High Dynamic Range Imaging Systems for Dynamic Scenes

Fahd Bouzaraa

Technische Universität München
Lehrstuhl für Datenverarbeitung

High Dynamic Range Imaging Systems for Dynamic Scenes

Fahd Bouzaraa

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Georg Sigl

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Klaus Diepold
2. Prof. Dr.-Ing. Eckehard Steinbach

Die Dissertation wurde am 10.03.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 05.10.2017 angenommen.

Acknowledgments

Firstly, I would like to extend my gratitude to my thesis supervisor Dr. Onay Urfalioglu for his strong and professional support. His guidance and valuable advises made this thesis possible.

Furthermore, my gratitude goes as well to my thesis advisor Prof. Klaus Diepold for his extremely valuable counseling and his understanding of the different challenges I faced during this PhD thesis. His unique affinity for innovation and efficient problem-solving skills are true lessons to appreciate and embrace for my future life and career.

Special thanks as well to my fellow teammates at the *Media Lab*, and especially to Ibrahim for our long and inspiring conversations about my work.

I would like to finish by dearly thanking my parents Hayouta and Hmida for their love, dedication and their unshakable trust in me and my skills. This thesis is yours as much as it is mine. Lastly, I would like to thank my brother Sayed and sister Jihen, their spouses and my three little angels Lina, Elias and Yassine.

Abstract

In the past decades, digital photography has gained a central role in groundbreaking technologies, where the capturing, manipulation, examination and transmission of images and videos using digital capturing devices represent important building blocks of the related applications. This includes *Artificial Intelligence* systems where the understanding and the analysis of the machine's environment is based on the information contained in the available digital material, namely images and videos. Furthermore, the recent technological advances made on handheld communication devices such as smartphones and tablet computers, together with the emergence of social media platforms, reshaped the functionality of digital images and videos as essential components of these technologies. In this context, digital photography is no longer perceived as an exclusively professional activity where high-end devices such as *Digital Single-Lens Reflex* (DSLR) cameras are necessary, but is increasingly considered as an essential byproduct of accessible devices such as smartphones and tablets computers.

With this in mind, emerging research topics related to digital photography such as *High Dynamic Range Imaging* (HDRI) play an important role in shaping the future of computational photography and computer vision. In this work, we provide a broad analysis of the topic of HDRI and the related challenges, especially in the context of dynamic scenes. The study of these challenges enables us to develop an initial low-complexity de-ghosting approach capable of rendering an artifact-free HDR image. The subsequent examination of our method and the gained results moves us towards the investigation of related topics, such as *color mapping*. This in turn allows us to develop a novel HDR rendering framework which offers the desired trade-off between the computational complexity of the enabling algorithm, and the quality of the gained results.

Contents

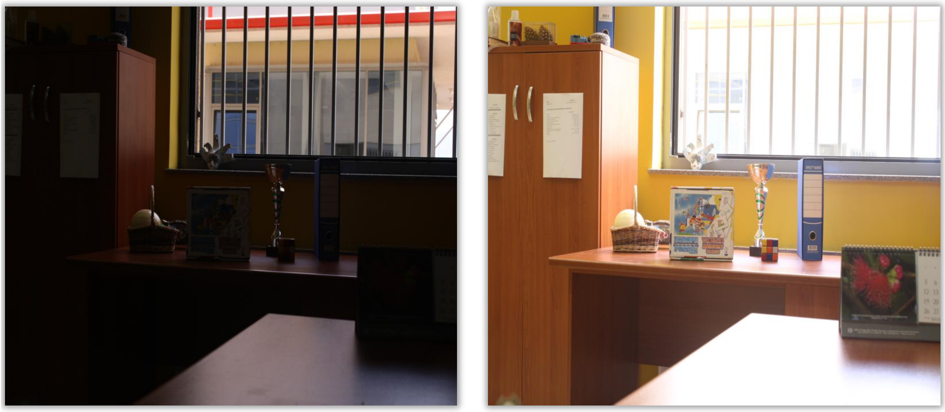
List of Figures

1 Introduction

The rising significance of digital photography as a daily consumer product increased the importance of image quality in terms of the depicted details as well as the amount of encompassed noise and artifacts. In this context, image quality is mostly linked to color-related characteristics. Aside from the well investigated topic of image de-noising [16, 17, 18] the perceptual quality of digital images is correlated with the portrayed portion of the *Dynamic Range* corresponding to the depicted scene. The dynamic range is determined by the ratio between the brightest and the darkest luminance (light) values emitted by the objects in the depicted scene [19]. Evidently, the amount of represented dynamic range impacts the quality of the contained details as well as the overall contrast ratio. The range of perceivable light expressed through luminance and measured in *Candelas per Square Meter* (cd/m^2) varies from approximately $10^{-5}cd/m^2$ which corresponds to weak light sources such as “starlight" [19] to $10^5cd/m^2$ emitted from the sun light [20]. This represents almost 9 orders of magnitude. This means that luminance values of typical indoor and outdoor scenes in the real-world are located in this specific range.

Accordingly, a perfect digital imaging system is capable of reproducing the full extent of the dynamic range of a scene. This is however not the case in real-world scenarios, where the depicted dynamic range is directly influenced by the hardware capabilities of the capturing device (camera sensors and optics), and more specifically its dynamic range span. In fact, the majority of capturing devices is only able to represent a limited portion of the dynamic range of the luminance corresponding to the photographed real-world scene. Note that the dynamic range in the context of digital photography is measured in *f-stop*, where for example 10 f-stops correspond to a dynamic range ratio of $1024:1$ (2^{10}) [20] . Accordingly, average capturing devices such as smartphone cameras are capable of representing 5 to 7 f-stops, while in comparison, the *Human Visual System* (HVS) perceives almost 10 to 14 f-stops [21]. There exist however high-end Digital Single Lens Reflex (DSLR) cameras which extend beyond the $5-7$ f-stops of average devices.

Subsequently, the rendering of indoor or outdoor scenes(or a combination of both) using such devices suggests that the actual dynamic range of the scene will be compressed according to the hardware capabilities of the camera. Fig. 1.1 shows an example of the previously discussed compression of the scene dynamic range. The shown scene is a combination of indoor parts (room details) with low light emission, thus low luminance values, and outdoor parts (window area) with relatively high luminance values. Accordingly, the large dynamic range inherent to the scene poses a challenge for common capturing devices, especially smartphone cameras. Depending on the exposure settings of the camera, the resulting image will be either under-exposed (left image in Fig. 1.1) or over-



Under-exposed (Dark) **Over-exposed (Bright)**

Figure 1.1: Example of 2 images depicting the same scene. The left image is *under-exposed*, which means that it was captured using exposure settings allowing for a limited amount of light to to hit the image sensor. On the other hand, the right image is *over-exposed*. The variable exposure settings between both images explain the difference in terms of colors as well as the amount of depicted scene details. Images courtesy of [1].

exposed (right image in Fig. 1.1). The term *exposure settings* refers to an aggregation of several settings including the *shutter speed* (also known as exposure time) which controls the span of exposure of the camera sensor to light, the *aperture* which directly influences the amount of light received by the camera sensor, and to a lesser extent the ISO setting which controls the sensitivity of the sensor.

Evidently, under-exposure corresponds to a set of settings allowing for a limited amount of light to be captured by the image sensor. This can be achieved either by decreasing the shutter speed (exposure time) and/or narrowing the aperture of the camera. As a consequence, the camera sensor will be briefly exposed to a reduced amount of light emitted by the scene, thus the *dark* aspect of the image, as illustrated in image 1.1. Under-exposure enables therefore the capturing of objects and scene parts with high luminance value, such as the window area. However, the details corresponding to regions with low light emission (e.g. the details of the indoor part) are lost.

In contrast to under-exposure, over-exposure represents the case where the exposure settings of the camera allow for an increased amount of light over an extended duration to be captured by the digital sensor. Consequently, the detection of light emitted from objects with low luminance values is enhanced, such that their details are visible in the final image. On the other hand, areas with high luminance values scenes will be totally saturated due to the extensive amount of light received from these regions. This can be explained by the fact that the dynamic range of the camera sensor is unable of representing both sets of details simultaneously. Accordingly, images similar to the ones shown in Fig. 1.1 are labeled as **Low-Dynamic Range (LDR)** images.

