processing. The former is more concerned with the perceptual aspects of brightness, whereas the latter deals with adapting images from a high dynamic range to a lower dynamic range.

## 2.2 Tone Mapping

In general terms, tone mapping is intended as a technique employed in digital imaging to manage, somehow, the dynamic range of an image, ensuring that details

in both bright and dark areas are perceptually balanced. This operation becomes crucial when dealing with HDR images as it aims to compress the wide range of luminance levels for display on standard monitors or print media.

### 2.2.1 Purposes of Tone Mapping

When referring to the term *tone mapping*, however, there is a multitude of objectives associated with this type of operation, as well as the methodologies employed and the applications addressed. This often leads to confusion and misunderstandings, making it crucial to clearly delineate these goals. In their work, Mantiuk et al.[21] claim that tone mapping operators can be broadly categorized into three groups:

- **Visual system simulators (VSS)**: They try to emulate human visual limitations, such as introducing glare or adapting for different lighting conditions.

- **Scene reproduction operators (SRP)**: The focus is on preserving the original scene's appearance, particularly on devices with limited color gamut, contrast, and peak luminance.
- **Best subjective quality operators (BSQ)**: They are designed to produce specific images that match the subjective appreciation or the artistic goal.

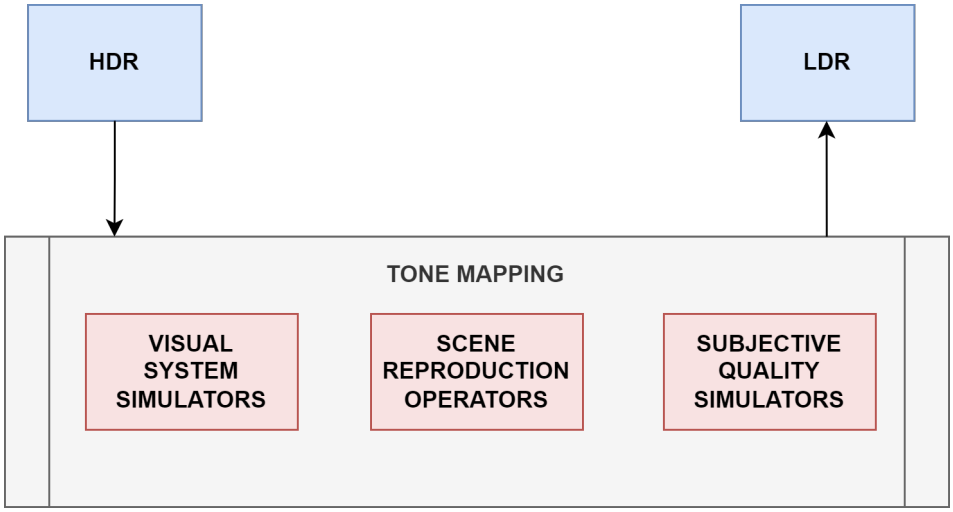


**Figure 2.5:** Tone mapping purposes. Adapted from [21].

Since the objective of this work is to find tone mapping solutions that do not alter the colors in an image and more generally its appearance, the tone mapping category of interest is that of **SRP** operators.

### 2.2.2 Traditional Approach

In the theoretical framework illustrated so far, it can therefore be argued that the tone mapping operation is nothing more than the application of a mathematical function $f$ that maps the content of HDR images to the reference display. This function $f$ concentrates on surpassing the constraints of the output medium, striving to attain the optimal representation within the restricted gamut and dynamic range.
Traditional tone mapping techniques often involve a process of extracting the luminance information from an RGB image, applying a tone mapping operator to the luminance, and then spreading the modification back to the color channels. To better illustrate the process with an example, a very popular TMO proposed by Reinhard et al.[22], is used below.

Let $\mathbf{I}_{\mathrm{RGB}}(x, y)$ be the original RGB image, where $x$ and $y$ are pixel coordinates. As mentioned before, the first step involves extracting the luminance, $L(x, y)$, from the RGB image. One common method is to use the following formula[1]:

$$L(x, y) = 0.2126 \cdot R(x, y) + 0.7152 \cdot G(x, y) + 0.0722 \cdot B(x, y), \tag{2.6}$$

where $R(x, y)$, $G(x, y)$, and $B(x, y)$ are the red, green, and blue channels of the original image, respectively.
Next, the TMO (*i.e.* the aforementioned function $f$ ) is applied to the luminance. In the case of the Reinhard operator, the mapping is defined as follows:

$$L_{\mathrm{mapped}}(x, y) = \frac{L(x, y) \cdot (1 + \frac{L(x,y)}{L_{\mathrm{white}}^2})}{1 + L(x, y)}, \tag{2.7}$$

where $L_{\mathrm{white}}$ is a parameter representing the luminance of the brightest white in the scene.
Finally, the modified luminance is used to adjust each color channel:

[1]This way of computing the luminance holds when the RGB image is encoded using Rec.709 standard [23]

$$R_{\text{mapped}}(x,y) = \frac{R(x,y)}{L(x,y)} \cdot L_{\text{mapped}}(x,y), \tag{2.8}$$

$$G_{\text{mapped}}(x,y) = \frac{G(x,y)}{L(x,y)} \cdot L_{\text{mapped}}(x,y), \tag{2.9}$$

$$B_{\text{mapped}}(x,y) = \frac{B(x,y)}{L(x,y)} \cdot L_{\text{mapped}}(x,y). \tag{2.10}$$

The tone-mapped RGB image is therefore given by:

$$\mathbf{I}_{\text{mapped}}(x,y) = \begin{bmatrix} R_{\text{mapped}}(x,y) \\ G_{\text{mapped}}(x,y) \\ B_{\text{mapped}}(x,y) \end{bmatrix}. \tag{2.11}$$

### 2.2.3 Limitations

The problem with this way of performing the tone mapping operation is that it might introduce some color distortion. This phenomenon is due to the fact that luminance and chrominance are not always well separated. In this sense, when the luminance values are altered, implicitly color information is changing as well.

**Figure 2.6:** Color distortion across different shots.

Figure 2.6 shows three images captured by the same smartphone camera but with different field of view and angle of shot. It can be easily seen how the red color of the seats changes in each of the three shots, ending up being particularly saturated in the most right example.

Having become aware of the problem that arises in this methodology, the following chapters of this project will focus on the proposal of an alternative method and the search for an effective possibility of obtaining qualitatively better results.

# Chapter 3

# Intrinsic Decomposition Step

## 3.1 Definition and Applications

Images are produced through a sophisticated interplay between light and matter, culminating in a blend of colors, tones, and shadows. The reversed process of identifying and separating the constructive components of an image, given itself, is known as *Intrinsic Image Decomposition* (IID).
IID is a fundamental concept in the field of computer vision and graphics and it involves the factorization of an image into two primary components: *reflectance* (or *albedo*) and *shading* (or *lighting*).

$$I = R \odot S \tag{3.1}$$

Equation 3.1 describes the intrinsic model, which aims to represent the original image $I$ as a pixel-wise multiplication between $R$ and $S$, respectively the *reflectance* and the *shading* component. On the one hand, *reflectance* delivers information about the shape, the texture and the inherent color of an object, discarding any kind of lighting effects. On the other hand, *shading* captures the effects of light interacting with the object's geometry, resulting in shadows and highlights.

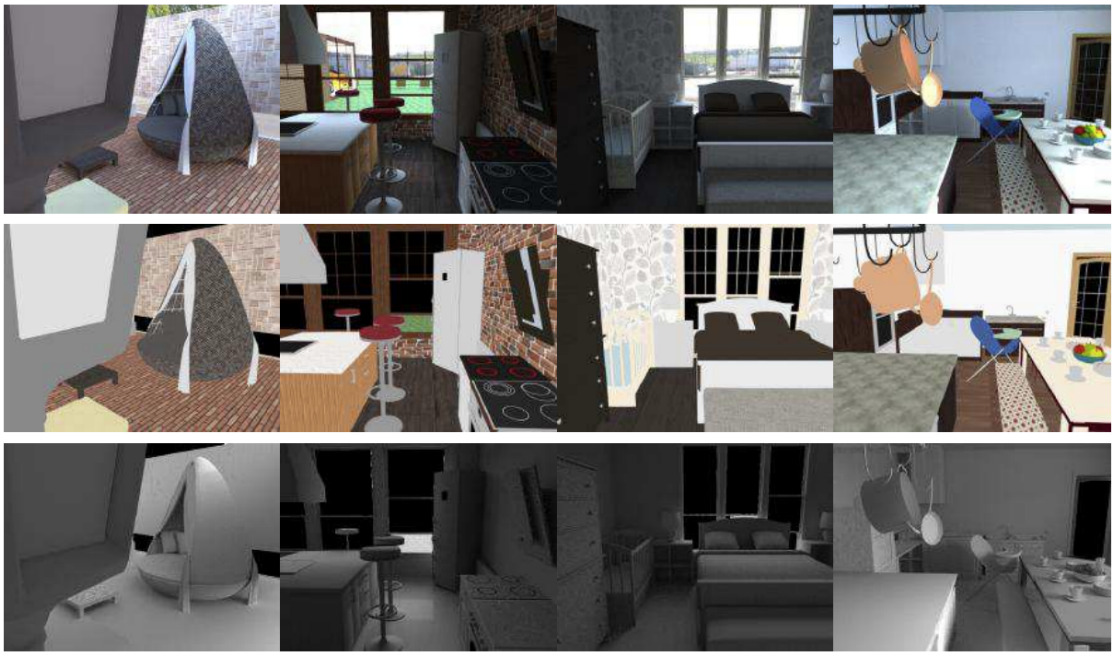

**Figure 3.1:** Decomposition example. On the top row original images, on the middle and bottom rows, respectively, reflectance and gray-scale shading. From [24].

This decomposition process provides a simplified yet insightful understanding of how images are formed, making it a valuable tool in various applications.
Since the intrinsic model should distinguish between properties of a scene that are unaffected by illumination and those that are dependent on lighting, it becomes crucial in numerous computational photography processes such as altering lighting conditions (*i.e.* relighting), changing colors (*i.e.* recoloring), and combining visual elements from various sources (*i.e.* compositing). Furthermore, this method is instrumental in tasks like object recognition and scenes understanding as, by isolating the effects of material properties and lighting conditions, it allows for more robust and versatile image analysis and manipulation.
However, the effectiveness of this method is often subjected to the satisfaction of the *Lambertian* assumption. What happens in reality is much more complex than what the intrinsic model seen so far tries to describe. As a matter of fact, even a straightforward environment can reveal numerous optical interactions. These can include indirect lighting, internal scattering within translucent objects, caustics, anisotropic and glossy reflections, among others. Taking all these factors into consideration when formulating an intrinsic model can overly complicate the problem. Instead, what is generally done is to consider a simplification of it, based on the *Lambertian* assumption. This hypothesis puts a constraint on how light

interacts with a surface. According to this model, the surface appears equally bright from all viewing directions. In other words, the surface is assumed to be perfectly diffusing, meaning it scatters incident light uniformly in all directions. The main limitation is that this assumption is not always valid in real-world images, as many surfaces are not perfectly diffusing and may exhibit properties such as specularity.

## 3.2 IID and Tone Mapping

There is convincing evidence in the literature that numerous processes in our visual system are designed to minimize the impact of light. For example, the human visual system has the ability to diminish the effect of lighting and approximately preserve the appearance of an object by making its color stable over a wide range of possible lighting. If, in this sense, light appears to be of lesser significance, it's probable that alterations to the shading component will be less noticeable than changes to the reflectance one. Limiting modifications only to the light component could be particularly beneficial in the context of tone mapping, as light is largely accountable for the extensive dynamic range in real-world settings. Indeed the maximum dynamic range generated solely by reflectance is less than two orders of magnitude. Instead, the dynamic range of the shading component can easily surpass four orders of magnitude, as reported in Mantiuk et al. [21]. This reasoning legitimizes the idea of carrying out the tone mapping operation *only* on the shading component obtained thanks to the decomposition model. The final tone mapped image is then the result of multiplying the reflectance component, which has remained unchanged, with the shading component, which has instead undergone the necessary modifications. Figure 3.2 outlines the process that this work intends to pursue: building a sort of framework that is able to perform, in chain, first the decomposition of an image and then the tone mapping operation, by exploiting the separation of the newly obtained intrinsic components. The objective is therefore to compare the result obtained by following this method with that which would be obtained if the same TMO were applied directly to the original image and without going through the decomposition step. It is important to underline that it is not wanted to ensure that the two results are almost identical but rather that, through decomposition, there are noticeable (and good) differences in terms of color distortion and overall picture quality.

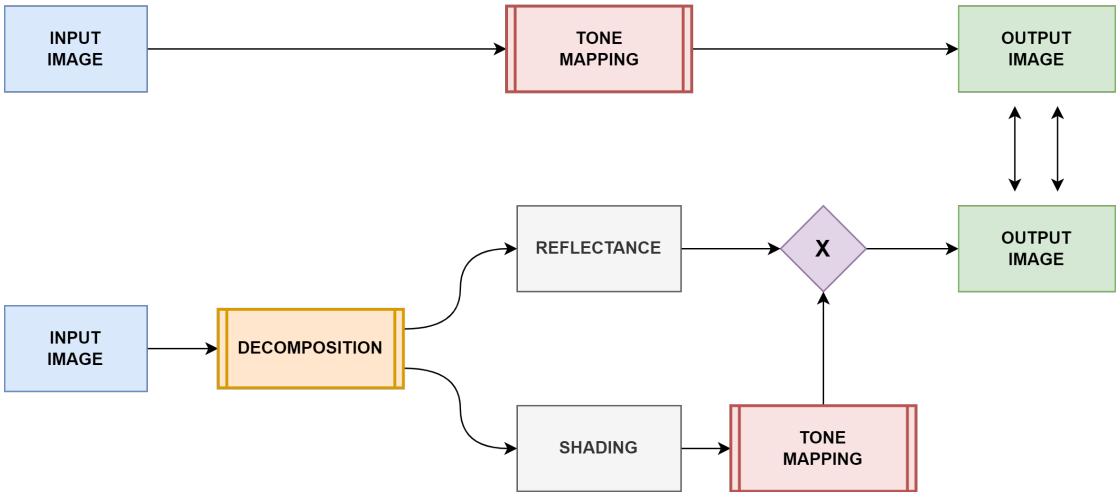


**Figure 3.2:** Project flowchart.

## 3.3 Employed Methods

Although the intrinsic model seems to require satisfying a simple equation like the one described in 3.1, the problem of image decomposition is an ill-defined problem. As a matter of fact, three different terms are involved into the equation, of which only one is known, namely the original image $I$. The task of identifying the value of the other two variables is complex because it is highly under-constrained – *i.e.* considered a certain value for $R$ and $S$ whose product reconstruct $I$, also $k \cdot R$ and $\frac{1}{k} \cdot S$ satisfy the equation for all the infinite possible values of $k > 0$.
Having taken into consideration the innate problem of intrinsic decomposition, the determination of an image splitting that is as optimal as possible is generally addressed with two different approaches: traditional methods and AI-based methods.