One possible way of expanding the dynamic range of capturing devices is to focus on the hardware aspect of the problem. Accordingly, the aim of such approaches is to develop a CMOS image sensor capable of representing as much dynamic range of the scene as possible. The field of HDR CMOS sensors is well investigated, with multiple prototypes already implemented on actual cameras. As thoroughly explained in [22, 23], several approaches have been already examined, most notably *spatially varying pixel exposures* [24, 25] where patterns with varying exposure settings are overlaid on top of the image sensor, *Time-to-Saturation* methods [26, 27] where the actual photocurrent resulting from the conversion of the detected light photons is estimated based on the measured saturation time at each pixel [28], or *Multiple Capture* approaches [29, 30] where several images captured using varying exposure times are combined at the sensor level.

Although HDR CMOS sensors generally perform well in terms of dynamic range expansion, they are almost exclusively deployed on professional and high-end DSLR cameras, mostly due to their relatively high development costs. Furthermore, performance issues related to noise, limited spatial resolution or decreased frame-rate must be tackled before integrating such sensors into the market of average cameras. Accordingly, for devices like smartphone or tablet computer cameras, software-based solutions for the expansion of the dynamic range are favored. In this context, the ultimate goal of *High Dynamic Range Imaging* (HDRI) is to artificially expand the dynamic range contained in a set of low dynamic range LDR images depicting the same scene. These LDR images are captured using different exposure settings (different exposure times) ranging typically from under-exposure (dark) to over-exposure (bright). Consequently, each single LDR of the input stack represents a different range of details, according to its exposure time. Therefore, the final HDR image contains typically all possible details available in the input LDR images. Understandably, a crucial assumption to this approach concerns the fact that these LDR images have to be aligned, in the sense that they depict the exact same content. The number of input LDR images depends of the actual application and the available computational power of the enabling system. In case a single LDR image is available, the task of expanding its dynamic range is called *Single-Image HDRI*. Evidently, the more LDR images available for the HDR rendering, the wider would be the dynamic range of the resulting image. In the following, we will focus on cases where at least 2 differently exposed LDR images are available.

In the next sections, we will go through the steps of two main methods for HDR rendering and talk about about the limitations related to this rising field of research.

1.1 High Dynamic Range Imaging

As mentioned previously, the underlying idea of HDRI is to combine differently exposed LDR images into a single HDR. The resulting image disposes of an enhanced contrast ratio between the darkest and the brightest regions of the scene, so that an improved representation of the details is possible. There exist however several approaches for effectively combining the set of differently exposed LDR images. In following, we will discuss two fundamental methods, namely *Exposure Bracketing* and *Exposure Fusion*.

1.1.1 Exposure Bracketing

Exposure Bracketing is a well-known approach for combining the set of differently exposed LDR images. The basic concept behind exposure bracketing is to estimate the *Camera Response Function* (CRF) together with its inverse. In fact, the image formation pipeline of most digital cameras is composed of a linear part and a subsequent non-linear part. As described by Debevec and Malik in [31], the incident light expressed throughout the *Radiance value* encounters linear transformations mainly influenced by the camera's physical parameters such as the lens's focal length and the corresponding aperture. This set of linear transforms results in the scene *irradiance value*. The final output of the camera pipeline, namely the pixel intensity values, are obtained by applying a set of non-linear transformations related to the electronics of the cameras. This includes the camera sensor, shutter, A/D converters and various other components [31]. Accordingly, and as suggested in [31, 32], the compression of the actual dynamic range occurs in this phase. Consequently, reversing these non-linear transformations enables the recovery of the scene's actual irradiance, and hence the full extent of dynamic range. In this context, the set of non-linear transformations is summarized into the CRF. This implies that estimating the CRF, and most importantly its inverse function (since we are interested in estimating the scene irradiance from the observed pixel values) plays a central role in exposure bracketing.

The topic of CRF estimation is in turn well-covered. Several approaches were proposed for the accurate modeling and estimation of the non-linearity applied to the scene irradiance values and summarized into the CRF [33, 34, 35]. Obviously, the estimation of the CRF and its inverse is based on the set of differently exposed LDR images, where prior knowledge of the corresponding exposure time enables the reverse transformation to the irradiance domain. For this purpose, Debevec and Malik propose in [31] to take the problem of CRF estimation to the logarithmic domain, in order to facilitate and accelerate the estimation procedure. Alternatively, Mitsunaga and Nayar introduced in [32] a different approach, where the CRF and its inverse are modeled as *Polynomial* functions. Accordingly, the recovery of the CRF is directly linked to the efficient estimation of the parameters of the polynomial functions.

Once the CRF and its inverse are estimated, all input LDR images are transformed back to the irradiance domain, according to their respective exposure times. This results in an *irrandiance map* for each input LDR. Next, the irradiance maps are merged so that a final combined irrandiance map of the scene is gained. The final irradiance map represents in fact the HDR image of the scene depicted by the input LDR images. There exist several strategies in order to effectively merge the recovered irradiance maps. Finally, a *Tone Mapping* step is applied on the combined irradiance map (HDR image of the scene). This step enables the display of the HDR image on LDR monitor or screens. Consequently, tone mapping can be considered as the opposite operation to HDRI, where the actual dynamic range of the HDR image is reduced so that it fits the dynamic range of the display device. However, the aim of most tone mapping methods is to reduce the dynamic range while maintaining a good balance in terms of details. in [36], Yoshida *et al.* conducted a subjective evaluation of several tone mapping techniques.

As explained in this section, the performance of exposure bracketing is strongly correlated with the accuracy of the CRF estimation step. High complexity approaches for CRF estimation significantly increase the computational cost of the HDRI system, especially for devices with limited dedicated computational power. Furthermore, a low number of input LDR images, as it is the case for smartphone cameras where the dynamic range expansion is based on 2 or 3 LDR images, impacts negatively the accuracy of the estimated CRF. In such cases, the visual quality of the final HDR image will be limited. Finally, exposure bracketing requires a prior knowledge of the camera parameters, and most importantly the exposure times of the input LDR images (or in some cases the corresponding exposure ratio).

1.1.2 Exposure Fusion

In comparison to exposure bracketing, *Exposure Fusion* (EF) is relatively recent. EF was first introduced by Mertens *et al.* in [2]. The basic idea behind EF is to fuse the differently exposed LDR images based on 3 color-related quality measures, namely *Contrast*, *Saturation* and *Well-exposedness*. Accordingly, from each input LDR image, 3 maps based on the previously mentioned quality measures are generated. Note that EF does not include a reverse transformation to the *irradiance* domain space, as opposed to exposure bracketing. The irradiance domain is considered to be the actual HDR representation of the scene. However, EF aims at rendering a final image where the full range of the scene details can be seen. Accordingly, the overall quality in terms of contrast and saturation is improved, so that the generated image presents a clearly larger dynamic range than the input LDR images. Consequently, the final input image can be fairly labeled as a HDR image.

As described in [2], the *contrast* map is generated by applying a *Laplacian Filter* and computing the absolute value of the filter response (on the grayscale version of each LDR). The contrast map assigns therefore higher weight to significant image details such as objects details and borders [2]. On the other hand, the *saturation* map gives a score to the saturation of the color channels of each pixel, based on the corresponding *Standard Deviation* value. Finally, the *Well-exposedness* map contains a measure on the intensity of each pixel, and namely how close this intensity is to 0.5 (or 127 in case the pixel intensities vary between 0 and 255).

Using the 3 generated quality measure maps, a combined weight map is computed for each input image, based on the following Eq. and as described in [2]

$$W_k(i,j) = C_k^{\omega_c}(i,j) \times S_k^{\omega_s}(i,j) \times E_k^{\omega_e}(i,j), \tag{1.1}$$

where $W_k(i,j)$ is the final weight assigned to the pixel with the coordinates (i,j) in the k^{th} LDR image. C, S and E represent respectively the contrast, saturation and well-exposedness maps of image k, and ω_c, ω_s and ω_e describe the assigned power values which control the impact of each quality measure on the final weight map.

Next, the computed weighting maps are used for the purpose of fusing the input LDR images into the final HDR image. The fusion step is done using a multi-scale approach using a *Laplacian Pyramid*-based decomposition of the input LDR images. This ensures a smooth merging of the weighted pixels intensities of the input images, so that the final HDR image is artifact-free. An example of the resulting HDR image using EF is shown in Fig. 1.2. In this case, the input stack of input LDR images is composed of 5 images.