### 3.3.1 Optimization based Algorithms

Traditional methods for intrinsic images rely on optimization-based techniques. These methods incorporate various assumptions - *i.e.* priors - about the world, in the attempt to constrain the problem and reduce the solution space. For example, possible constraints explored by previous works are piece-wise constancy, parsimony of reflectance, shading smoothness (etc.). However, these manually designed assumptions often fall short when applied to real-world images, as they are challenging to formulate and frequently breached.

To enhance the quality of decomposition, recent techniques have attempted to incorporate surface normals or depth information from RGB-D cameras into the optimization process. Nevertheless, these techniques presuppose the availability of depth maps during the optimization, which limits their applicability to a broad spectrum of everyday photographs.

### 3.3.2 Learning based Algorithms

Recently, learning methods have been investigated as an alternative to traditional models with manually designed priors for intrinsic images, or as a means to automatically set the parameters of these models. Since the data taken into consideration are images, Convolutional Neural Networks (CNNs) become a very powerful tool and essentially the first choice in addressing the problem. With this approach, the task is usually considered as a regression one and it is supervised (*i.e.,* the are labels associated with the images). More precisely, the problem is defined as a multi output learning task since, given an input image, the objective is to get two different output images. In these circumstances, the architectures typically chosen involve one shared encoder and two distinct decoders. The encoder will be responsible for carrying out feature extraction, providing a more compact and meaningful version of the input image as an intermediate output, while the two separate decoders will be responsible for reconstructing two separate images starting from the latent space generated by the encoder. To retain intricate spatial details, models are often augmented by incorporating skip connections between corresponding layers in both the encoder and decoder networks.

#### Popular IID training datasets

Building a model capable of directly regress to the output decomposition requires using various training datasets. Within this scope, some popular datasets are *MPI Sintel* [25], *MIT Intrinsics* [26], and *ShapeNet* [27]. Two other very famous and particularly used datasets are *Intrinsic Images in the Wild (IIW)* [28] and *Shading Annotations in the Wild (SAW)* [29]. Both of them are based on indoor real-world scene images but they consist of sparse, crowd-sourced reflectance and shading annotations. Following this, several papers have trained CNN-based classifiers on these sparse annotations and used the classifier outputs as priors for guiding decomposition.
However, all the above described datasets alone are found to be inadequate for training a direct regression approach due the fact that they either contain only sparse annotations or just few images with a sophisticated structure and lighting which allow the model to learn to behave even in more complex examples, such as real ones.

| Dataset | Size | Setting | Nature | Illumination | GT type |
|---|---|---|---|---|---|
| MPI Sintel | 890 | Animation | non-PB Rendered | Spatial-varying | Full |
| MIT Intrinsics | 110 | Object | Real | Single Global | Full |
| ShapeNet | 2M+ | Object | PB Rendered | Single Global | Full |
| IIW | 5200+ | Scene | Real | Spatial-varying | Sparse |
| SAW | 6600+ | Scene | Real | Spatial-varying | Sparse |
| CGIntrinsics | 20,000+ | Scene | PB Rendered | Spatial-varying | Full |
| OpenRooms | 118,000+ | Scene | PB Rendered | Spatial-varying | Full |
| Hypersim | 77,400 | Scene | PB Rendered | Spatial-varying | Full |

**Table 3.1:** IID datasets comparison. PB stands for *physically based*. Adapted from [24]

Since it is very difficult to obtain images with ground truth full data associated, the most used approach is to exploit synthetically generated data to train CNN-based models. In this direction, a very important work was conducted by Li et al. [24], who created *CGIntrinsics*, the first large-scale scene-level intrinsic images dataset based on high-quality physically-based rendering with over 20,000 images of indoor scenes. Another massive work from Li et al. [30] led to the generation of a new dataset called *OpenRooms* with broad applicability across computer vision, graphics and robotics. It contains 118,233 HDR images from 1,287 different scenes with associated ground truth geometry, material, lighting and semantics. Finally, a very similar dataset recently released is *Hypersim* [31], a photorealistic synthetic collection of images for holistic indoor scene understanding. To create this dataset, a large repository of synthetic scenes created by professional artists was leveraged and 77,400 HDR images of 461 indoor scenes with detailed per-pixel labels and corresponding ground truth geometry were generated.
These last three datasets have significantly fueled the adoption of deep learning solutions for the IID problem as they provide high-quality images, leading to substantial enhancements in the generalization of CNNs trained on synthetic data to real-world scenarios.

### Related Works

*CGIntrinsics*
In addition to generating a synthetic image dataset for training deep learning models, Li et al. [24] have proposed in their work a partially supervised learning

method for training a CNN to directly predict reflectance and shading, by combining ground truth from *CGIntrinsics* and sparse annotations from *IIW* and *SAW*. The skeleton architecture is based on a variant of *CycleGAN* [32] which is a network that performs *unpaired* image-to-image translation by moving from a specific style or semantic domain to another one. However, in this work, a form of *paired* image-to-image translation task is considered, as it suits best the overarching goal. For each input image, there are specific corresponding *reference* images to take into account when performing the intrinsic decomposition of an image in the way that best approximate those ground truths.

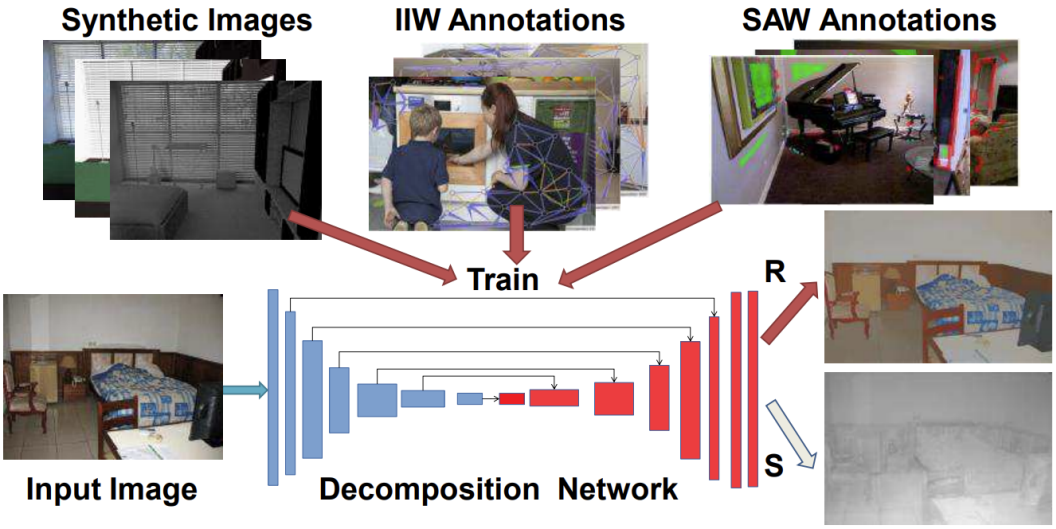


**Figure 3.3:** CGIntrinsics network architecture. From [24].

*PIE-Net*

Previous approaches to IID primarily concentrated on the exploration of manually crafted priors as a means to narrow down the solution space. These methods use explicit priors to constraint the problem by formulating assumptions, sometimes hard, about the world. However, this may highly restrict the usefulness of these techniques. Deep learning methods, instead, have been developed based on implicit restrictions as defined by the losses. But, these methods are entirely data-driven and this may still limit the possibility of making them able to generalise well in different scenarios. In addition, they can be negatively influenced by strong illumination conditions causing the well-know problem of *shading-reflectance leakages*. As a matter of fact, it often happens that, in the presence of strong shadows, the network might misinterpret shading transitions as reflectance transitions. This occurs when the gradient assumption, which wants that shading changes correspond to soft gradients and reflectance transitions to hard ones, is breached. A good way to

go is to merge these two different approaches using edge-driven hybrid networks. *PIE-Net*, from Das et al. [33], represents an intrinsic CNN-based model whose basic idea is to use illumination invariant gradients which only depend on reflectance changes. Furthermore, by using a hierarchical CNN approach, with global and local layers, this work aims at solving the aforementioned problem of *shading/reflectance* misclassifications.

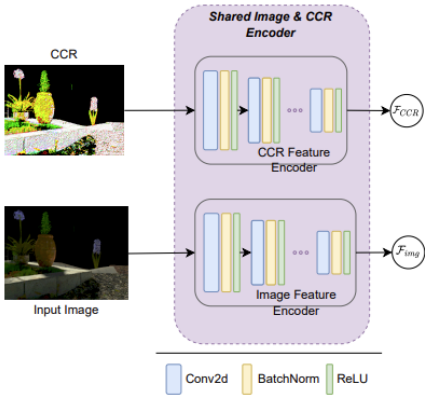


**Figure 3.4:** Shared Image and CCR Encoder. From [33].

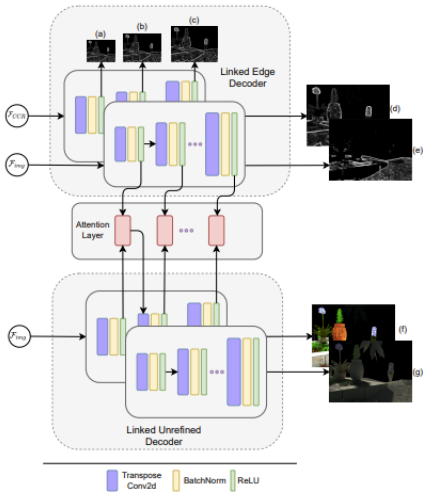


**Figure 3.5:** Linked and Unrefined Edge Decoder. From [33].

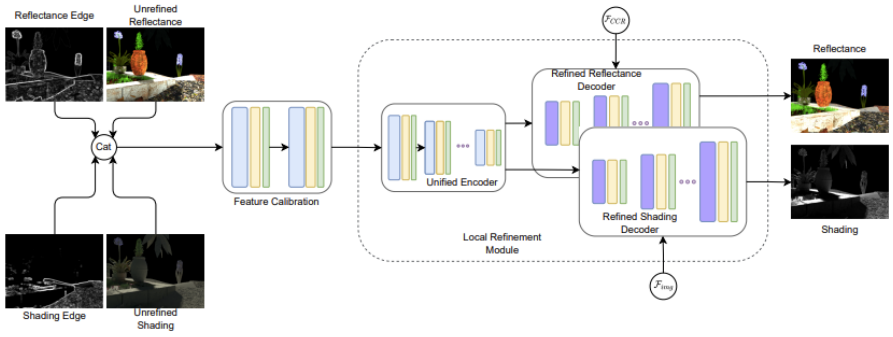


**Figure 3.6:** Local Refinement Module. From [33].

The distinctive character of this work lies in the use of gradients based on illumination invariant descriptors, *Cross Color Ratios (CCR)*(Figure 3.7). For an RGB image and two adjacent pixels, denoted as $p1$ and $p2$, the Cross Color Ratios

(CCRs) are defined as follows:

$$[MRG = \frac{R_{p1}G_{p2}}{R_{p2}G_{p1}}, \quad MRB = \frac{R_{p1}B_{p2}}{R_{p2}B_{p1}}, \quad MGB = \frac{G_{p1}B_{p2}}{G_{p2}B_{p1}},] \tag{3.2}$$

where $MRG$, $MRB$, and $MGB$ represent the CCRs for the (R,G), (R,B), and (G,B) channel pairs, respectively.
As shown in Figure 3.4, the original input image and its corresponding CCR image undergo encoding via distinct encoders.

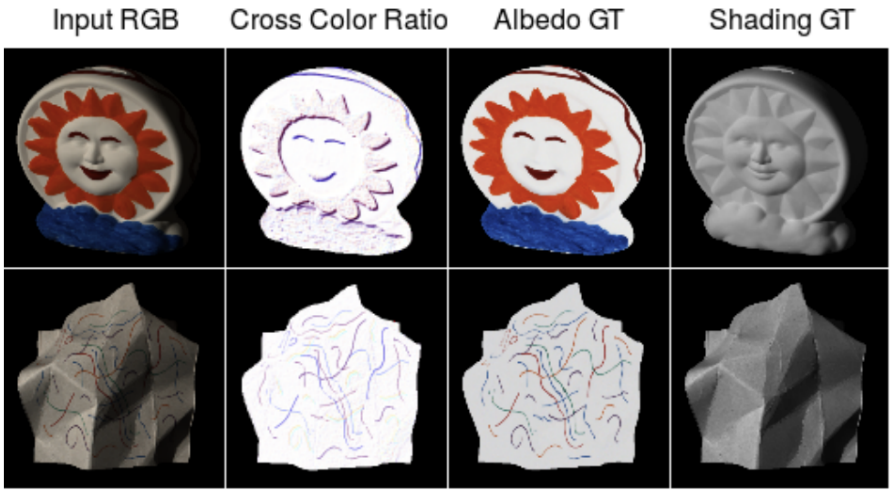


**Figure 3.7:** CCR as illumination invariant descriptors. They change only when reflectance changes. From [33].

This enables the CCR Encoder to grasp features related to invariant reflectance transition irrespective of shading variations. Simultaneously, the image encoder acquires a fused feature encompassing both shading and reflectance cues, autonomously. These encoded representations are subsequently employed in the later stages of the architecture, facilitating separate utilization of features for both global and local layers. Consequently, the Linked Edge Decoder (LED) can learn both reflectance and shading edges jointly and the output from each block of the LED is then fed through an attention layer before being convolved through the respective block in the unrefined decoder (Figure 3.5). As intrinsic components exhibit spatial dependence, spatial attention layers have been incorporated allowing the network to focus on specific image areas, where it is challenging performing a correct decomposition. Finally, Figure 3.6 shows that the edge maps of reflectance and shading, along with the unrefined reflectance and shading pairs, are combined and subjected to convolution through a feature calibration layer for then undergoing processing through an encoder-decoder to yield the ultimate outputs.

*IID via Ordinal Shading*

In the continuous study of the IID problem, a very recent work from Careaga et al. [34] seems to have outlined an AI-based architecture that achieves good results in a wide range of possible images (e.g. indoor, outdoor, with human faces etc..). In the official paper, the authors claim that they successfully accomplish the task of high-resolution intrinsic decomposition by dividing the problem into two segments. Initially, they introduce a dense ordinal shading method that utilizes a shift-and scale-invariant loss to predict ordinal shading cues, without confining the predictions to adhere precisely to the intrinsic model equation (3.1). Subsequently, they merge the low- and high-resolution ordinal predictions using another network to produce a shading estimate that maintains both global consistency and local nuances. In the first of the two steps, the idea is to create two ordinal shading estimates: one at the receptive field resolution of the network, and another at a significantly higher resolution. The low-resolution estimate offers globally coherent ordinal constraints, but it lacks high-resolution details. Conversely, the high-resolution estimate includes detailed shading discontinuities, providing dependable local constraints. These two ordinal estimates are then used as input to the second network, along with the original input image. With both global and local constraints readily available to the network, it is possible to generate a globally consistent shading with high-resolution details and sharp shading discontinuities. Finally, the corresponding reflectance is calculated using the input image and the estimated shading, based on the intrinsic equation.

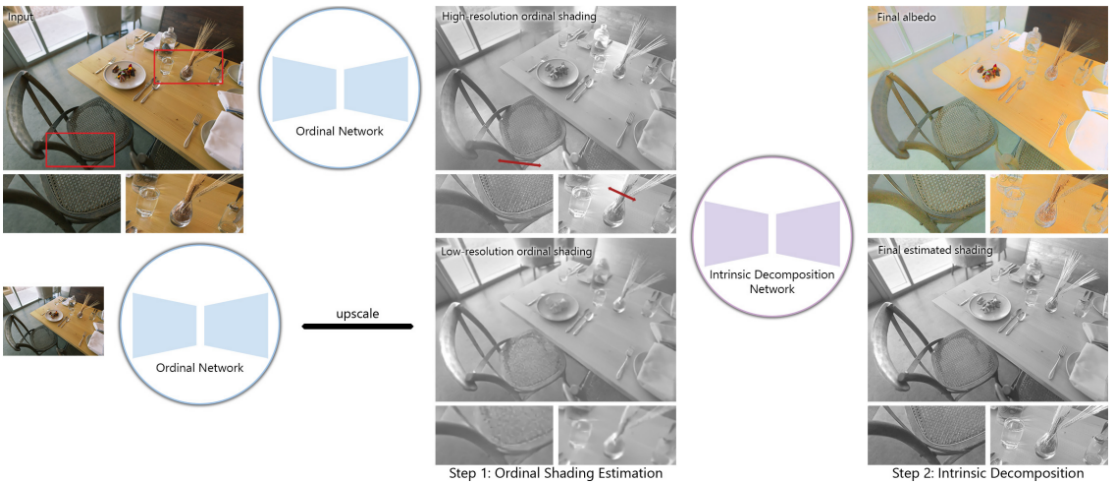


**Figure 3.8:** IID via Ordinal Shading overview. From [34].

Figure 3.8 summarizes what has been said so far. The proposed architecture is made up of two separate parts which, when put together, produce the final decomposition.

## 3.4 Testing Available IID Models

Of the models described above, particular attention has been paid to the last two in the course of this work. In fact, in addition to being quite recent models, *PIE-Net* [33] and *IID via Ordinal Shading* [34] promise to obtain good decomposition results in terms of general perceived quality. Furthermore, for both architectures, the authors have respectively released an already pre-trained version of their models, making it possible to test their capabilities with any type of custom image. As the study revolves around HDR images, the goal is to test the aforementioned models using these kind of images as the primary dataset. However, to evaluate the quality of the results, it would be very convenient to initially have images that act as a reference for the decomposition. In other words, for a given image that is sent as input to the IID model, the ground truth shading and reflectance associated with that image must be possessed. In this way, the evaluation of the predictive power of each model and of the images produced becomes quite immediate since a simple visual comparison can be made.

### 3.4.1 Dataset: Hypersim

As already mentioned in the previous paragraphs, *Hypersim* [31] represents a photorealistic synthetic dataset of HDR images, for which the entire ground truth decomposition is available. This therefore implies that it is a dataset that lends itself very well to the type of experiment that it was decided to conduct. These images are part of a collection featuring various indoor scenes, each possessing a resolution of 768 x 1024 and encoded across three color channels. However, in addition to the now well-known intrinsic components reflectance and shading, there is also a new term associated with each image, called *residual*, that captures view-dependent lighting effects. All these three variables adhere, exhibiting negligible error, to the following equation:

$$I = R \odot S + C \tag{3.3}$$

$I$, $R$, $S$, and $C$ denote the input image, the reflectance component, the shading component, and the non-diffuse residual term, respectively. In summary, the decomposition proposed in this dataset is therefore formulated in a slightly different way from that seen so far.

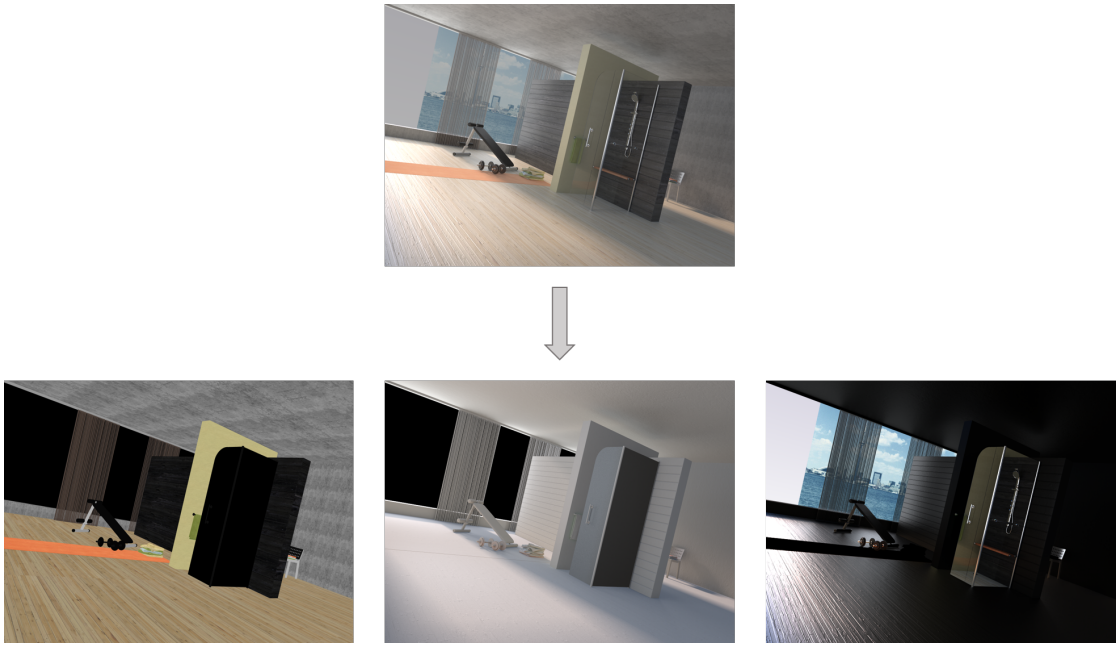

**Figure 3.9:** Hypersim full ground truth decomposition. At the top, the original image. At the bottom, from left to right, the reflectance, the shading and the residual ground truth components.

Figure 3.9 depicts the complete ground truth decomposition of a given color image. It can be observed how crucial information is embedded in the residual term: in addition to completely capturing the content of certain regions of the image (*e.g.*, windows), it also illustrates how objects interact with light.

## 3.5 PIE-Net Predictions

### Input Normalization

IID architectures generally require the input to be a LDR image, as its content is more easily manageable. In this sense, a typical operation that is performed before testing the model is to apply a sort of encoding to the HDR image that remaps its content to the point of making it an LDR image. In the specific case of the *Hypersim* images, the same authors of the dataset have released an image encoding algorithm, which they found out to be suitable for reproducing more natural-looking images.
Given a linear HDR radiance image $I_{\text{HDR}}$, the procedure involves first identifying the *90th* percentile intensity value, denoted as $p_{90}$, which represents the pixel value of the image such that the 90% of the remaining pixel values are less than or

equal to that value. Following this, the image $I_{\text{LDR}}$ is then derived through the transformation:

$$I_{\text{LDR}} = \alpha \cdot (I_{\text{HDR}})^{\gamma} \tag{3.4}$$

where $\gamma$ is the standard gamma correction factor of $1/2.2$, and the parameter $\alpha$ is computed to ensure that the value corresponding to $p_{90}$ maps to 0.8. The resultant image is finally constrained within the range $[0, 1]$. This mapping approach ensures that, at most, 10% of the image pixels become saturated afterwards, producing LDR images with a visually natural quality.

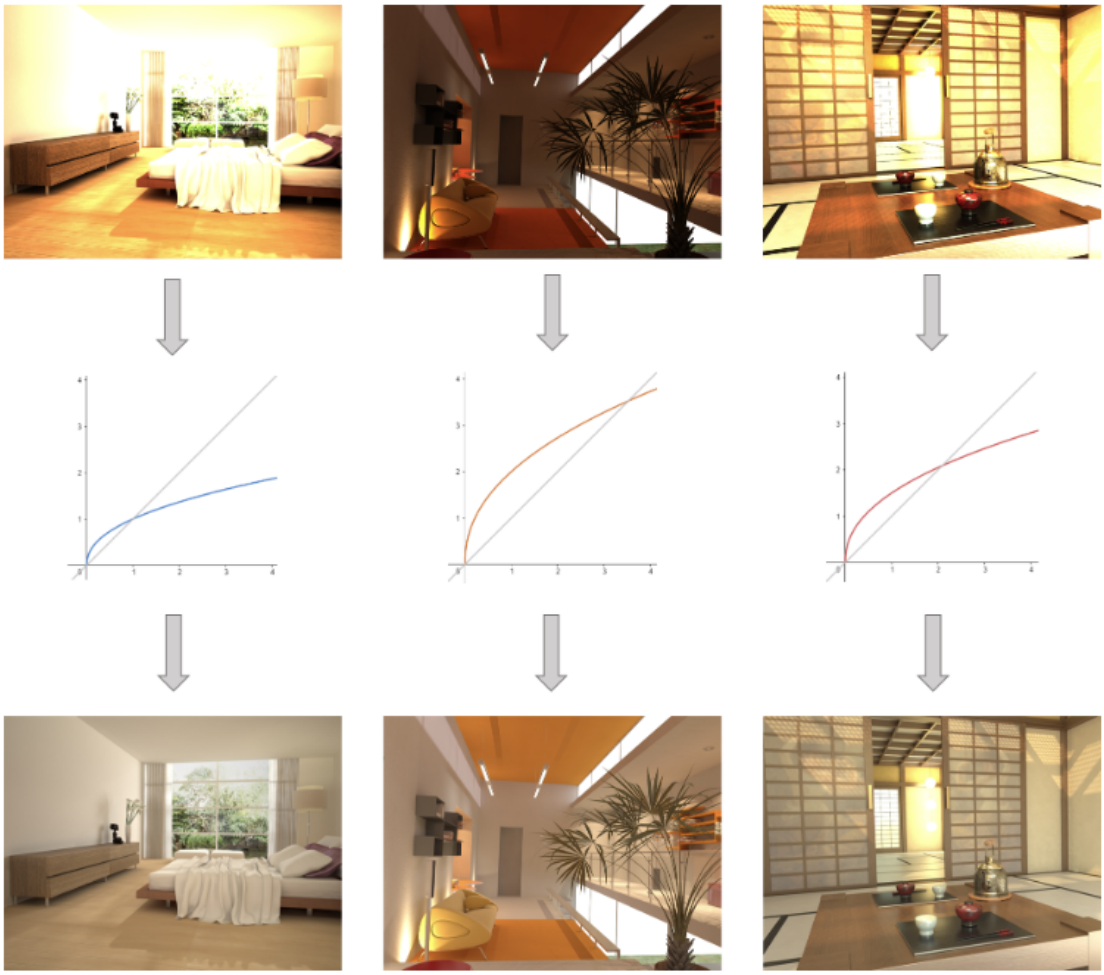

**Figure 3.10:** Hypersim preliminary normalization. Images at the top row represent the HDR images without normalization (clipped for visualization). Images at the bottom are the results after normalization.

What is done to the images is exactly a tone mapping operation: a sort of gamma correction is applied in order to map the linear (HDR) values in the [0, 1] interval, as described in Figure 3.10. Each image undergoes modification by means of a specific curve for each of it as it depends on the $\alpha$ parameter, which in turn is recovered from the image itself via the percentile method.

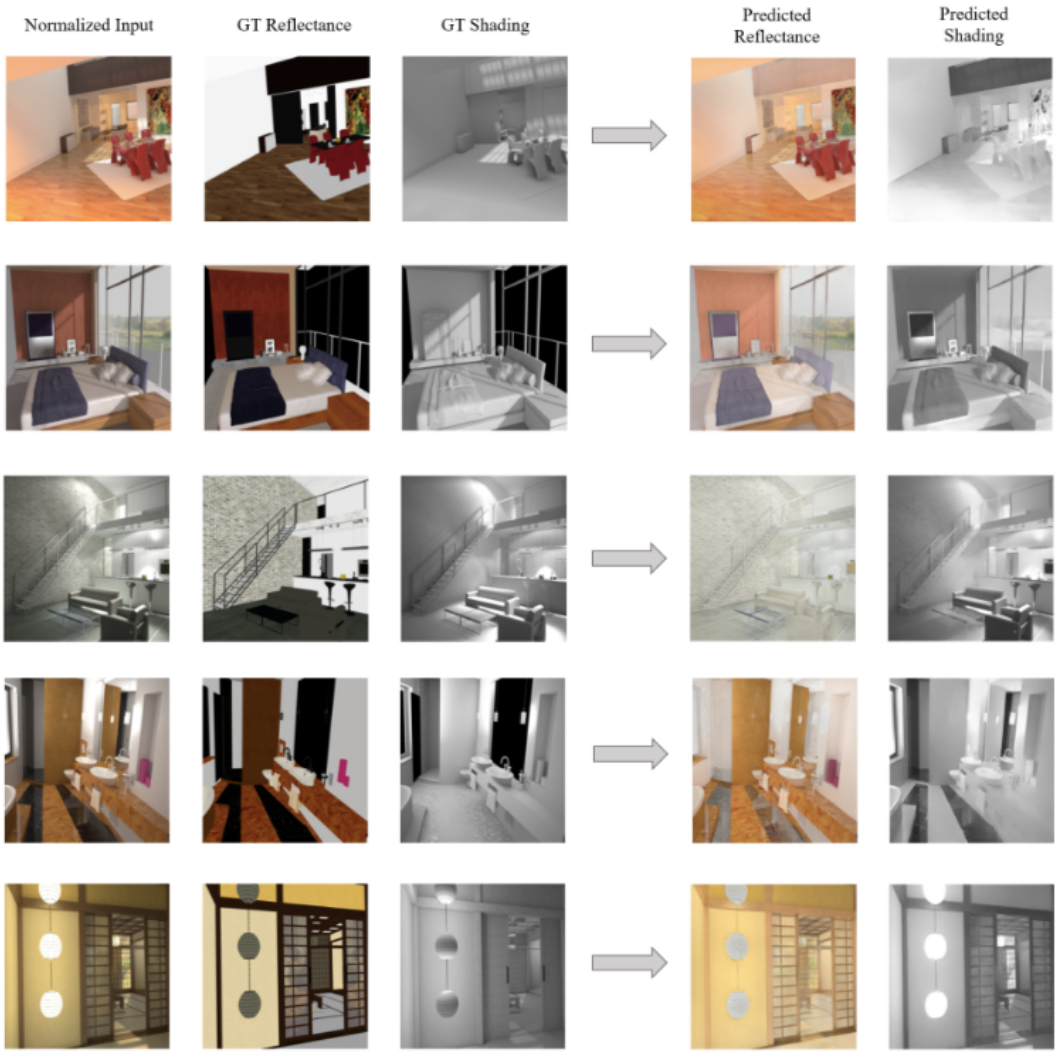


**Figure 3.11:** Pre-trained model predictions.

#### Results of the Pre-Trained Model

With the normalized images, the decomposition results can be finally obtained and shown in Figure 3.11. It can be argued that the model does not perform very well as the predictions it makes look distant from the available ground truths. Reflectance components should show the object's colors as they are, without any effects from lighting. However, many reflections and specular highlights are not removed from them. Furthermore, the shading components appear to be flat and sometimes fail to faithfully recreate the original content of the images. The ineffectiveness of the results can be generously justified by the fact that a simple pre-trained model was used and it has been tested on images belonging to a domain perhaps different from the one on which it was trained. As a matter of fact, this model has been trained on *Natural Environment Dataset (NED)* [35], a collection of real outdoor images representing gardens, whereas *Hypersim* contains only synthetic indoor scenes. Taking into account what has been said, we believed that, by downstreaming a new learning process using part of *Hypersim*, the model could refine its ability to generate more accurate predictions.