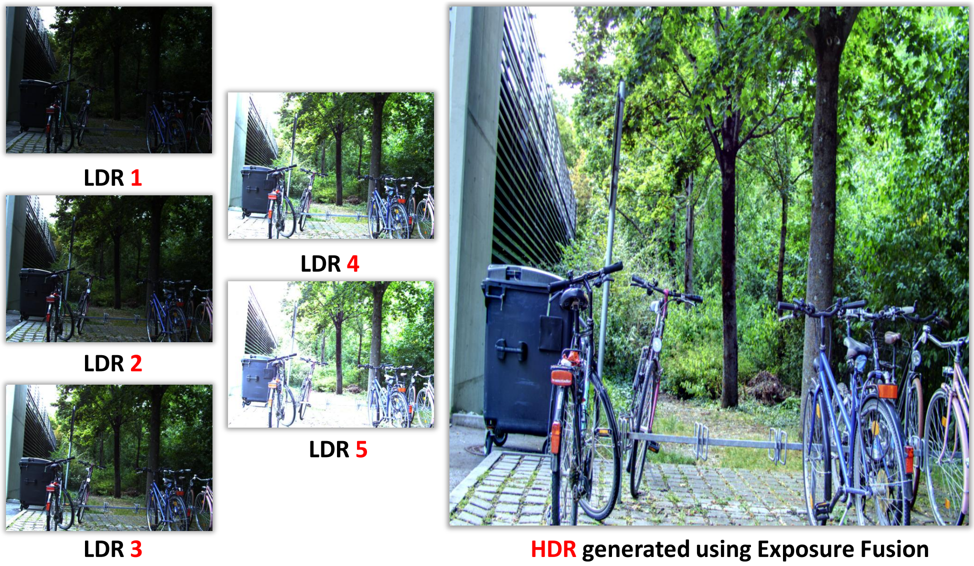


Figure 1.2: HDR result generated from the merging of 5 differently exposed LDR images and using the *Exposure Fusion* [2] algorithm.

EF offers several advantages in comparison to exposure bracketing. The relative simplicity of the enabling algorithm makes EF more suitable for smartphone cameras, where low-complexity HDRI algorithms are preferred. Furthermore, EF does not depend on the estimation of the CRF and its inverse, as opposed to exposure bracketing. This results in a significant increase in the stability of the proposed technique for HDRI, mostly in terms of HDR quality. In fact, performance stability is a critical feature for capturing devices developed for the consumer market. In addition, EF does not require any prior knowledge concerning the exposure times of the set of input LDR images. This represents a valuable feature, as such prior information is either not available or not accurate enough to use it for further processing. Finally, the special attention given to specific image quality measures such as contrast and saturation ensures a very balanced final HDR in terms of color contrast and vividness.

Considering these points, EF is nowadays the method of choice for rendering an HDR image from a set of differently exposed LDR images. The proposed approaches in this work are based on the EF algorithm, mostly since it enables the usage of input LDR images, where no information about their corresponding exposure settings or at least exposure ratio are available.

In the next section, we will examine the limitations of exposure bracketing and EF, especially in the context of dynamic scenes.

1.2 HDRI on Dynamic Scenes

So far, the basic idea of using several LDR images in order to extend their dynamic range is based on the assumption that these images are **aligned**. In the remaining parts of this work, we will refer to this assumption as the *alignment assumption*.

Accordingly, the alignment assumption implies that these images depict the exact same pixel-wise content. This important feature of the set of input images is insured only when the depicted scene as well as the camera used to capture them are both **static**. In this context, a scene is static only if there exist no differences between the captured LDR images. These differences are usually caused by moving objects inside the scenes, so that these objects appear in different locations over the set of input LDR images. In addition, the camera itself needs to be static over the period of capture.

However, the serial nature of the capturing procedure of the input LDR images makes the alignment assumption invalid, as motion related to *dynamic* objects or the capturing device will almost certainly occur in real-world scenes. As a result, the content depicted by the set of LDR images is different. Logically, applying exposure bracketing or EF on misaligned LDR images results in visible artifacts in the rendered HDR image. The most notable artifact is known as the *Ghost effect*, where multiple instances of a dynamic object appear in different locations in the final HDR image, thus giving it a ghost-like effect. An example of the motion-induced *Ghost effect* is shown in Fig. 1.3. The person moving in the scene is changing his position and shape from frame to frame (non-rigid motion). Consequently, the final HDR contains the previously described ghost effect, which results in a decrease of its perceptual quality.

One possible alternative to deal with scene-related motion is to make the shift from *serial* to *parallel* capturing modes. Parallel capturing implies that the LDR pictures are taken simultaneously in time, using several cameras with different exposure settings. The most notable setup in this context is the *stereo capturing setup*, where 2 cameras are mounted and configured to yield 2 differently exposed LDR images of the scene. However, the captured images present content differences caused the perspective shift inherent to the stereo setup. Accordingly, this type of content difference can be categorized as *camera-related motion*, where theoretically a single camera was used to make the first capture, and then was moved to location of the second stereo camera to make the second image.

On the other hand, rendering an artifact-free HDR of dynamic scenes has been thor-

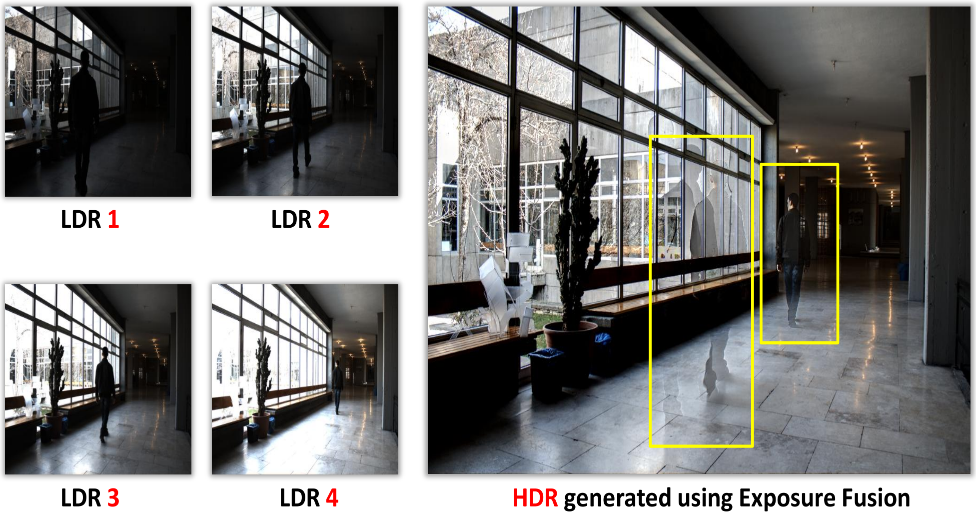


Figure 1.3: Example of an HDR image with visible *Ghost* artifacts (areas highlighted by the yellow boxes) caused by the moving person in the set of 4 input LDR images. As a result, 4 ghost-like instances of the same person appear in the final HDR image rendered using EF [2]. Input images are courtesy of [3].

oughly investigated by the computer vision community. Several approaches claim to successfully handle the misalignment and the associated inconsistencies, so that the final HDR is ghost- and blur-free. The sphere of these methods can be split into two major categories, namely *de-ghosting approaches* and *correspondences-based methods*. In the following sections, we will examine state-of-the-art approaches relevant to our work.

1.2.1 State-of-the-Art De-ghosting Approaches

The first category of approaches for correcting the artifacts related to dynamic objects falls under the scope of the *De-ghosting* methods. The idea behind these approaches is to detect inconsistencies belonging to dynamic pixels, compared to a reference LDR image selected from of the input stack. These methods usually assume that the camera is static or rather propose a global registration step to compensate for the misalignment. In general, the final merging procedure excludes dynamic regions and inconsistencies from the final HDR image

In [37], Khan *et al.* propose a technique based on the estimation of the likelihood that a pixel belongs to a moving object. This method assumes that the majority of input LDR images depict the common parts of the scene, namely the background. The algorithm iteratively weights the contribution of each pixel of the input stack to the background region (static region), creating therefore a non-parametric representation of the static parts of the scene.

Similarly, Galo *et al.* describe in [6] a ghost removal approach which renders a HDR version of a selected reference LDR image from the stack of available exposures, unlike in [37] where the final HDR is composed of different regions from the input images. As explained in [6], the selection of the reference image implies that the final HDR is consistent, thus does not contain any *duplicated* objects. The selection step assesses the amount of saturated pixels of each input LDR image and picks the images with the least number of these pixels. Next, the proposed method takes into consideration the *Photometric relation* for the purpose of detecting dynamic pixels in the input stack. The photometric relation implies that the intensity $I_{\Delta t_1}$ of a static pixel in an image captured with the exposure time Δt_1 is smaller than its intensity $I_{\Delta t_2}$ in a second image captured with the exposure time Δt_2, provided that $\Delta t_1 < \Delta t_2$. The photometric relation is not valid in case of saturated pixels (pixels intensities close to 0 or 1) and generally when changes in the scene due to motion occur.

Likewise, Jaehyun *et al.* describe in [38] a method, which is in turn based on the photometric relation and composed of two main stages. In the first stage, the algorithm assesses the number of pixels which do not satisfy the photometric relation in every patch in the input stack. If a patch contains at least 1 of such pixel, it will be consequently discarded from the weighting map during the final merging using EF. The second stage aims at removing probable outliers, by evaluating the resemblance of each patch towards the corresponding patch in the reference image. This evaluation is based on the *zero Mean Normalized Cross Correlation* metric.

A similar approach is proposed by Pece *et al.* in [39]. The described algorithm starts by generating the *Median Threshold Bitmaps (MTB)* of the input images. The MTB maps were initially introduced by Ward *et al.* in [40] for the purpose of creating a common representation for differently exposed images (representation independent on the illumination conditions of the images). If all input LDR images represent are aligned (no camera- or scene-related motion), the MTB binary maps would be identical. This assumption is used for the purpose of detecting probable dynamic pixels, which results in an initial binary motion map. Using morphological operations such as *Dilation* and *Erosion*, the initial motion map is enhanced and subsequently transformed into a cluster map. Finally, the algorithm seeks to find, for each motion region in the labeled motion map, the sub-group of input images, which are similar up to a certain threshold. This information is included into the merging operation of the Exposure Fusion algorithm in order to render the final HDR.

These methods perform generally well when the input stack offers a large number of differently exposed LDR images. However, they depend strongly on the motion detection step, where assumptions such as the photometric relation do not always hold. In addition, these methods generally fail in case of 2 input LDR images with large illumination difference.

More recently, An *et al.* describe in [5] a novel approach for generating binary motion maps and integrating them into the weighting maps of the exposure fusion step. The proposed method starts with the selection of the reference image. The reference frame is identified as the image from the input stack which contains the largest well-contrasted parts. Next, the authors suggest to build binary motion maps for each non-reference image, respectively for the *low-contrasted* regions (regions with less texture) and *high-contrasted* regions. Both motion maps are combined together to form the final binary *Ghost Map* [5] for each non-reference image, which will be subsequently included into the merging procedure of the exposure fusion algorithm. The underlying idea for the detection of dynamic pixels in high-contrasted regions is to create the Zero Mean Normalized Cross-Correlation (ZNCC) representation for each non-reference image and to model the distribution of these regions in the ZNCC images as a mixture of Gaussian functions, which parameters are to be estimated. On the other hand, the detection of dynamic objects in low-contrasted regions is based mostly on the photometric relation.

Despite the high quality of the de-ghosting operation, the generation of the ZNCC maps is a computationally expensive task, as shown later. In addition, the accuracy of the binary ghost maps (motion maps) is limited in case of few input LDR images.

1.2.2 State-of-the-Art Correspondence-based Approaches

The second category of approaches dealing with dynamic objects is composed of approaches relying on correspondences (sparse or dense) in order to align the input LDR images. In this context, alignment can be either *spatial* where the non-reference LDR images are warped to the view of the selected reference LDR, or *color-related* by aligning the reference LDR to each non-reference LDR image separately in terms of colors (color mapping). In both cases, the goal is to reproduce a stack of aligned but differently exposed LDR images.

In [41], Kang *et al.* propose a technique based on a global alignment operation followed by a refinement step using local optical flow. Although this method presents clear advantages over conventional de-ghosting approaches, the lack of a global optical flow estimation affects the accuracy of the final motion vectors, especially in flat regions. In [42], Zimmer *et al.* propose a joint framework for *Super-Resolution* (SR) and HDR, by aligning all images to the reference using *Optical Flow* (OF). The described approach gets around the issue of color inconsistency for OF by including a gradient constancy assumption in the data term of the energy function. Alternatively, Sen *et al.* describe in [43] a solution for simultaneous HDR reconstruction and LDR alignment using a joint patch-based minimization framework. The alignment is based on a modified version of the *PatchMatch* (PM) [44] algorithm. The final HDR is rendered from the well-exposed regions of the reference LDR and from the remaining stack of LDR images for low-exposed regions in the reference image.

Likewise, Hu *et al.* propose in [15] to align every non-reference LDR image to the selected reference LDR image, which typically has the highest number of well-exposed pixels. The patch-based alignment approach uses the generalized PM algorithm for well-exposed patches in the reference LDR and suggests an additional modification to PM for over- or under-exposed patches in the reference image. The final HDR is composed using the EF algorithm. As demonstrated in [15], the approach of Hu *et al.* is considered to be the state-of-the-art approach for correcting misalignment due to moving objects in the context of HDRI.

Recently, Gallo *el al.* proposed in [45] an approach based on the matching of sparse feature points between the designated reference and non-reference images. The *matcher* developed for this purpose is robust towards saturation. Once a dense flow field is interpolated, the warped images and the reference LDR are merged using a modified *Exposure Fusion* (EF) algorithm, which minimizes the effects of faulty alignment.

These methods usually achieve accurate alignment results, which in turn helps creating an artifact-free final HDR image. However, the common drawback of these methods is related to the computational cost of the enabling algorithm. This fact hinders the deployment of such approaches on devices with limited computational resources such as smartphones. In addition, smaller stacks of input LDR images with significant exposure and scene differences (due to large motion) hampers the generation of an artifact-free HDR.

1.3 Goals and Structure of the Thesis

Taking into account the previously described state-of-the-art approaches, there still exist the need to find the best possible compromise between the computational complexity of the proposed solution and the quality of the final HDR image. In this context, the main goals of this work are:

- Develop an HDR framework for dynamic scenes, which typically suits capturing devices with limited computational power, such as smartphone cameras. This implies that we seek to develop and test low-complexity algorithms capable of handling several LDR images with scene- or camera-related motion, in order to render an artifact-free HDR image of the corresponding scene.

- We are interested as well in achieving high quality HDR rendering results, while making sure that the complexity of the enabling algorithm remains low. The quality of the rendered results can be described in terms of freedom from noise and motion-induced artifacts as well as the perceptual quality of the HDR image.

- Considering the fact that we focus on devices with limited computational resources, the proposed solutions have to account for extreme scenarios. This includes cases where the number of differently exposed LDR images is limited to 2 images. Additionally, cases where the exposure ratio (and hence the color difference) between the input LDR images is large are considered to be very challenging for the majority of existing state-of-the-art methods. However, such cases are frequent when using smartphone cameras and more specifically when dealing with outdoor scenes.

- We seek as well to examine the generalization performance of our solutions. Accordingly, our approaches must yield high de-ghosting and HDR rendering results, irrespective of the nature of the scene (for example indoor or outdoor) or the type of the depicted motion. Furthermore, the nonexistence of prior knowledge about the settings and parameters of the capturing device must not affect the performance of the proposed solutions.

Based on these points, we start by examining a low-complexity approach for HDR de-ghosting in Chapter 2. The proposed method is based on the detection of motion inside the input LDR images and the modification of the EF algorithm in order to exclude the detected dynamic objects from the final HDR image. Based on the evaluation of the proposed approach, we investigate in Chapter 3 the possibility to improve the quality of the *Histogram Matching* (HM) algorithm [4], which represents an essential building-block of the proposed de-ghosting approach. In addition, we evaluate the impact of the manipulation of HM on the de-ghosting performance. In Chapter 4, we examine a novel color mapping framework in order to propose a better performing alternative to HM. The proposed framework is based on *Convolutional Neural Networks* (CNN). Finally, and based on the evaluation of the color mapping approach in Chapter 4, we propose in Chapter 5 to adapt the CNN-based framework to our target application, namely HDR rendering on dynamic scenes. We propose a novel end-to-end CNN-based approach, which renders an artifact-free HDR image from 2 differently exposed LDR images with large content difference. The experimental part of Chapter 5 shows that the proposed solution fits the previously discussed requirements and goals.