### 3.5.1 Fine-tuning on Hypersim

All the reasoning just made boils down to performing the fine-tuning of the pre-trained network on *Hypersim*. This technique allows to build upon the patterns and structures already learnt from other datasets, reducing the time and computational resources required for training. This is also particularly beneficial when dealing with smaller datasets, as it mitigates the risk of overfitting. In fact, the idea is to consider only a small part of the totality of images present in *Hypersim* in order to fine-tune the model. To be able to arrange a dataset of images from *Hypersim* with the aim of starting a learning process, for each input image entry there should be associated two images, representing its decomposition ground truths.

| **Input Image** | Ground Truth Reflectance | Ground Truth Shading |
|---|---|---|

**Table 3.2:** Ideal dataset entry example.

#### Residual Term Handling

Recalling the *Hypersim* structure, each image is accompanied by a ground truth reflectance, a ground truth shading and a ground truth residual. However, in configuring the dataset on which to fine-tune the pretrained model, as described

in Table 3.2, the residual term is voluntarily neglected. This choice is made to simplify the process and also by virtue of the fact that *PIE-Net*, in its original architecture, is able to generate only two output images and not three. Apart from this, a problem still emerges with this setting. In fact, the model is trained to split an image into two components such that their product exactly returns the original input. However, considering the product between the ground truth reflectance and the ground truth shading, it never reconstructs the original image since crucial information is missing because contained in the additive residual term. When the model performs inference over an image, the predictions it makes contain within them the residual term, somehow embedded. In this sense, when comparing each of them with the respective ground truths, there will always be a gap. Looking at Figure 3.11, it is possible to notice that in the ground truth decomposition components there is less information than there is in the predicted ones. Some regions are totally masked because their contents are supposed to be owned by the residual term. The problem lies in the fact that the model does not know this, so it will always produce results that are never close to the ground truths that are available. In this sense, neglecting the residual term can lead to significant losses, therefore its management must be more intelligent. One possibility is to incorporate its content into the reflectance and shading ground truths. Specifically, the idea is to move from the original expression of the input image, as described in 3.1, to the following one:

$$I = R' \odot S' \tag{3.5}$$

where $R' = f(R, C)$ and $S' = g(S, C)$. The obstacle, however, lies precisely in identifying these two functions $f$ and $g$ that can satisfy the equation. The transformation to be applied is not simple because the residual term is additive and not multiplicative.
The other option suggests simply modifying the original image by subtracting its residual content. In mathematical formulation:

$$I' = R \odot S \tag{3.6}$$

where $I' = I - C$. In this way, it is directly the content of the original image that is changed. Since finding a solution to the embedding of the residual term in both intrinsic components turns out to be an under-constrained problem, we decided to opt for the more immediate solution of changing the content of the image by subtracting the residual term (3.6).

### Dataset Pre-Processing

Before starting a learning task, images need to be pre-processed to make them cleaner and more suitable for this purpose. First of all, we selected only a specific subset of HDR images with complete ground truth labels associated. The dimensions of these images are then altered from their original size of 768 x 1024 pixels to a more manageable *squared* size of 124 x 124 or 256 x 256 pixels. Furthermore, we identified and removed all completely black images. The dataset is further refined by addressing infinite values, which we treated as *fireflies* and handled using a more elegant version of a median filter that remove these pixels and replace them with pixels in the neighborhood rather than the maximum, which might not be in that area. Lastly, any value in the images that are not a number (NaN) are replaced with the mean value of the images, ensuring a more uniform and clean dataset. In addition to the previous steps, we further reduced the dimension of the dataset to 600 images. In fact, since the objective is to only perform fine-tuning, there is no need of a large amount of data to train the model. These images were then split into two distinct sets: 500 images were allocated for the training process. The remaining 100 images were set aside for validation. This process of dividing the dataset enhances the model's ability to generalize its learning to new, unseen data. In addition, the division was carried out by establishing a fixed random seed, which guarantees that all experiments could be reproduced consistently. Finally, we normalized both the original (input) images and the associated ground truth shading according to the mapping described in 3.4. In *Hypersim*, those two image have a dynamic range that exceeds the range [0, 1]. Instead, reflectance images are into this range. It is therefore the shading component that carries out the HDR information. Applying a preliminary normalization is particularly useful in scenarios where the HDR images are used for further processing or analysis, as it helps to ensure that the algorithms are not unduly influenced by the extreme brightness values that can be present this kind of images.

### Dataset Statistics

In order to ensure that the model is built on a solid understanding of the data, improving its performance and reliability, it is very important to compute some statistics about the dataset we want to train it on. In this direction, it becomes crucial, for example, to ensure that the training and validation sets are representative of the same underlying distribution. This principle of maintaining a representative distribution of relevant features or characteristics within different subsets of the data is known as *stratification*. Deep learning models learn to make predictions based on the patterns they identify in the training data. If the test data comes from a significantly different distribution, the model's performance may degrade, as it has not learned the patterns relevant to the test data. In this specific case,

it was deemed appropriate to focus on the computation of the distributions of mean luminance and the histograms of colors for both the training and test sets. By analyzing the luminance distributions, we can understand if images in the training set are generally darker than those in the test set (or vice versa); instead, the histograms help us understand which color components predominate in the images and whether there is an analogy between the two sets or not. In this way, we can identify some potential issues and take steps to address them to ensure a more balanced distribution so that the model does not perform poorly on one set compared to the other.

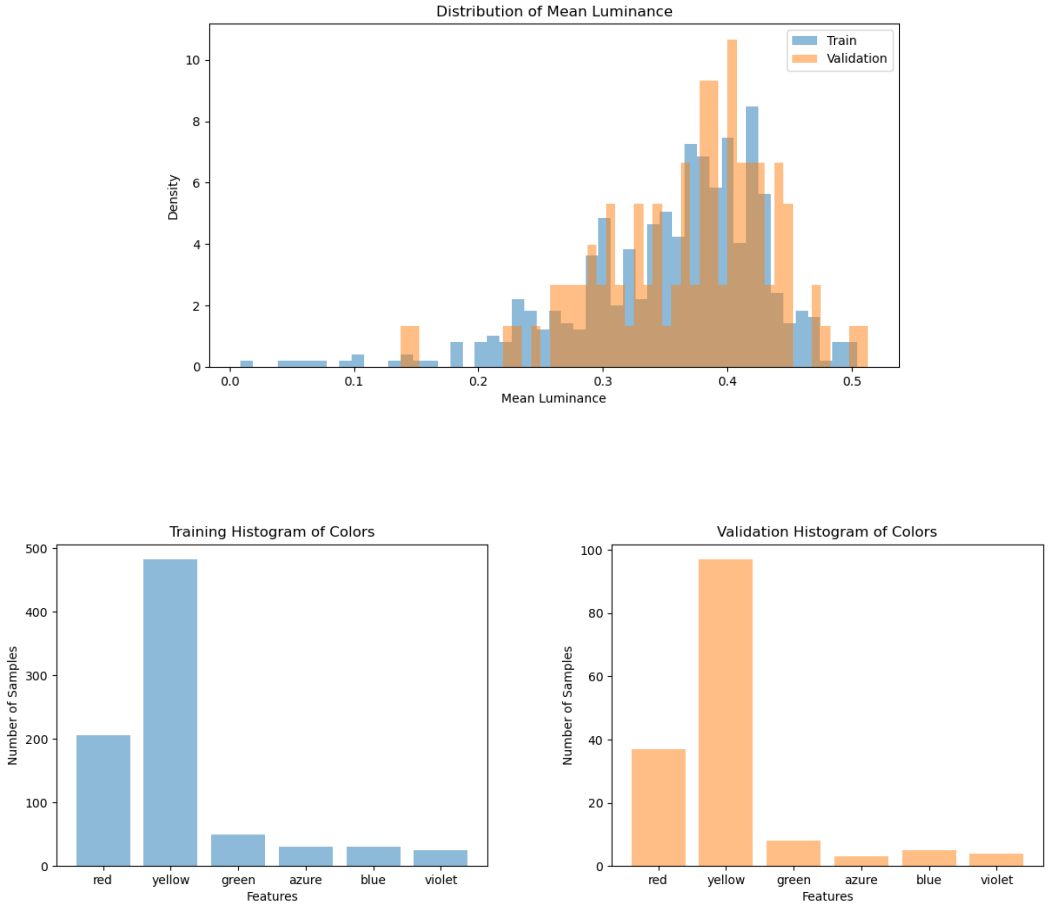


**Figure 3.12:** Dataset statistics. Luminance distributions and colorsEhistograms of training and validation set.

By inspecting the aforementioned distributions (Figure 3.12), it is easy to notice that, as regards the colors, these are distributed in the exact same way between the two sets meaning that no intervention is required. As regards luminance, it seems

that the validation set has a deficiency in the lower luminance values compared to the training set. Despite that, we decided against performing stratification as there doesn't seem to be a significant issue that could impact the predictive capabilities of the model.

### First Training

At this point we finally started a learning process, which was performed for 25 epochs, each of which took on average 45 minutes. Different configurations for fine-tuning were tested, exploring several techniques and hyperparameters values. Specifically, regarding the optimizer, we experimented *SGD* and *Adam*. Instead, as criterion for minimizing the error the model makes in outputting the predictions, we considered the *L1* loss and the *MSE*.
On the one hand, the *L1* metric calculates the absolute differences between corresponding pixel intensities in the predicted image ($P_\text{i}$) and the ground truth image ($G_\text{i}$).

$$L1 = \frac{1}{N}\sum_{i=1}^{N}|P_i - G_i| \tag{3.7}$$

This metric provides a measure of the average magnitude of these differences, emphasizing their absolute values.
On the other hand, the *MSE* metric, is expressed as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(P_i - G_i)^2 \tag{3.8}$$

It computes the mean of the squared differences between pixel intensities and it places greater emphasis on larger deviations, penalizing larger errors more significantly than *L1*.
Both metrics generate a single numerical value for the dissimilarity between the predicted and ground truth images, with lower values indicating a closer match. However, it is important to note that these metrics may not always align with human perceptual judgments as they can end up attributing an excellent loss value to a result that is not so good or vice versa. The challenges lie in the complexity of human visual perception, where factors such as texture and structure play crucial roles. Regardless of the chosen criterion, the loss function for a given image, denoted as $y$, comprises two terms, each of which is dedicated to evaluating the disparity between the predicted reflectance component ($\hat{R}_y$) and the corresponding ground

truth ($R_y$), as well as the predicted shading component ($\hat{S}_y$) and its corresponding ground truth ($S_y$).

$$\mathcal{L}(y) = criterion(R_y, \hat{R}_y) + criterion(S_y, \hat{S}_y) \tag{3.9}$$

Table 3.3 shows the values of the hyperparameters involved in fine-tuning, highlighting in yellow the combination that lead to better outcomes.

| **Batch size** | **Image size** | **Loss** | **Optimizer** | **Learning rate** |
|---|---|---|---|---|
| 2 | 124 | L1 | SGD | 0.005 |
| 4 | 256 | MSE | Adam | 0.01 |

**Table 3.3:** Fine-tuning setting for the first major attempt.

### Results of the First Training

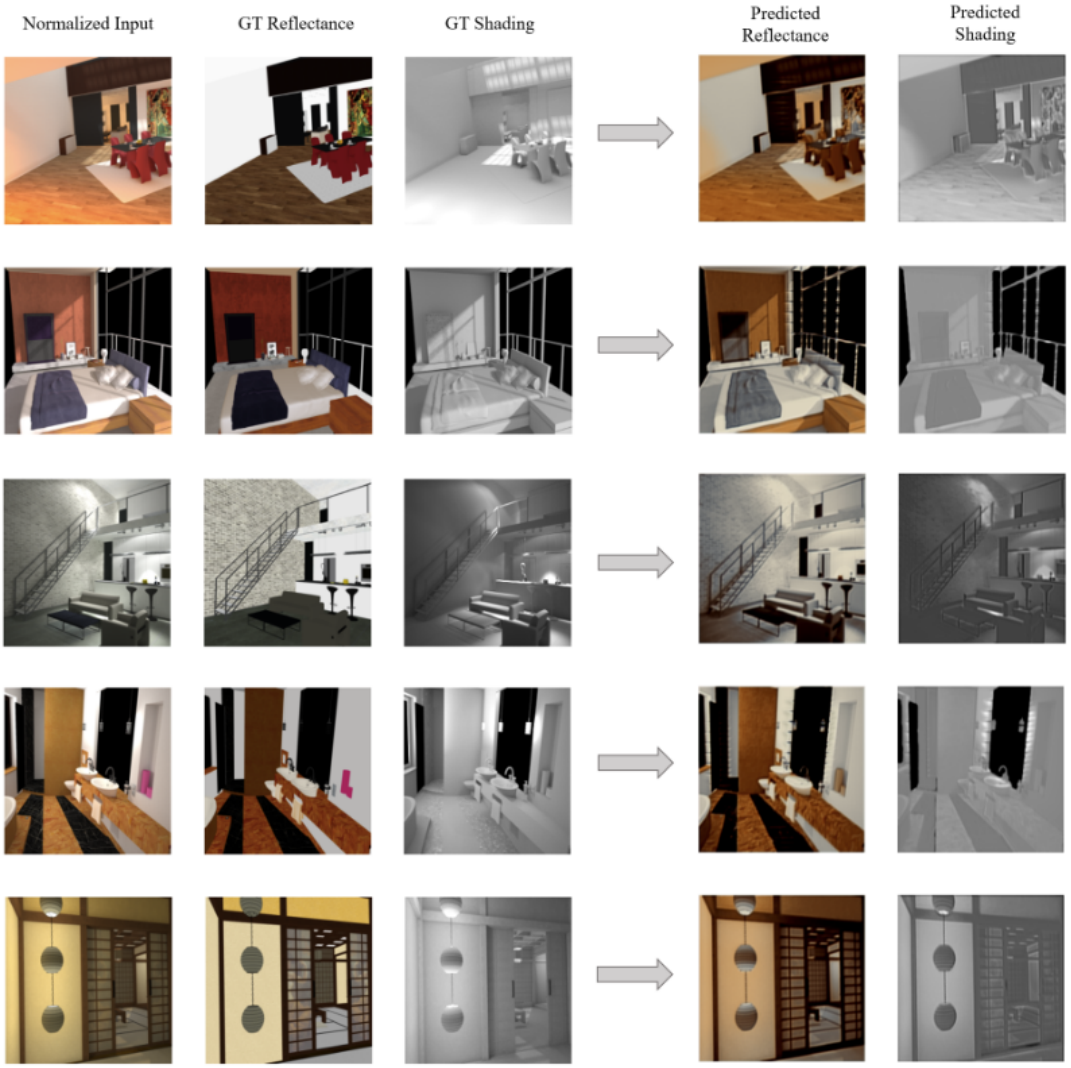


**Figure 3.13:** Results of the first fine-tuning.

In Figure 3.13, we can see how even after a fine-tuning process, the model produces images that are still distant from the target ones. The predicted images present distortions in the color information and the disentangling between the shading and the reflectance is in most cases defective.

### Model Inference Modification

The results obtained so far by both the pre-trained model and the fine-tuned one (according to the specifications just reported) are to be considered insufficient and far from what should be expected. Beyond the values that various metrics can give when comparing the predictions of the decomposition model with the ground truths, the judgment that matters most for now is the visual one and clearly the images generated are not accurate.
A plausible reason why the fine-tuning process does not work as hoped and does not lead to any real improvements is the fact that the network used is too complex for such simple fine-tuning. As a possible solution, it is considered to be worth trying to lighten the network by ensuring that, given the input image, it exclusively predicts the reflectance. Furthermore, the shading can be recovered simply dividing the input image by the predicted reflectance. In many intrinsic decomposition methods, the estimation of shading and reflcetance involves treating them as separate variables. However, this approach often leads to the inability of the estimated shading and reflectance to accurately reconstruct the original image. In contrast, the idea now is to compute the shading based on the estimated reflectance, ensuring adherence to the intrinsic model (3.1).

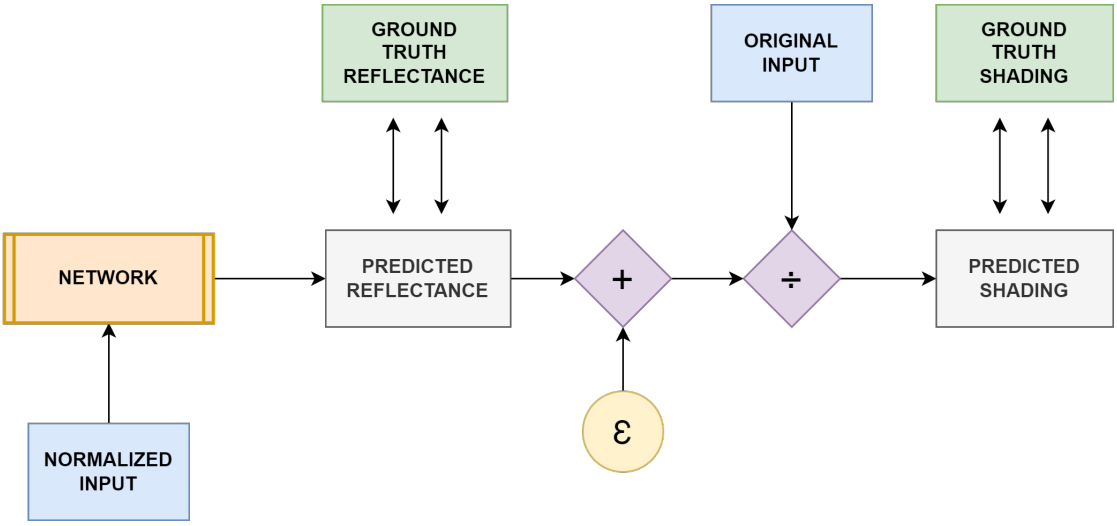


**Figure 3.14:** Changing the way the model makes inference. Only the reflectance is predicted. Original input is used to recover the shading.

Figure 3.14 schematizes the pipeline we wanted to follow with this new setting for fine-tuning the model. Specifically, we proposed a new method which, given a previously normalized image belonging in the [0, 1] range, is able to predict the

reflectance component (also in [0, 1]) and to recover the entire dynamic range of the shading component. To do this, the original image, that is *not* normalized and therefore exceeds the range [0, 1], has to be also taken into account. In this way, when the pixel-wise division is computed between the aforementioned image and the predicted reflectance, the result will be an image with a wide dynamic range. To avoid possible divisions by 0, an $\epsilon$ with a very low value (*i.e.,* $10^{-7}$) is added to the denominator.

### Second Training

With this new setting, we tried a new fine-tuning attempt. The configuration of the learning process remains similar but with some important differences.
In the first place, we fixed the choice of the loss, the batch size and the image crop size to *L1*, 4 and 256 x 256 respectively, without trying any other options.
Furthermore, we decided to test whether adding a regularization technique could improve the performance of the model. Through the imposition of penalties on substantial weights, *L2 regularization* assists in managing the model's complexity. This can become crucial when adapting a pre-trained model to a different task, as the pre-trained model may have acquired intricate features unrelated to the new dataset.
We also made some interventions on the learning rate by experimenting with different possible values and by employing a scheduler. Specifically, we made use of the PyTorch function *ReduceLROnPlateau* [36], which dynamically adjusts the learning rate during training by reducing it when a specified metric (*e.g.,* validation loss) stops improving, helping the model converge faster and potentially escape local minima. There is a specified *patience* parameter to send as input to the function, which determines the number of epochs to wait for improvement before actually reducing the learning rate.
Another significant change we decided to implement is that now only the input images undergo tone mapping using the usual curve (3.4) in such a way as to normalize them to [0,1]. Instead, the ground truth shading components remain in their wide dynamic range because the predicted shadings are also in the full range. As a direct consequence, however, the *L1* loss measures very high distance values between the ground truth and predicted shading if compared with those obtained in the case of the reflectance. In order to make the shading loss going down, the model deeply changes the predicted reflectance ending up with really poor results. In this sense, we deemed to be necessary to add a scaling factor over the shading loss formulation in order to tackle the range mismatch with respect to the reflectance loss and bring both of them in the same scale. This scaling factor is an hyperparamter and its value has been found empirically. Taking all this into account, the loss function to be optimized is now defined as:

$$\mathcal{L}(y) = L1(R_y, \hat{R}_y) + w_s \cdot L1(S_y, \hat{S}_y) \tag{3.10}$$

As an alternative countermeasure to address the mismatch between reflectance loss and shading loss, we also thought about reformulating the latter, considering its logarithmic version. Specifically, we modified the shading images (predicted and ground truths) , before computing the *L1* loss, as follows:

$$S_y^{\log} = \log(1 + S_y) \tag{3.11}$$

$$\hat{S}_y^{\log} = \log(1 + \hat{S}_y) \tag{3.12}$$

In this way, there is no longer a need to place a scaling factor in the shading loss term because the logarithm reduces the range discrepancy. A new possible formulation of the loss function therefore becomes the following:

$$\mathcal{L}(y) = L1(R_y, \hat{R}_y) + L1(S_y^{\log}, \hat{S}_y^{\log}) \tag{3.13}$$

The last and most significant change to the fine-tuning setting concerns the addition of a formulation of a new metric in the calculation of the total loss on which the back-propagation phase is then called.
A problem that has emerged in all the results obtained so far is that the predicted reflectance lose fundamental information on the colors, which instead end up in the predicted shading. To put a constraint on this gap between the predicted reflectance and the ground truth in terms of colors, we decided to use a color difference metric often employed in image processing, called *Delta E 76*, that quantifies the perceptual colors difference in the CIE 1976 ($L^*$, $a^*$, $b^*$) color space, where $L^*$ is lightness, and $a^*$ and $b^*$ are color components:

$$\Delta E_{76}(R_y, \hat{R}_y) = \sqrt{(L_{\hat{R}_y} - L_{R_y})^2 + (a_{\hat{R}_y} - a_{R_y})^2 + (b_{\hat{R}_y} - b_{R_y})^2} \tag{3.14}$$

$L_{\hat{R}_y}$, $a_{\hat{R}_y}$, $b_{\hat{R}_y}$ represent the $L^*a^*b^*$ triplet of the predicted reflectance while $L_{R_y}$, $a_{R_y}$, $b_{R_y}$ the $L^*a^*b^*$ triplet of the ground truth reflectance. A $\Delta E_{76}$ of 0 means identical colors, while higher values indicate increasing perceptual differences.

Upon the inclusion of this metric, the formulation of the loss function is consequently revised in two other alternative ways:

$$\mathcal{L}(y) = L1(R_y, \hat{R}_y) + w_s \cdot L1(S_y, \hat{S}_y) + w_\Delta \cdot \Delta E_{76}(R_y, \hat{R}_y), \tag{3.15}$$

$$\mathcal{L}(y) = L1(R_y, \hat{R}_y) + L1(S_y^{\log}, \hat{S}_y^{\log}) + w_\Delta \cdot \Delta E_{76}(R_y, \hat{R}_y) \tag{3.16}$$

The $\Delta E_{76}$ value is weighted by another hyperparameter, $w_\Delta$, which is chosen to ensure that the model pays equal attention to all the terms that appear in the loss formulation.

| **Epochs** | **Optimizer** | **Learning Rate** | | **Loss Formulation** | **Regularization** |
|---|---|---|---|---|---|
| 25 | Adam | 0.0001 | | 3.10<br>$w_s = 0.0001$ | None |
| 50 | SGD | ReduceLROnPlateau | $patience = 10$<br>$\lambda_{init} = 0.001$<br>$\mathrm{K}_{scale} = 0.1$ | 3.15<br>$w_s = 0.0001$<br>$w_\Delta = 0.0001$ | L2<br>$\alpha_{decay} = 0.1$ |
| 100 | / | ReduceLROnPlateau | $patience = 7$<br>$\lambda_{init} = 0.0001$<br>$\mathrm{K}_{scale} = 0.1$ | 3.16<br>$w_\Delta = 0.0001$ | L2<br>$\alpha_{decay} = 0.01$ |

**Table 3.4:** Fine-tuning setting for the second major attempt.

The combination of yellow cells shown in Table 3.4 is the one that led to obtaining the best predictions. An increase in the number of epochs compared to the first attempt was found to be beneficial but, at the same time, by increasing it further no improvement was made. As regards the addition of the regularizer, it seems that this has accompanied the model to obtain more satisfactory results in some cases while in others the change made is almost imperceptible. We believed that the real discriminants of the improvement of the predicted images are the choice of a lower initial learning rate associated with an intelligent scheduling and the introduction of the $\Delta E_{76}$ metric, as it has helped alleviating the problem of color information placed in the shading rather than the reflectance.

### Results of the Second Training

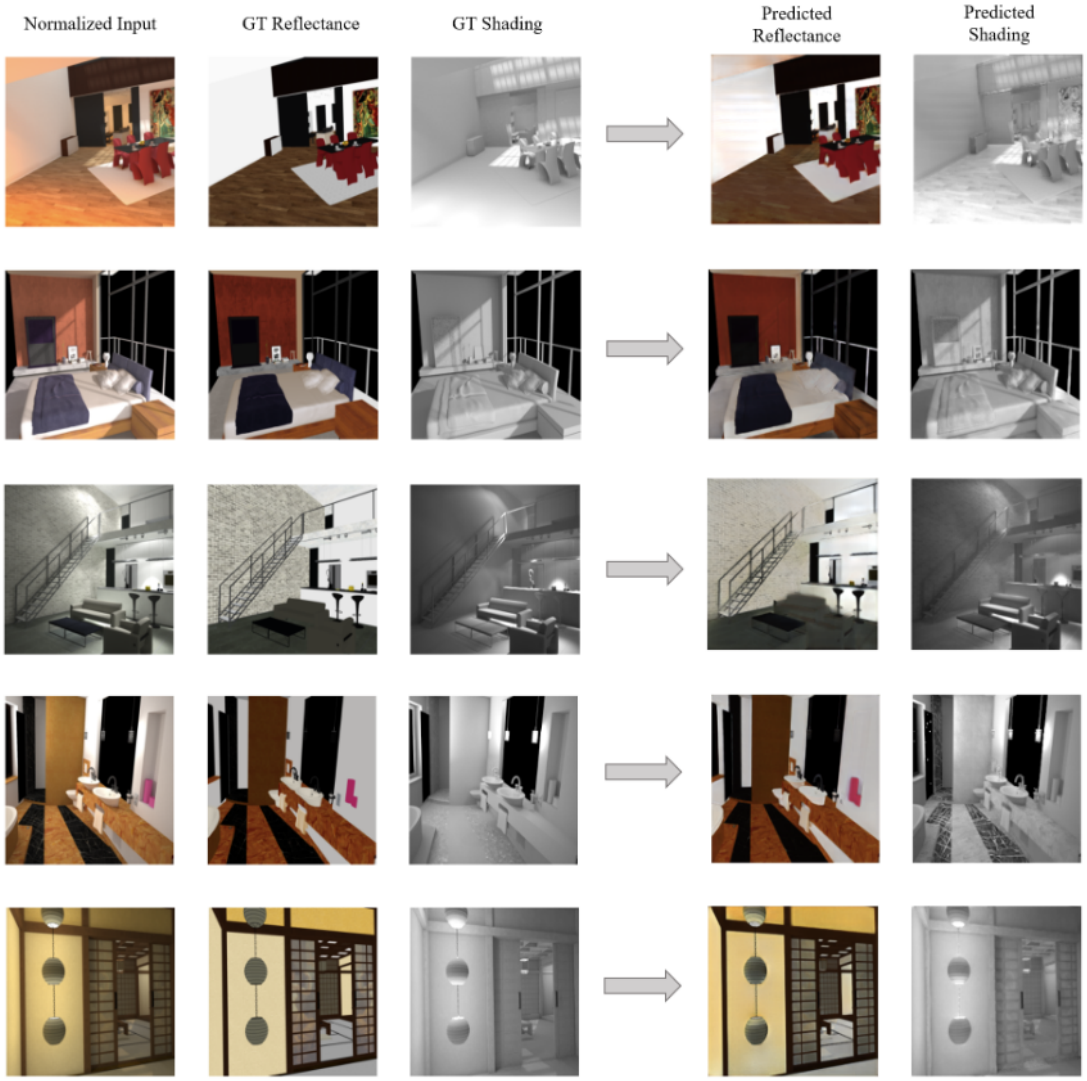


**Figure 3.15:** Results of the second fine-tuning.

Figure 3.15 shows how the predictions finally get much closer to the ground truths. The images are visually similar and in general the model has refined the way of disentangling the two intrinsic components. However, the results are not yet optimal since some artefacts are still present or in certain regions the model incorrectly classifies the reflectance and shading features.