2 HDR De-ghosting

As explained in the previous section, rendering an HDR image based on real-world LDR images is a challenging task to tackle especially when the device at hand disposes of limited computational power such as smartphone devices. In these particular circumstances, dealing with the ghost-effect artifacts caused by camera- or scene-related motion requires a straightforward method where the enabling algorithm fits into the available computational resources. This consequently excludes computationally expensive approaches which aim at realigning the input LDR images to a designated reference image (correspondence-based methods), using several motion estimation and correction techniques such as *Optical Flow* [46, 47, 48, 49], *Block Matching* [50, 51] and various other methods.

On the other hand, *De-ghosting* approaches offer the perfect trade-off between computational efficiency and quality of the final HDR image. The main focus of most HDR de-ghosting techniques evolves around the detection of the ghost-inducing motion objects as well as their exclusion from the final HDR image. The latter stage depends on the method used for generating the HDR image, namely whether the algorithm renders the radiance map of the scene and applies tone mapping as a final step (exposure bracketing), or merges the input LDR images directly using EF. However, the main idea behind existing HDR de-ghosting approaches is to suppress the effect of motion pixels by decreasing their assigned weights during the merging procedure, or by excluding them completely.

With this in mind, our principal goal is to develop a de-ghosting algorithm which fits the previously mentioned quality requirements, especially in case of a limited number of input LDR images, while making sure that the computational complexity of the enabling algorithm remains low. In this context, we propose a novel HDR de-ghosting approach based on the detection of dynamic objects in the scene (also called motion-related objects), using for this purpose a selected reference LDR image from the input stack. Based on the comparison of each non-reference image to the selected reference, we create a set of *Binary Motion Maps*, where each map contains the locations of dynamic objects in the corresponding non-reference image. Finally, we include these maps into the weighting stage of the EF algorithm as proposed in [5], so that their impact on the final HDR image is reduced. As explained in the next sections, the proposed method puts no restrictions on the number of input images, as well as the exposure difference between them. An earlier version of this de-ghosting technique is outlined in [52].

2.1 Proposed Approach

In this section we provide a detailed description of the proposed de-ghosting approach based on motion detection and modification of the EF algorithm for the case of 2 input LDR images I_b (over-exposed bright image) and I_d (under-exposed dark image). The building blocks of our approach are shown in Fig. 2.1. Section 2.1.3 contains the implementation details for the case where more than 2 input LDR images are available. In the following,

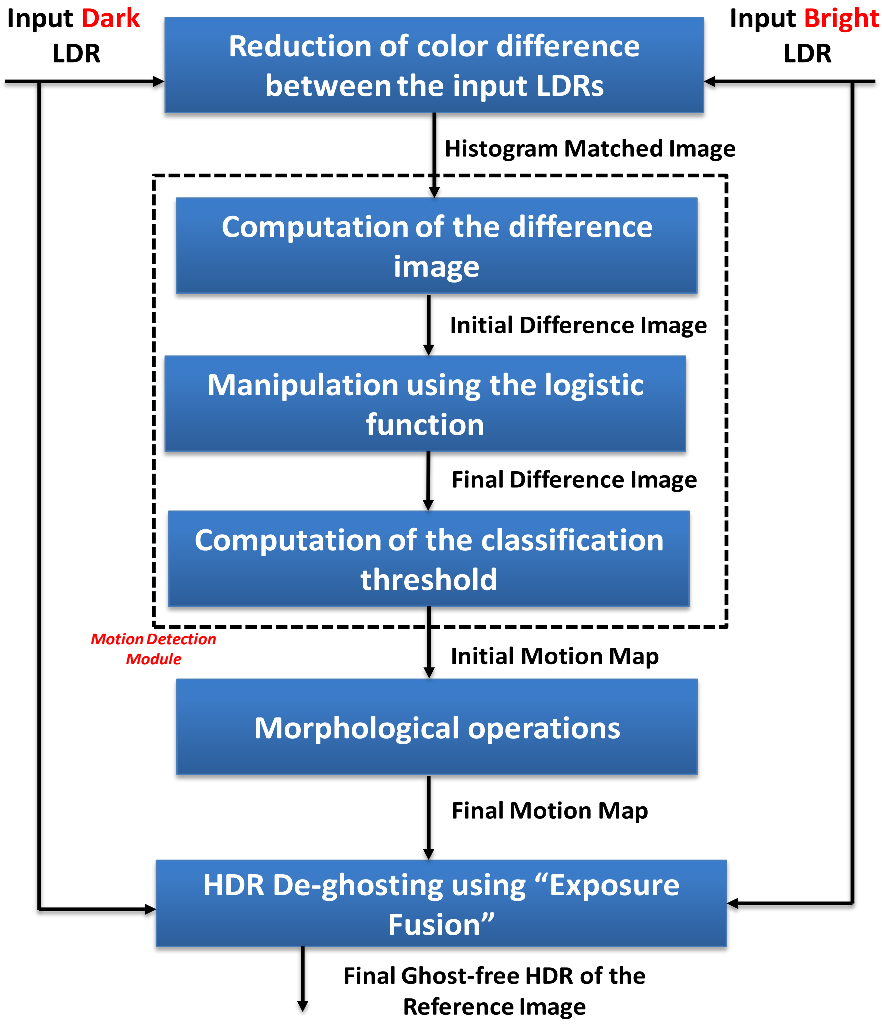


Figure 2.1: Illustration of the different stages of the proposed HDR de-ghosting approach.

we go through the main stages of our algorithm, as described in Fig. 2.1.

2.1.1 Motion Detection

The underlying idea of the proposed motion detection algorithm is to explore the **difference image** between the input bright image I_b and dark image I_d (in case of 2 input images). Evidently, one of the input images is selected as the reference frame. The selection of the reference image is either executed automatically using one of the several existing approaches such as in [37, 6, 38, 39, 5], or relinquished to the user.

The main purpose of the motion detection step is to analyze the information provided by the difference image between both input LDR images, in order to accurately classify each pixel of the non-reference image. Two categories can be identified:

- **Static Pixels**: This class corresponds to static regions of the captured scene. Pixels which belong to this class are consistent over the set of input LDR images.
- **Dynamic Pixels**: This class describes motion-related regions of the non-reference image. Understandably, the focus of the motion detection stage is to accurately detect these pixels.

The classification of each pixel in the non-reference LDR image requires the computation of a classification threshold, which accurately assigns each pixel to the corresponding class based on its difference value (or values, in case of RGB images). In the following sections, we provide a detailed description of the necessary pre-processing as well as threshold computation steps.

Reduction of Color and Content Differences

The inherent color difference between the input images caused by the varying exposure times puts a strain on the accurate calculation of the classification threshold. This implies that a reliable motion detection procedure must include an initial step for the reduction of the color difference between the reference image and non-reference image.

To deal with this particular task, we propose a *color mapping* step in order to decrease the color difference between I_b and I_d. Color mapping is a well-researched topic in image processing which focuses on transferring the color properties of a *target image* to a *source image*, thus decreasing the color difference between both images. There exist several methods which propose different perspectives on how to deal with this problem with varying computational complexity. Logically, we shift our focus to low-complexity algorithms for color mapping, in order to keep the overall computational requirements of the proposed deghosting algorithm as limited as possible. In this context, HM [4] satisfies the constraints of this framework. HM is a straightforward yet efficient method for color mapping between 2 differently exposed images. The main idea behind HM is to match the histogram of the *source image* to the histogram of the *target image*. This is done by computing the *Cumulative Distribution Functions (CDF)*, F_{src} and F_{trg}, of both histograms. Next, for every pixel intensity value $I_s \in [0,255]$, HM computes the corresponding matching intensity I_t, according to the equation

$$F_{src}(I_s) = F_{trg}(I_t). \tag{2.1}$$

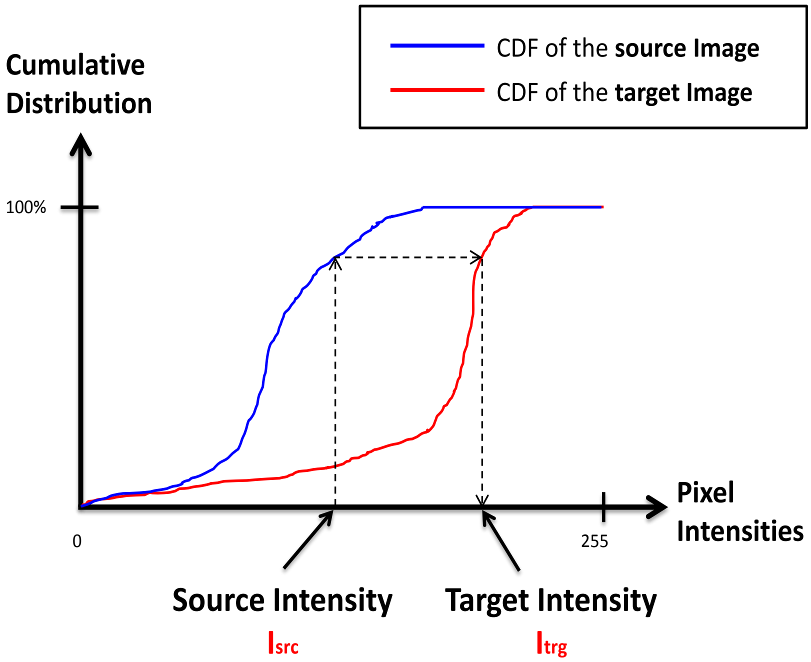


Figure 2.2: Estimation of the matching intensity based on the target *Cumulative Distribution Function* (CDF), according to the HM algorithm [4] and as described in Eq. 2.1.

The matching operation is performed for every color channel separately. The operation described in Eq. 2.1 is graphically illustrated in Fig. 2.2. Furthermore, Fig. 2.3 shows an example of color mapping results using HM.

HM yields good color mapping results in case of relatively moderate color difference between the source and target images. However, artifacts due to a faulty matching procedure might occur when the input images contain saturated (over- or under-exposed) areas, and/or the depicted scenes of the two images are different due to motion or simply different capturing times or scene content.

In our case, we designate I_d as the source image and I_b as the corresponding target image. The choice of this setup is meant to guarantee reliable color mapping results, and evolves from the fact that under-exposed (dark) images generally contain more details than over-exposed images, where over-exposed regions are more frequent.

Moreover, we propose to initially down-sample I_b and I_d prior to the computation of the difference image. The down-sampling allows for the elimination of noise and artifacts caused by the HM step, as we use a *Gaussian Pyramid* which typically acts as a low-pass

Figure 2.3: Example of HM results using 2 differently exposed source and target images. The input images also present content differences due to camera motion.

filter. In addition, the down-sampling step decreases the computational cost of the ensuing manipulation of the difference image and the subsequent estimation of the classification threshold. Empirically, it is sufficient to down-sample the images to 1 or 2 levels.

The color mapping step results in an image I_h, which has the same color properties as I_b while having the same content as I_d. It is important to notice that in the case of 2 input images to the de-ghosting algorithm, the choice of the source and target images does not influence the selection of the reference image used for the motion detection step.

Computation and Manipulation of the Difference Image

As mentioned previously, the detection and subsequent classification of the pixels of the non-reference image is based on the exploration of the distribution of the difference image. In case of 2 input images, the choice of the reference image at this stage is not mandatory, as the gained binary motion map can be assigned to the subsequently selected non-reference image. However, in case of more than 2 images, the selection of the reference image at this stage is necessary, as explained in Section 2.1.3.

In case of 2 input LDR images, the difference image $I1_{diff}$ is computed between the bright image I_b and the histogram matched image I_h according to Eq. 2.2:

$$I1_{diff}(i,j) = |D(I_h(i,j)) - D(I_b(i,j))|, \tag{2.2}$$

where D represents the down-sampling operator and (i, j) the corresponding pixel coordinates.

The ensuing classification step is based on the information provided by the difference image. Typically, 2 types of difference values in image $I1_{diff}$ can be identified:

- Difference values from motion-related objects. These difference values belong to the dynamic pixels which we seek to detect. We assume that these values are *large* and *less frequent*.
- Difference values from static regions. Ideally, these values are equal to 0. However, I_h is an approximation to the bright image I_b, so that the pixel intensities of image I_h might still be different from the intensities of the bright image I_b. In this context, we assume that these difference values are *smaller* than motion-related difference values and *more frequent*.

The goal of the next steps is to accurately distinguish the previously mentioned types of difference values. To this end, we propose to manipulate the difference image $I1_{diff}$ using the *Logistic Function*, according to

$$I2_{diff,c}(i,j) = \frac{1}{1 + k_1 \mathrm{e}^{-k_2(I1_{diff,c}(i,j) - 0.5)}}, \tag{2.3}$$

where k_1 and k_2 are control parameters and c indicates the corresponding color channel. Note that k_2 is fixed to 12. The manipulation using the logistic function aims at improving the quality of the subsequent classification step, by extending the *contrast* of the difference image so that large difference values corresponding to dynamic pixels are enhanced in comparison to smaller difference values. Consequently the range between both types of difference values is increased. This is important for the subsequent detection step, since it allows for an accurate classification. The manipulation of the difference image using the logistic function is performed for every color channel separately.

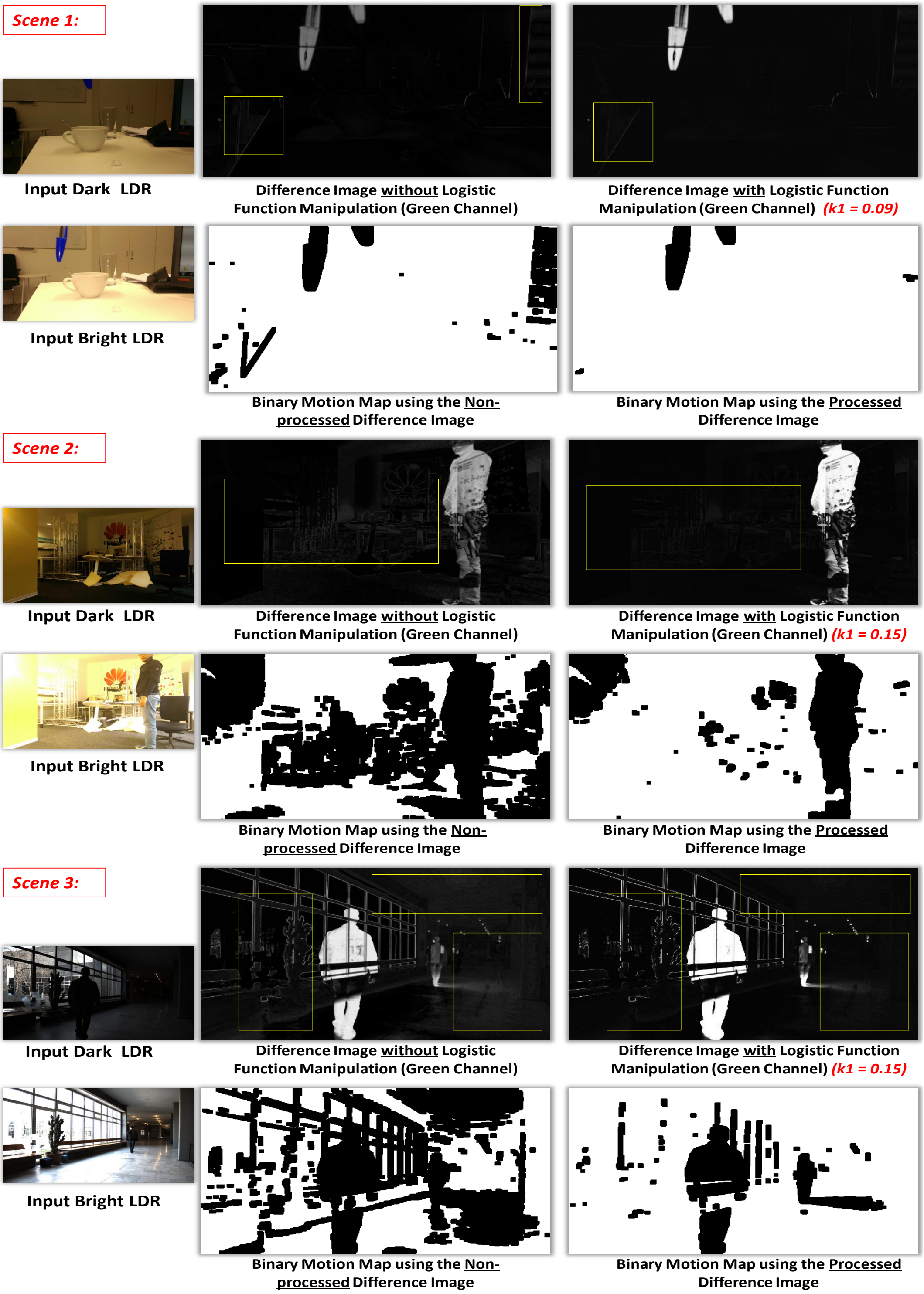


Figure 2.4: Illustration of the improvement brought by the logistic function-based manipulation of the difference image. The resulting binary motion map contains less outliers (wrongly labeled pixels) while accurately detecting dynamic pixels. Images of scene 3 are courtesy of [3].