## 3.6 IID via Ordinal Shading Predictions

The work of Careaga et al. is really recent (October 2023) and we discovered it only after having already undertaken the entire process of improving *PIE-Net*. The strong point of this model is that it has already been trained on a portion of *Hypersim*, along with other synthetic datasets that share similarities (*e.g., OpenRooms, CGIntrinsics*). Even more crucial is the model's ability to deliver very good results with real-world images, which are indeed the primary focus of our project. The strategy of training a model on synthetic images is only dictated by the fact that with this kind of dataset the ground truth decomposition is available, but the goal we want to achieve is to exploit these images to build a model whose capabilities are then tested on real-world images.
In this context, we had the opportunity to assess the quality of the results by testing the available pre-trained model on images sourced from both *Hypersim* and the *HDR Photographic Survey* [37], a collection of real-world HDR images provided by Professor Mark D. Fairchild.

### Models Comparison

As with *PIE-Net*, *IID via Ordinal Shading* also requires that the dynamic range of the input image to the model is normalized and limited in [0, 1] before determining the intrinsic components.
In this regard, the images coming from *Hypersim* are normalized with the usual gamma encoding described in 3.4. For what concerns the images from the *HDR Photographic Survey*, however, we tested two different normalization algorithms (1 and 2).

**Algorithm 1** Input Normalization

1: $p_{99} = percentile(image, 99)$
2: $image = \frac{image}{p_{99}}$
3: $image = image^{\frac{1}{2.2}}$
4: $image = clip(image,\ 0,\ 1)$

**Algorithm 2** Input Normalization

1: $p_{99} = percentile(image, 99)$
2: $image = image^{\frac{1}{2.2}}$
3: $image = \frac{image}{p_{99}}$
4: $image = clip(image,\ 0,\ 1)$

The two algorithms differ only in the order of operations 2 and 3. In the former, the image is first divided by the *99th* percentile ($p_{99}$) and then raised to the power of $\frac{1}{2.2}$. In the latter, gamma encoding is first performed and the result is then divided with $p_{99}$. We noticed that the choice of algorithm dictates the quality

of the decomposition and, on the basis of the two types of results obtained, we preferred Algorithm 2.

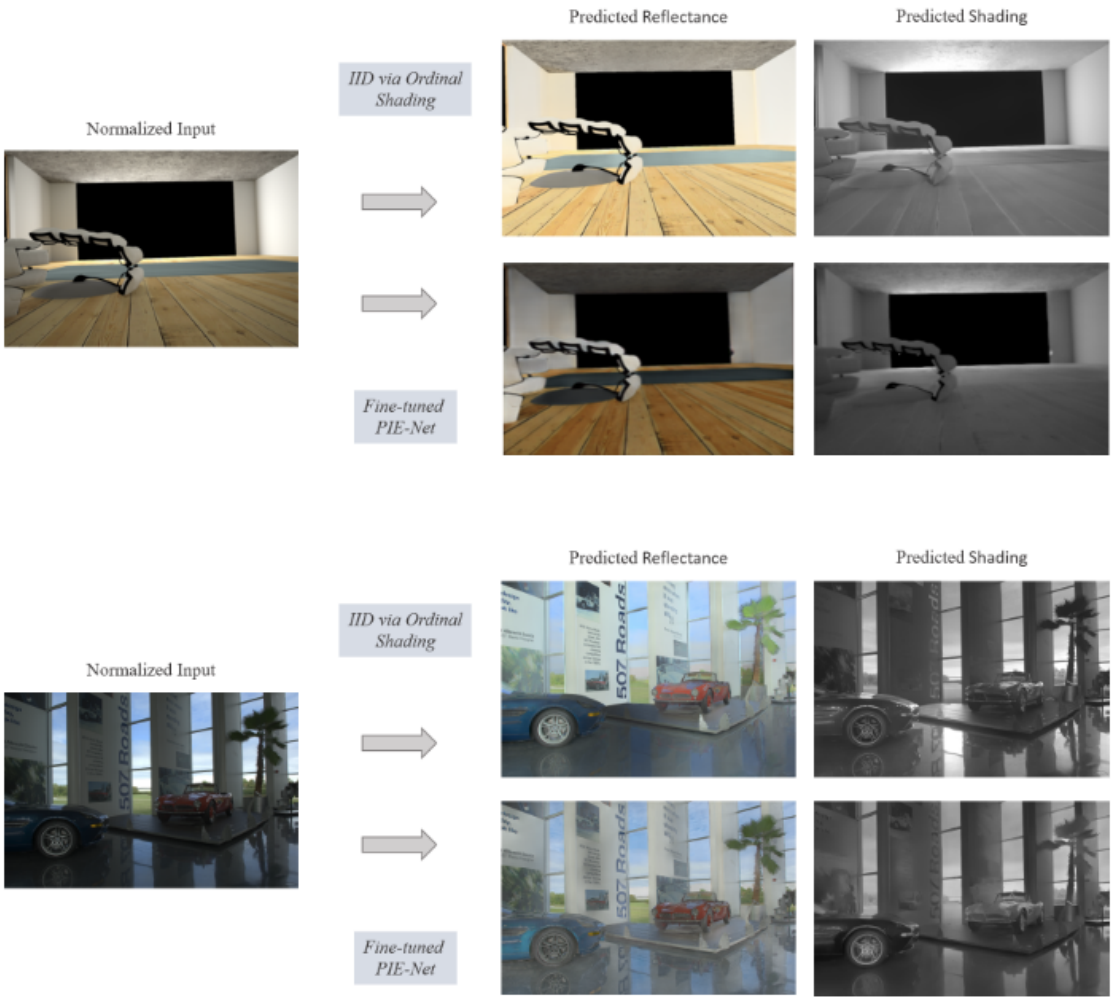


**Figure 3.16:** Decomposition comparison between *IID via Ordinal Shading* and *PIE-Net* (after fine-tuning). Image at the top is from *Hypersim*. Image at the bottom is from the *HDR Photographic Survey*.

The *IID via Ordinal Shading* predictions are notably satisfactory and if juxtaposed with the *PIE-Net* ones seem to be qualitatively better. In Figure 3.16, we can notice how Careaga's model exhibits good performance on *Hypersim* images, benefiting from its training on this dataset, but most of all how it reaches really good results for real-world images from the *HDR Photographic Survey*. In this specific case, it can be seen that, in correspondence with the panel in the background, the text is almost completely removed by the shading component predicted by the *IID*

*via Ordinal Shading* model, while in that predicted by *PIE-Net* the disentangling is less effective. At the same time, the blue color of the car on the left is also much more uniform along its entire shape in the *IID via Ordinal Shading* predicted reflectance, while it takes on different shades of blue in the *PIE-Net* one. In general, as demonstrated in Careaga et al. [34], the *PIE-Net* model often fails to disentangle the shadows in the estimated reflectance and sometimes it is either not able to reconstruct bright colors or it introduces some artifacts.
Furthermore, for an intrinsic decomposition algorithm to be practical for image editing tasks, it must be both time-efficient and memory-efficient. If the algorithm takes too long to run or requires high-end GPU resources, it can significantly restrict the applicability of the intrinsic decomposition method. The work we conducted aims to exploit a decomposition model to solve image distortion issues that happens when an HDR image is displayed (*i.e., tone mapped*) on smartphones. Implementing an highly complex neural network on these devices is unfeasible due to the stringent hardware constraints. In Careaga et al. [34], the authors claim that, for processing a 768 x 768 image, their model takes almost 26 times less than *PIE-Net*, with the same computational power. When it comes to memory efficiency, instead, *IID via Ordinal Shading* uses 7 gigabytes less compared to *PIE-Net*.
Having noted that *IID via Ordinal Shading* outperforms any other decomposition model we have seen so far in terms of efficiency and quality of the results, we therefore decided to exploit its decomposition power in order to move to the next step and finally carry out the tone mapping operation, following the scheme described in 3.2.

# Chapter 4

# Tone Mapping Step

## 4.1 Final Objective

Considering the intrinsic components of reflectance $R$ and shading $S$, predicted from an input image $I$, and denoting any TMO as $t$, the outcome of the tone mapping process through the IID step can be expressed as follows:

$$t^{IID}(I) = R \odot t(S) \tag{4.1}$$

where $t^{IID}(I)$ indicates the tone mapped image obtained by first carrying out the tone mapping operation only on the shading component ($t(S)$) and then multiplying the result with the reflectance component in an element-wise manner. Once the decomposition model has been chosen and the final images have been produced as described above, the objective becomes that of evaluating these results by comparing them with those that would be obtained from the application of some very well known post processing color reproduction methods.
As already explained, tone mapping serves a dual purpose, requiring the manipulation of images to compress their absolute luminance range while also adjusting pixel relationships to enhance visible detail and alter overall contrast. Nevertheless, modifications to contrast and luminance frequently result in shifts in color appearance, impacting saturation and hue. If the overarching purpose is to accurately depict a scene on a display, maintaining its authenticity and without altering its appearance, what is done very often in the literature is to develop color correction algorithms that mostly act on colors by reducing the distortion introduced into the tone mapped image, if compared with the original one. These algorithms perform therefore a processing of the tone mapped images that occurs *a posteriori* and which tries to compensate for the alterations produced by a specific TMO.

In the context of our work, we selected three very famous methodologies, that aim at solving the same issue, as baseline against which we can compare our results:

- **Quantization techniques for the visualization of high dynamic range pictures** from Schlick [38].
- **Color Correction for Tone Mapping** from Mantiuk et al. [39]
- **Automatic saturation correction for dynamic range management algorithms** from Artusi et al. [40]

## 4.2 Color Correction Algorithms

### 4.2.1 Schlick and Mantiuk Methods

Schlick's approach [38] to color treatment in tone mapping involves introducing a saturation control parameter $s$ in the equation. This method focuses on preserving color ratios by adjusting the output color ($C_{out}$) based on the input color ($C_{in}$) and the ratio between the luminance of the tone mapped image ($L_{out}$) and the luminance of the original input ($L_{in}$).

$$C_{out} = \left(\frac{C_{in}}{L_{in}}\right)^{s} \cdot L_{out} \tag{4.2}$$

However, this formulation has its drawbacks, as it can significantly impact the luminance, deviating from the desired outcome, especially for highly saturated pixels.
In the realm of Mantiuk's method [39], instead, the color correction is performed as follows:

$$C_{out} = (\left(\frac{C_{in}}{L_{in}} - 1\right) s + 1) \cdot L_{out} \tag{4.3}$$

This formula retains luminance while exclusively incorporating linear interpolation between chromatic and corresponding achromatic colors. The adjustment in the color correction factor in Equation 4.2 not only alters chroma but also impacts the lightness of colors. In Equation 4.3, instead, such shifts in lightness are mitigated. However, it introduces a more pronounced hue shift, particularly noticeable in red and blue colors.
For both these two algorithms, manual parameter $s$ adjustment is often necessary, since it allows to produce more accurate results although, at the same time, it can

be very time consuming. Automation can be carried out by exploiting the slope of the tone curve at each luminance level. Specifically, the parameter $s$ can be deduced by computing the following formula:

$$s = \frac{(1 + k_1)c^{k2}}{1 + k_1 c^{k2}} \tag{4.4}$$

where $k_1$ and $k_2$ are constants whose value varies depending on the correction method used. Specifically, in Schlick's case, $k_1 = 1.6774$ and $k_2 = 0.9925$. Instead, for Mantiuk's method, $k_1 = 2.3892$ and $k_2 = 0.8552$. The factor $c$ signifies the degree of compression or expansion implemented and it can be estimated using the gradient of the tone curve when plotted on a logarithmic scale.
Considering $\hat{L} = \log(L)$, $c$ is computed as follows:

$$\hat{L}_{out} = t(\hat{L}_{in}) \tag{4.5}$$

$$c(L_{in}) = \frac{d}{d\hat{L}} t(\hat{L}_{in}) \tag{4.6}$$

This way of estimating $s$ yields satisfactory outcomes provided that certain prerequisites are fulfilled. The most crucial of these is that the TMO must be global.

### 4.2.2 Artusi Method

In Artusi et al. [40], an entirely automated method for saturation correction is introduced. The algorithm aims to adjust the tone mapped image so that its color presentation aligns with the hue and saturation of the HDR image, while maintaining the same luminance levels. The inputs consist of the original HDR image and the distorted (*i.e.,* tone mapped) one, both given in linear RGB space. After a normalization with the maximum, the two images are first converted into *IPT* space and then moved into the cylindrical color space *ICh*.
As regards this last color space transformation, the $I$ (*i.e.,* lightness) channel of the image is left unchanged while the $h$ (*i.e.,* hue) and $C$ (*i.e.,* chroma) ones are defined as follows:

$$h = \tan^{-1}\left(\frac{P}{T}\right) \tag{4.7}$$

$$C = \sqrt{P^2 + T^2} \tag{4.8}$$

At this point the chroma channel of the tone mapped image, $C_t$, is first scaled in such a way that it approximate the result if the tone mapping of the original HDR image is performed in the *ICh* space:

$$C'_t = C_t \frac{I_0}{I_t} \tag{4.9}$$

where $I_0$ and $I_t$ respectively denote the lightness of the unprocessed HDR image and of the tone mapped one. Afterwards, the corrected chroma channel $C_c$ is finally computed by performing:

$$C_c = rC'_t \tag{4.10}$$

where $r$ represents the ratio between the saturation of the original and the (*scaled*) tone mapped image which are computed following Equation 4.12

$$r = \frac{s(C_0, I_0)}{s(C_t, I_t)} \tag{4.11}$$

$$s(C, I) = \frac{C}{\sqrt{C^2 + I^2}} \tag{4.12}$$

As a last step, the hue $h_t$ of the tone mapped image is restored by replicating the value $h_0$ from the HDR image. The combination of the new hue, in conjunction with the adjusted chroma $C_c$ and the preserved lightness $I_t$, yields the final corrected output, which can subsequently be transformed back into the *RGB* color space.