The results of the manipulation step of the difference image according to Eq. 2.3 are presented in Fig. 2.4. The processed difference images show an increased *contrast* in terms of difference values between dynamic pixels and static pixels. This enhancement enables an effective subsequent classification as presented in the included final binary motion maps. As illustrated in the figure, using $I2_{diff}$ instead of $I1_{diff}$ for the detection step successfully eliminates false positives, which typically belong to static pixels with large difference values caused by inaccurate estimation of the target intensities during the HM step (areas marked by the yellow boxes). On the other hand, dynamic pixels belonging to motion objects are correctly detected, which in turn enhances the quality of the final maps and the shape of the motion objects.

As shown in Fig. 2.4, the control parameter k_1 influences directly the contrast of the difference image $I2_{diff}$ and consequently the accuracy of the resulting binary motion map. This fact is examined in more details in Fig. 2.5, where the difference images of the scenes presented in Fig. 2.4 are manipulated using varying values for the parameter k_1 and compared to the ground-truth binary motion maps. Note that for these particular scenes, the enclosed ground-truth binary maps are in fact a manually-crafted approximation of the true motion maps.

As illustrated in 2.5, increasing the value of the control parameter k_1 generally induces a reduction of the amount of detected outliers (false positives), therefore improves the accuracy of the related binary motion map. This observation can be seen in the regions marked under the red boxes in Fig. 2.5. However, the reduction of outliers might cause the erroneous labeling of correctly detected dynamic pixels as possible false positives, thus causing a slight decrease in the precision of the binary motion map. This observation can be seen in the regions marked by the yellow boxes in 2.5.

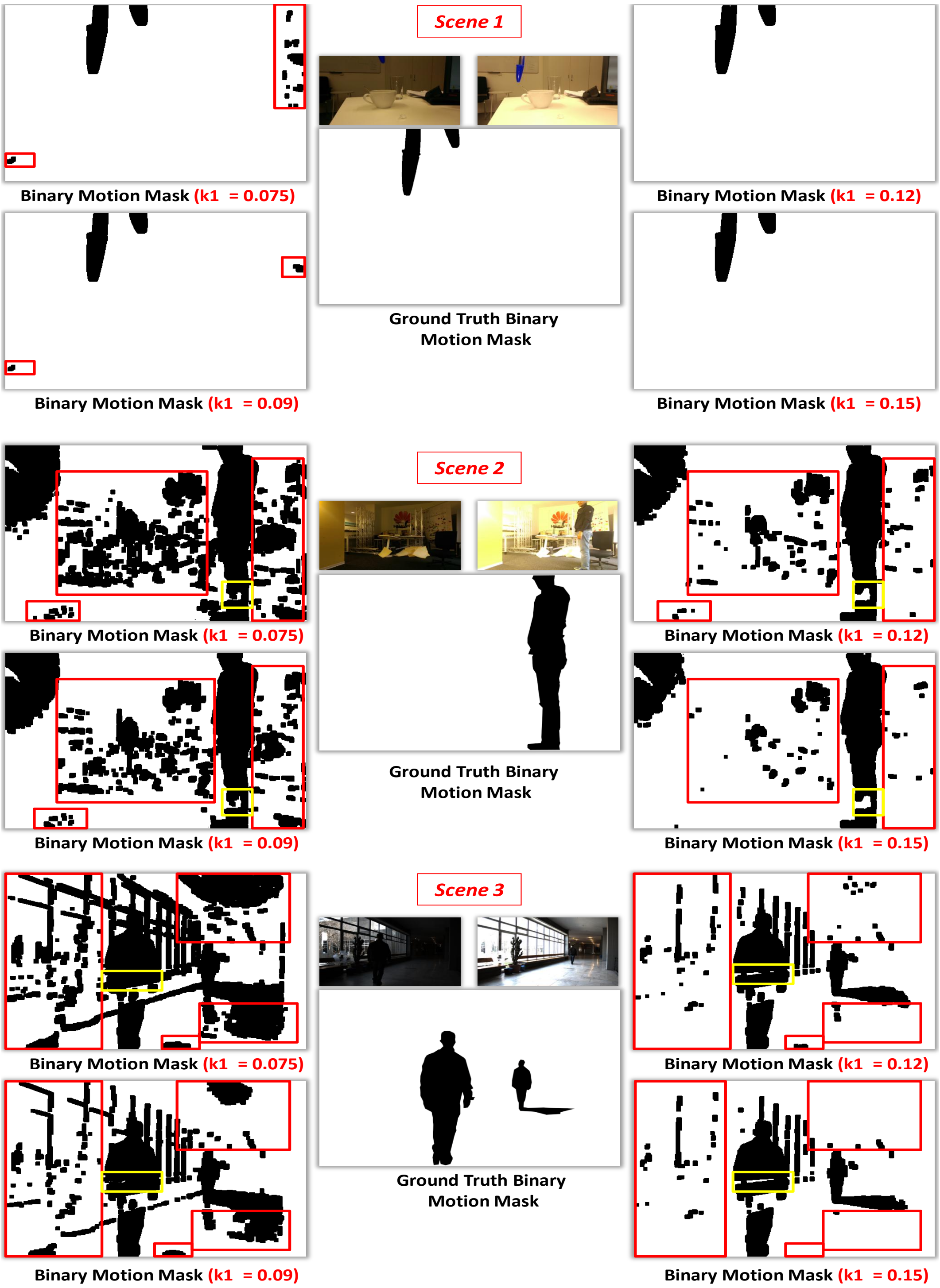


Figure 2.5: Representation of the impact of the parameter k_1 on the accuracy of the final binary map in comparison to the ground-truth map. Areas under the red boxes indicate regions where the manipulation successfully reduces noise related to faulty detection, whereas areas under yellow boxes highlight the regions which are wrongly labeled as false positives. Images of scene 3 are courtesy of [3]

The best-performing value of the parameter k_1 for each scene depends on the nature of the input LDR images, and more specifically on 3 major properties:

- The difference in terms of exposure ratio between the input images. A large initial color difference between the input LDR images impacts the quality of the color mapping step using HM. This increases the number of initially detected false positives which correspond to difference values from static regions large enough to be marked as dynamic. Subsequently, a much higher parameter k_1 is needed to correct this effect. This is the case for scenes 1 and 2 in Figures 2.4 and 2.5.
- Amount of texture in the presented scene. A highly textured scene influences the distribution of the difference image so that an accurate classification of the static and dynamic pixels is more challenging. This again requires a higher k_1 value.
- The nature of the motion in the depicted scene. For example, objects totally disappearing from the scene are generally easier to detect. This means that a smaller value of k_1 is sufficient. This observation applies for scene 1.

Computation of the Classification Threshold

Using the processed difference images $I2_{diff,c}$, we compute the classification threshold T_c for every color channel. We aim at computing a threshold value for each color value which successfully classifies the difference values observed in $I2_{diff,c}$, based on the assumptions made earlier. For this reason, we propose an approach inspired from our work presented in [53]. The proposed approach starts with the computation of the color histograms for each difference image $I2_{diff,c}$. The threshold T_c of color channel c depends on the mean value of the bins where an *abrupt* decrease in the number of pixels is observed. Accordingly, the location of the observed abrupt decrease indicates that almost all static pixels are detected, according to the assumptions discussed earlier. This can be translated into the following equation:

$$T_c = \underset{T_c^i}{\arg\max} \, |N_p(T_c^i) - N_p(T_c^{i+1})|, i = 0, ..., B-2, \tag{2.4}$$

where $N_p(T_c^i)$ is the number of pixels around the bin center T_c^i of the bin number i out of B bins. The threshold T_c is equal to $\frac{T_c^{i+1} - T_c^i}{2}$. Accordingly, a pixel is marked as *dynamic* if at least 1 difference value in $I2_{diff}$ of a color channel c is larger than the corresponding threshold T_c. This results in an initial binary motion map M, which indicates the location of the dynamic pixels. The computation of the threshold T_c is graphically represented in Fig. 2.6.