## 4.3 Visual Comparison

The algorithm illustrated below exactly describes the process followed to produce the final results of the tone mapped images passing through the intermediate decomposition step.

**Algorithm 3** Tone Mapping through IID

**function** IID-TMO(*image*, *shading*)
    ▷ *image* is the unprocessed HDR image
    ▷ *shading* is the intrinsic component predicted by the IID model when the encoded version (Algorithm 2) of *image* is given as input
    ▷ *TMO* indicates a certain global tone mapping operator
    ▷ Normalize both images
    $image \leftarrow \dfrac{image}{max(image)}$
    $shading \leftarrow \dfrac{shading}{max(shading)}$
    ▷ Retrieve the reflectance
    $reflectance = image \odot \dfrac{1}{shading}$ ▷ Pixel-wise division
    ▷ Tone map the *shading*
    $shading_{tm} = TMO(shading)$
    ▷ Multiply with the reflectance
    $image_{iid-tm} = reflectance \odot shading_{tm}$ ▷ Pixel-wise multiplication
    ▷ Adjust the brightness
    $p_{99.5} = percentile(image_{iid-tm}, 99.5)$ ▷ Compute $99.5th$ percentile
    $image_{iid-tm} \leftarrow \dfrac{image_{iid-tm}}{p_{99.5}}$
    $image_{iid-tm}(image_{iid-tm} > 1) \leftarrow 1$ ▷ Clipping
    **return** $image_{iid-tm}$
**end function**

It can be easily seen that, to obtain the final result ($image_{iid-tm}$), the algorithm does not only consist of taking the tone mapped shading and multiplying it with the reflectance. In fact, further interventions are carried out on each of the images involved in the total process. Each of these interventions is aimed at solving a low brightness problem that $image_{iid-tm}$ shows. We have in fact noticed that, generally, the images produced through the IID method are darker than those that would be obtained following the traditional approach.
As a first approach to the problem, we thought of scaling the pixel values of $image_{iid-tm}$ through a multiplicative factor $\alpha$ computed as follows:

$$\alpha = \frac{\mu(L_{tm})}{\mu(L_{iid-tm})} \tag{4.13}$$

$$image_{iid-tm} = \alpha \cdot image_{iid-tm} \tag{4.14}$$

where $L_{iid-tm}$ and $L_{tm}$ represent respectively the luminance component of the directly tone mapped image and that of the tone mapped image by passing through the decomposition step, whereas $\mu(*)$ returns the mean value. While this workaround helps to fix the brightness mismatch issue, it also causes strong clipping in some regions of most tested images.
The approach proposed in Algorithm 3 instead involves a normalization with the percentile, which guarantees the obtaining of brighter images while simultaneously avoiding producing too many unwanted clipping effects.

In order to evaluate the effectiveness of the results we obtain following our algorithm, we made use of two different TMOs:

- *Reinhard '02 (Photographic Operator)* [41]
- *Reinhard-Devlin '05* [42]

Therefore, we juxtaposed our outcomes with the ones from the Artusi's method and the implementations of Schlick's and Mantiuk's algorithms with the saturation parameter automatically estimated from the tone curve, following the procedure described in Equation 4.4. For both TMOs, we exploited the implementation present in the HDR Toolbox [40] and we evaluated them using the default parameters values.

**Figure 4.1:** Tone mapping results for *Reinhard '02 TMO*. Each column represents the method employed. From left to right: *Traditional Reinhard '02*, *Schlick*, *Mantiuk*, *Artusi* and *Ours*.

**Figure 4.2:** Tone mapping results for *Reinhard-Devlin '05 TMO*. Each column represents the method employed. From left to right: *Traditional Reinhard-Devlin '05*, *Schlick*, *Mantiuk*, *Artusi* and *Ours*.

## 4.4 Evaluation through Objective Metrics

What we want to demonstrate with this work is that preferring a preliminary decomposition of the image can bring advantages in terms of color distortion and overall picture quality when performing tone mapping. The most immediate and intuitive approach to prove this is to make use of specific and accurate metrics which, in a certain way, can provide an indication of the level of goodness of the results obtained. Nevertheless, quantifying this hypothetical advantage with an objective metric that gives a precise score can be rather challenging. In Image Processing, there are many metrics that try to judge an image according to some specific criteria. However, very often it is difficult to understand on what basis they are founded and how close the returned result is to the perceived image quality. In addition, while a large number of image metrics are designed for the assessment of LDR content, far fewer metrics are available for HDR content.
Taking into account what has been said, we decided to make use of 3 different metrics, each of which contributes to the evaluation of specific and different aspects of the results obtained.

### Hue Distance

The *Hue Distance* metric is able to exclusively focus on the hue component ($h$) and quantify the difference between two images: the original, unprocessed HDR image and the distorted version after tone mapping. The metric is presented and described in [40] and its implementation can be found in the HDR Toolbox [20].
Going into more detail, the algorithm first converts the images from the *RGB* color space into the *IPT* color space for then transitioning to the cylindrical *ICh* color space, where the distance is actually computed, providing a more accurate measure of color differences. Usually, color differences are not calculated by independently considering the luminance and the hue components. Furthermore, these measurements are often obtained in the *CIE L* a* b** color space, but it has been proven that this space does not maintain a consistent hue across all hues. This metric overcomes both these two issues and outputs a score that the lower the better. Figures 4.3 and 4.4 show, in logarithmic scale, the average hue distance values computed for each proposed method. From the bar graphs, it is clear that Artusi's method and ours obtain better results than those of Schlick and Mantiuk. Specifically, Artusi's method leads to a lower distortion in the case of *Reinhard '02* while our algorithm shows remarkable behavior with *Reinhard-Devlin '05*, with a distortion score 4-5 orders of magnitude lower than that of the other methods.

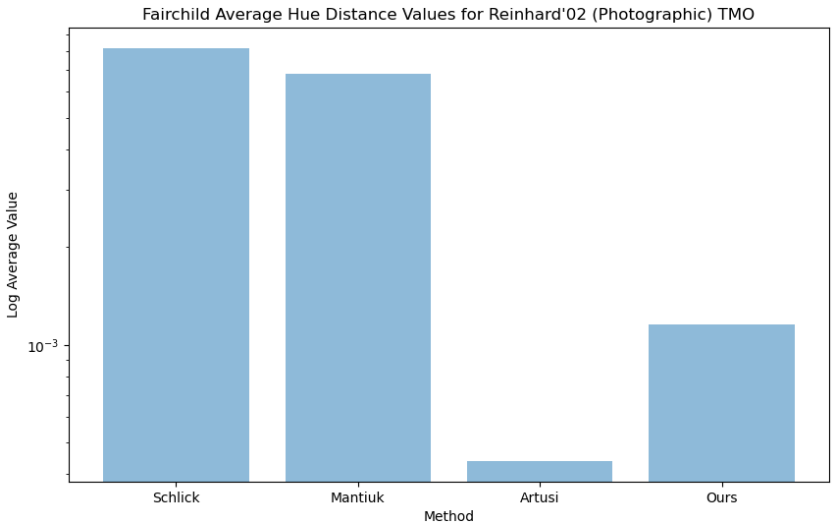


**Figure 4.3:** Hue distance values for Reinhard '02 (Photographic) TMO. Results are computed for all the color correction methods and our proposed approach, considering all the images present in the *HDR Photographic Survey*.

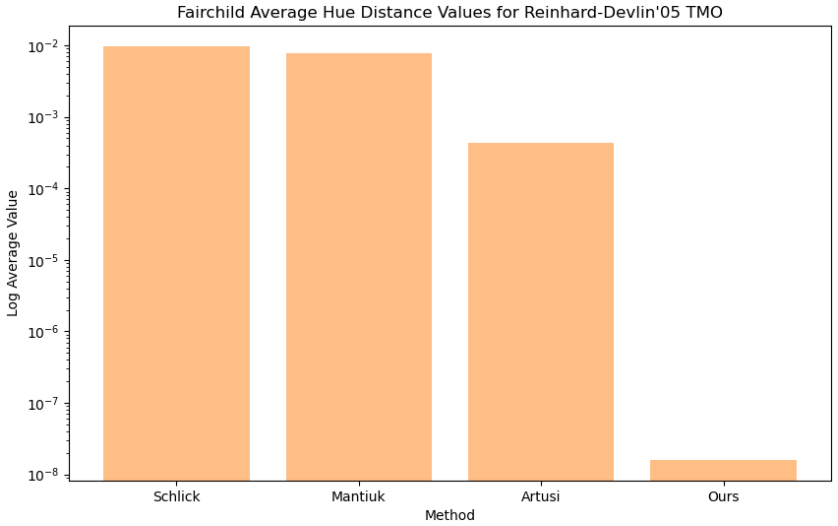


**Figure 4.4:** Hue distance values for Reinhard-Devlin '05 TMO. Results are computed for all the color correction methods and our proposed approach, considering all the images present in the *HDR Photographic Survey*.

### TMQI

The implementation of this metric is described in Yeganeh et al. [43]. The authors developed an objective Image Quality Assessment (IQA) model for tone mapped

LDR images, using their corresponding HDR images as references. This work draws inspiration from two successful design principles in IQA literature. The first one is the *structural fidelity*, which posits that the primary function of vision is to extract structural information from the visual scene. Therefore, it is a reliable predictor of perceptual quality. The second principle is the *statistical naturalness* which suggests that the visual system is highly adapted to the natural visual environment and uses deviations from natural image statistics as a measure of perceptual quality. A high-quality tone-mapped image should strike a balance between preserving *structural fidelity* and maintaining *statistical naturalness*, which can sometimes be competing factors. This metric combines the two measures, culminating in one unique score, the Tone Mapped image Quality Index (TMQI). Here, the aim is to maximize the score.

| **Method** | **TMQI** | **Fidelity** | **Naturalness** |
|---|---|---|---|
| Schlick | **0.7950** | 0.7837 | 0.1859 |
| Mantiuk | 0.7947 | 0.7825 | **0.1865** |
| Artusi | 0.7939 | 0.7818 | 0.1837 |
| Ours | 0.7925 | **0.7907** | 0.1812 |

**Table 4.1:** TMQI scores for Reinhard '02 (Photographic Operator) TMO. Results are computed for all the color correction methods and our proposed approach, considering all the images present in the *HDR Photographic Survey*.

| **Method** | **TMQI** | **Fidelity** | **Naturalness** |
|---|---|---|---|
| Schlick | 0.7621 | 0.7399 | 0.1107 |
| Mantiuk | 0.7605 | 0.7374 | 0.1076 |
| Artusi | 0.7625 | 0.7387 | 0.1145 |
| Ours | **0.7849** | **0.7666** | **0.1807** |

**Table 4.2:** TMQI scores for Reinhard-Devlin '05 TMO. Results are computed for all the color correction methods and our proposed approach, considering all the images present in the *HDR Photographic Survey*.

Table 4.1 shows the scores that each method under analysis obtains when the images are tone mapped using the photographic version of the *Reinhard '02* TMO.

The TMQI values are very close to each other. Our proposed approach obtains a better result in terms of *structural fidelity* but seems to be penalized by the criterion that measures the *naturalness* of the image.
In Table 4.2, instead, we can see how once again when the TMO under analysis is *Reinhard-Devlin '05*, our algorithm achieves the best result.

### HDR-VDP3

The HDR-VDP3 metric, presented in Aydin et al. [44], is designed for assessing perceived quality in HDR images. As with TMQI, the notable feature of this metric is that it is dynamic range independence, meaning that it is able to handle and produce a comparison taking simultaneously into account HDR and LDR images. By incorporating human perceptual models, the metric aims to align with observer sensitivity, providing a comprehensive evaluation of HDR image quality in a perceptually relevant manner. In order to represent regions of the LDR image where distortions occur, two distortion maps are generated:

- *Loss of visible contrast*, which occurs occurs when a contrast that was perceptible in the original image becomes imperceptible in the distorted one.
- *Amplification of invisible contrast*, which happens when a contrast that was not discernible in the original image becomes noticeable in the distorted one.

Because of time constraints and the less good metric results we obtained with our method using the *Reinhard '02* TMO to test the images, we decided to evaluate with this new metric only for this TMO.
Figure 4.5 shows for three different images from the *HDR Photographic Survey*, the distortion maps associated with each of the four methods under analysis. The regions in green highlight distortions in the image attributable to the loss of visible contrast; those in blue instead refer to the amplification of the invisible contrast. It is immediate to notice how, in the two top images, both distortions are minimized following our proposed approach. In the image at the bottom, although the blue distortion is essentially the same for each method, the green distortion is present in less quantity when following the IID pipeline.

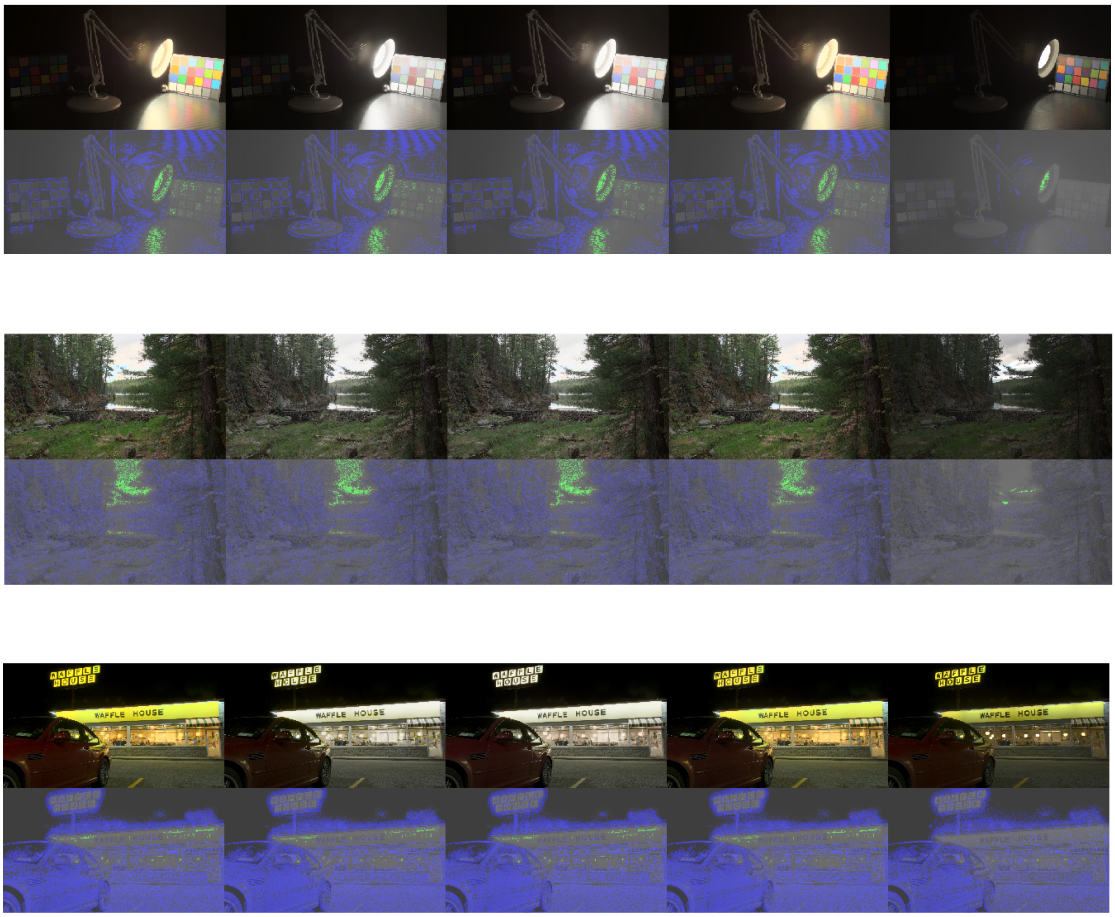


**Figure 4.5:** Examples of *distortion maps* produced by the HDR-VDP3 metric for three different images from *HDR Photographic Survey*. Each column represents the method employed. From left to right: *Reinhard '02*, *Schlick*, *Mantiuk*, *Artusi* and *Ours*. Green regions indicate distortion referable to loss of visible contrast. Blue regions represent distortion referable to amplification of invisible contrast.

The reported results are to be linked to the following information:
HDR-VDP v3.0.7, 52.72 [PIX/DEG], (CPU).

While these high-quality maps demonstrate strong correlations with subjective assessments of image degradation, they fall short in offering a singular quality score for an entire image. Although it's not exactly recommended, we thought about performing a pooling and get a numerical estimate of the general distortion that occurs on the images from *HDR Photographic Survey*. Specifically, the metric produces for each sample image two additional 1-channel images, respectively containing information on the specific type of distortion. We therefore computed the average of each of these two images (for all the dataset) and obtained two unique scores that quantify the two distortions.

| **Method** | **Loss** | **Amplification** |
|---|---|---|
| Schlick | 0.0031 | 0.2022 |
| Mantiuk | 0.0032 | 0.2002 |
| Artusi | 0.0029 | 0.2010 |
| Ours | **0.0003** | **0.1855** |

**Table 4.3:** HDR-VDP3 scores for Reinhard '02 (Photographic Operator) TMO. Results are computed for all the color correction methods and our proposed approach, considering all the images present in the *HDR Photographic Survey* dataset

As already suggested by the examples shown in the Figure 4.5, the metric detects the least amount of distortion in images tone mapped following our method.

## 4.5 Psychophysical Experiment

To corroborate the idea of getting better objective results following the approach we proposed, the next step that is conventionally done is to conduct a psychophysical experiment.
This experiment aims to have participants express their preferences between images processed using our algorithm and those generated by the three previously described methods. The established choice criterion is that of the perceptual proximity of the tone mapped image to the original HDR one, which therefore acts as a reference. Since the experiment requires the reproduction of the unprocessed content of the HDR image, there is the need of using a display which is actually capable of handling HDR images producing a wider range of brightness levels and colors. Specifically, the display used is an EIZO CG3145 with a peak luminance equal to 1000 $cd/m^2$.

**Figure 4.6:** Experiment setup. The subject looks at the HDR image first. Then a gray scale image is projected for an adaptation time for the eyes. Finally, the subject is asked to express a preference between 2 options and for 6 times for each test image.

The experiment requires the expression of a preference for 20/25 images randomly selected from the *HDR Photographic Survey*. The type of experiment is 2AFC (2-Alternative-Forced-Choice), meaning that each time the subject will be asked to choose between two options. Since we have 4 methods to compare in total, the pair comparison will be done 6 times for each HDR image analyzed.
All images (*i.e., HDR and tone mapped*) are encoded with a PQ curve that the display removes when it projects them. Perceptual Quantization (PQ) encoding is often preferred over simple gamma encoding for HDR images due to its ability to handle a wider brightness range and its basis on human visual perception.
It is important to highlight that when the tone mapped images are shown for the pair comparison, the display simulates the properties of an SDR display, emitting a limited range of brightness, with a peak up to 100 $cd/m^2$.
Both for reasons of time and the fact that objective data suggests that our proposed approach does not achieve the best result when *Reinhard '02 TMO* is used, it was decided to limit the experiment by using images tone mapped with this operator. The experiment has not yet been finished, but we are confident that we can collect the final results in the shortest time possible. From the first analyses, the results that emerged support the quantitative findings, revealing a consistent preference among participants for images processed through our proposed method.

# Chapter 5

# Conclusions

## 5.1 Closing Remarks and Future Works

In conclusion, this thesis addressed the challenges associated with traditional tone mapping approach, highlighting its inherent limitations.
Driven by the scientific evidence that the human visual system seems to be less sensitive to the effects of light and tends to discard them, the innovation introduced by this work is the use of AI-based models for intrinsic decomposition as a means through which to perform the tone mapping operation. Through a comprehensive review of state-of-the-art IID works, we first fine-tuned one model and then leveraged the capabilities of another one, ultimately generating the final tone mapped images. The benchmarking process against three widely recognized alternative methods revealed that our proposed approach outperformed them in terms of objective metrics in most cases, affirming its efficacy in achieving superior results. Building on this promising success, the next crucial step in our research involves finishing the psychophysical experiment, with the setup described above.
The real discriminating part of this project is the IID task. The more refined and optimal the separation between reflectance and shading, the more effective the tone mapping operation will be. Future research with the aim of improving the results must therefore mainly focus on improving the decomposition step. At the same time, the work was carried out only analyzing traditional TMOs. It therefore becomes interesting to try to include more complex tone mapping solutions, perhaps also based on artificial intelligence algorithms.

# Acknowledgements

First and foremost, I express all my sincere gratitude to Praveen. His unwavering presence and unparalleled expertise were indispensable in steering this project to completion.
I would also like to extend my thanks to Joe for his remarkable efficiency and meticulousness in contributing to the project.
Additionally, I am deeply thankful to professors Artusi and Banterle for their invaluable support and enriching inputs.
Special appreciation also go to my academic professors Bottino and Dugelay for trusting me in embracing this work.
I also want to express my affection to everyone I encountered at the Huawei Nice Research Center for their warm welcome and constant inclusion during this experience.
Finally, a special thank you to all the people who have been closest to me during this journey, leading me to achieve goals that seemed far away and giving me the strength to overcome the difficulties.

# Bibliography


[1] *argparse — Parser for command-line options, arguments and subcommands*. Python Software Foundation. `https://docs.python.org/3/library/argparse.html` (cit. on p. 4).

[2] G. Bradski. «The OpenCV Library». In: *Dr. Dobb's Journal of Software Tools* (2000) (cit. on p. 4).

[3] *glob — Unix style pathname pattern expansion*. Python Software Foundation. `https://docs.python.org/3/library/glob.html` (cit. on p. 4).

[4] Andrew Collette and H5py contributors. *H5py: A Python interface to HDF5*. `http://www.h5py.org` (cit. on p. 4).

[5] ImageIO contributors. *ImageIO: Python library for reading and writing images*. ImageIO. `https://imageio.github.io/` (cit. on p. 4).

[6] Matplotlib Development Team. *Matplotlib: A 2D plotting library*. `https://matplotlib.org` (cit. on p. 4).

[7] Travis E. Oliphant and NumPy contributors. *NumPy: The fundamental package for scientific computing with Python*. NumPy. `https://numpy.org/` (cit. on p. 4).

[8] *os — Miscellaneous operating system interfaces*. Python Software Foundation. `https://docs.python.org/3/library/os.html` (cit. on p. 4).

[9] Wes McKinney and pandas contributors. *pandas: Powerful data structures for data analysis, time series, and statistics*. pandas development team. `https://pandas.pydata.org/` (cit. on p. 4).

[10] Alex Clark and Contributors. *Pillow (PIL Fork): Python Imaging Library*. Python Imaging Library (PIL) Fork. `https://pillow.readthedocs.io/` (cit. on p. 5).

[11] *random — Generate pseudo-random numbers*. Python Software Foundation. `https://docs.python.org/3/library/random.html` (cit. on p. 5).

[12] *re — Regular expression operations*. Python Software Foundation. `https://docs.python.org/3/library/re.html` (cit. on p. 5).

[13] SciPy developers. *SciPy: Open Source Scientific Tools for Python*. SciPy. https://www.scipy.org/ (cit. on p. 5).

[14] *shutil — High-level file operations*. Python Software Foundation. https://docs.python.org/3/library/shutil.html (cit. on p. 5).

[15] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. *scikit-image: Image processing in Python*. scikit-image contributors. https://scikit-image.org/ (cit. on p. 5).

[16] *time — Time access and conversions*. Python Software Foundation. https://docs.python.org/3/library/time.html (cit. on p. 5).

[17] PyTorch core team. *PyTorch: An open source deep learning platform*. Facebook AI Research (FAIR). https://pytorch.org/ (cit. on p. 5).

[18] PyTorch Vision contributors. *torchvision: Datasets, transforms, and models for computer vision*. PyTorch. https://pytorch.org/vision/ (cit. on p. 5).

[19] Noam Yorav-Raphael. *tqdm: Fast, Extensible Progress Bar for Loops in Python*. https://tqdm.github.io/ (cit. on p. 5).

[20] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced High Dynamic Range Imaging (2nd Edition)*. Natick, MA, USA: AK Peters (CRC Press), July 2017. ISBN: 9781498706940 (cit. on pp. 5, 51).

[21] Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. «High Dynamic Range Imaging». In: *Journal Name* (Apr. 2016). Published on April 18, 2016 (cit. on pp. 7, 11, 17).

[22] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. «Photographic Tone Reproduction for Digital Images». In: *ACM Transactions on Graphics (TOG)* 21.3 (2002), pp. 267–276. DOI: 10.1145/566570.566575 (cit. on p. 12).

[23] Wikipedia contributors. *Rec. 709 — Wikipedia, The Free Encyclopedia*. 2023. URL: https://en.wikipedia.org/wiki/Rec._709 (cit. on p. 12).

[24] Z. Li and N. Snavely. «CGIntrinsics: Better Intrinsic Image Decomposition through Physically-Based Rendering». In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on pp. 16, 20, 21).

[25] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. «A naturalistic open source movie for optical flow evaluation». In: *European Conf. on Computer Vision (ECCV)*. Ed. by A. Fitzgibbon et al. (Eds.) Part IV, LNCS 7577. Springer-Verlag, Oct. 2012, pp. 611–625 (cit. on p. 19).

[26] R. Grosse, M.K. Johnson, E.H. Adelson, and W.T. Freeman. «Ground truth dataset and baseline evaluations for intrinsic image algorithms». In: *Proceedings of the International Conference on Computer Vision (ICCV)*. 2009 (cit. on p. 19).

[27] J. Shi, Y. Dong, H. Su, and S.X. Yu. «Learning non-Lambertian object intrinsics across ShapeNet categories». In: *Proc. Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 19).

[28] S. Bell, K. Bala, and N. Snavely. «Intrinsic images in the wild». In: *ACM Trans. Graphics* 33.4 (2014), p. 159 (cit. on p. 19).

[29] B. Kovacs, S. Bell, N. Snavely, and K. Bala. «Shading annotations in the wild». In: *Proc. Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 850–859 (cit. on p. 19).

[30] Z. Li et al. «OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets». In: *Proc. Computer Vision and Pattern Recognition (CVPR)*. 2021 (cit. on p. 20).

[31] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. «Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding». In: *International Conference on Computer Vision (ICCV) 2021*. 2021 (cit. on pp. 20, 25).

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. «Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks». In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017 (cit. on p. 21).

[33] P. Das, S. Karaoglu, and T. Gevers. «PIE-Net: Photometric Invariant Edge Guided Network for Intrinsic Image Decomposition». In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 (cit. on pp. 22, 23, 25).

[34] Chris Careaga and Yağız Aksoy. «Intrinsic Image Decomposition via Ordinal Shading». In: *ACM Trans. Graph.* (2023) (cit. on pp. 24, 25, 43).

[35] Anil S Baslamisli, Thomas T Groenestege, Partha Das, Hoang-An Le, Sezer Karaoglu, and Theo Gevers. «Joint learning of intrinsic images and semantic segmentation». In: *European Conference on Computer Vision*. Springer. 2018, pp. 286–302 (cit. on p. 29).

[36] PyTorch Contributors. *torch.optim.lr_scheduler.ReduceLROnPlateau*. PyTorch. 2022 (cit. on p. 37).

[37] Mark Fairchild. *HDR Photographic Survey* (cit. on p. 41).

[38] Christophe Schlick. «Quantization Techniques for Visualization of High Dynamic Range Pictures». In: *Photorealistic Rendering Techniques*. 1995 (cit. on p. 45).

[39] Radoslaw Mantiuk, Rafał K. Mantiuk, Anna Lewandowska, and Wolfgang Heidrich. «Color correction for tone mapping». In: *Computer Graphics Forum* 28 (2009) (cit. on p. 45).

[40] Alessandro Artusi, Tania Pouli, Francesco Banterle, and Ahmet Oguz Akyuz. «Automatic saturation correction for dynamic range management algorithms». In: *Signal Processing: Image Communication* 63 (Apr. 2018), pp. 110–120 (cit. on pp. 45, 46, 49, 51).

[41] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. «Photographic tone reproduction for digital images». In: *Proceedings of SIGGRAPH 2002*. ACM. 2002, pp. 267–276 (cit. on p. 49).

[42] Erik Reinhard and Kate Devlin. «Dynamic range reduction inspired by photoreceptor physiology». In: *IEEE Transactions on Visualization and Computer Graphics* 11.1 (2005), pp. 13–24 (cit. on p. 49).

[43] Hojatollah Yeganeh and Zhou Wang. «Objective quality assessment of tone-mapped images». In: *IEEE Transactions on Image Processing* 22.2 (2013), pp. 657–667 (cit. on p. 52).

[44] Tolga O Aydin, Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. «Dynamic range independent image quality assessment». In: *ACM Transactions on Graphics (TOG)* 27.3 (2008), p. 69 (cit. on p. 54).