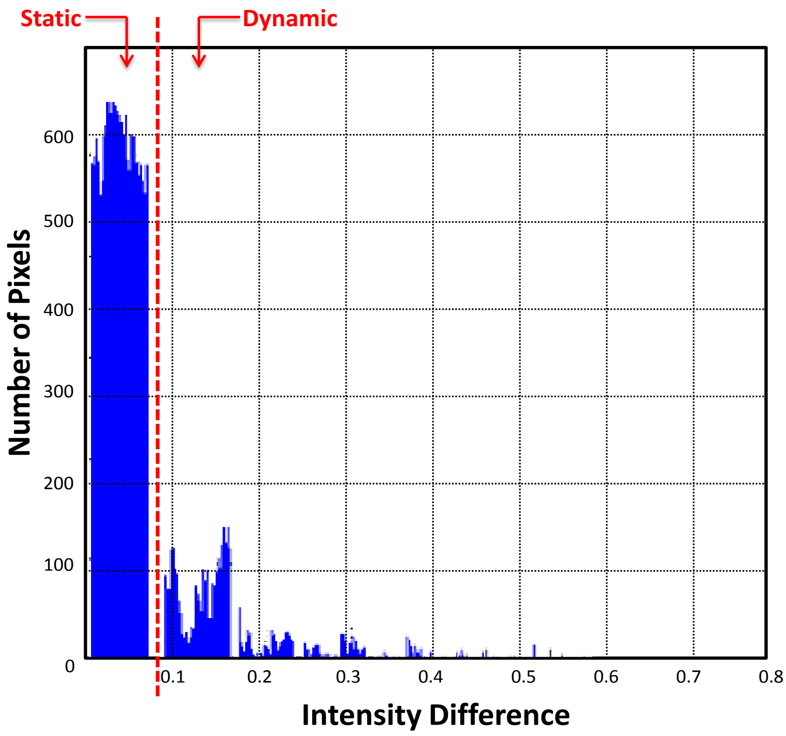


Figure 2.6: Graphical representation of the threshold selection process. The desired threshold value corresponds to the location where the abrupt decrease in the number of pixels occurs.

The generated initial binary motion map M is up-sampled back to the original size of the input images. Furthermore, we propose to process the motion map M using basic morphological operations in order to improve the detection accuracy and especially the shape and filling of the dynamic objects.

Morphological Operations

The morphological operations performed in this step aim at removing possible detection noise (wrongly detected pixels) and enhance the shape and filling of dynamic objects in the final motion map. Evidently, the choice of the proper morphological operations depends on the corresponding computational complexity as well as the quality of the final binary map.

The first operation starts with counting the number of motion pixels N_w inside a window of size $w_1 \times w_1$ centered around each dynamic pixel in the motion map M. The parameter w_1 is typically set to 3 or 5. The processing of the motion pixel under investigation is done as following:

- $N_w \leq \lfloor \frac{w_1^2}{2} \rfloor$: Probable falsely detected pixel. The pixel will be discarded from the final motion map.

- $N_w > \lfloor \frac{w_1^2}{2} \rfloor$: The pixel under investigation is confirmed as dynamic pixel and is kept in the final binary motion map.

The second morphological operation marks areas in the immediate vicinity of dynamic pixels as dynamic as well, using a similar window-based approach as previously described. This step enables to fill-up possible missing dynamic pixels inside motion-related objects, and thus improves their shape in the final motion map $\hat{M}$. Likewise, the corresponding window size w_2 varies between 3×3 or 5×5.

The reduction of wrongly detected pixels and the enhancement of the shape and fill of dynamic objects is shown in Fig 2.7. The proposed morphological operations significantly increase the accuracy of the final binary motion map, even if the initial binary motion map is noisy. Note that one side-effect of the second morphological operation is that it tends to over-fill some objects, therefore making them look wider than in the original images. However, this has no negative impact on the final HDR image, as shown in later results (Fig. 2.9).

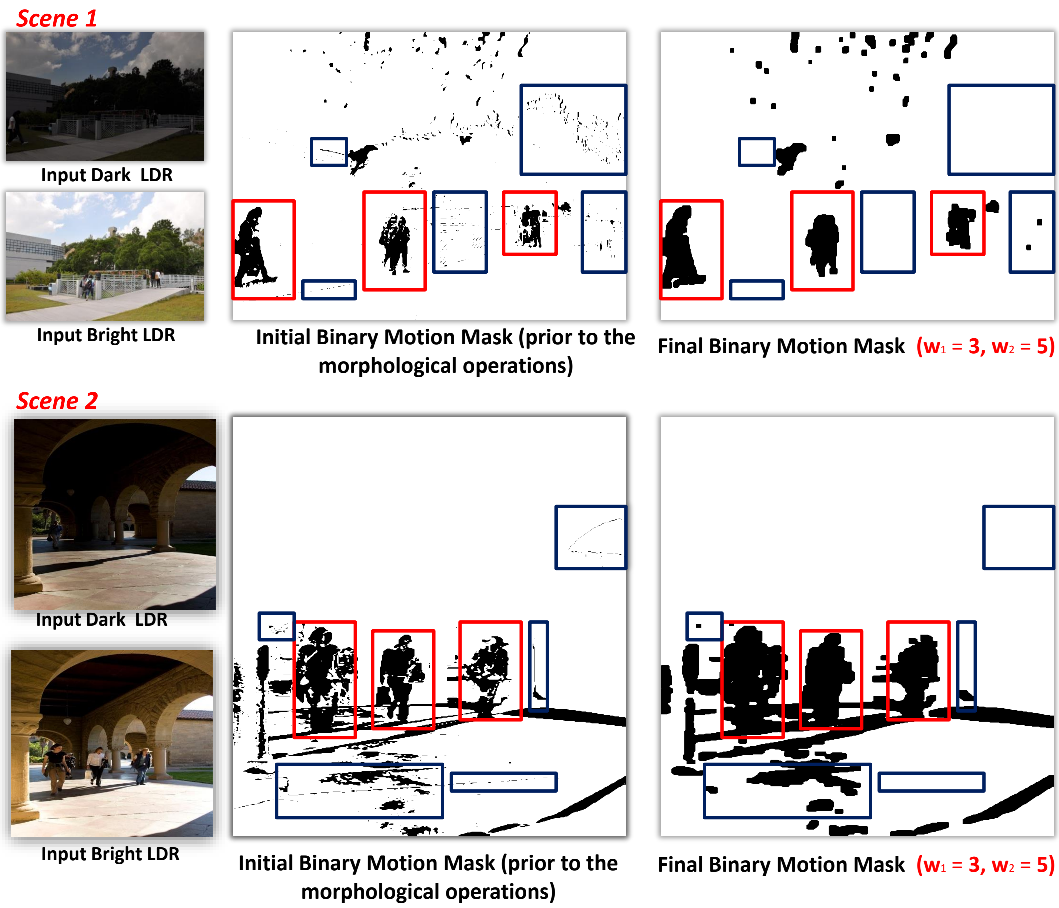


Figure 2.7: Illustration of the impact of the proposed morphological operations. Red boxes highlight the areas where the proposed operations improve the shape of the dynamic objects. Blue boxes indicate the parts where the morphological operations successfully reduced the detection noise. Images from scene 1 are courtesy of [5], while images of the second scene are courtesy of [6]. Note that k_1 is set to 0.1.

Note that the motion map corresponding to the designated reference image is composed of ones, as we assume that all pixels in the reference image are static. Nonetheless, the selection of the reference image is not required at this stage (in case of two images).

2.1.2 HDR De-ghosting

In this section, we discuss the actual de-ghosting step which aims at rendering an artifact-free HDR image. In fact, this step is based on the previous motion detection stage, where the motion information enclosed in the generated binary motion map (or maps, in case of more than 2 input LDR images) is taken into consideration for the purpose of decreasing the effect of dynamic objects in the final HDR image.

The final ghost-free HDR image is rendered using the merging procedure introduced by the EF algorithm. As discussed in earlier sections, the EF algorithm merges the input LDR images using a weighting scheme which assesses each of the input images according to their corresponding *Well-exposedness*, *Saturation* and *Contrast*, according to the equation 2.5:

$$W(p) = C(p)^{\omega_C} \times S(p)^{\omega_S} \times (E(p))^{\omega_E}, \tag{2.5}$$

where $C(p)$, $S_i(p)$ and $E_i(p)$ are respectively the *contrast*, *saturation* and *well-exposedness* maps. The parameters ω_C, ω_S and ω_E represent the corresponding control values, which regulate the influence of each measure on the final weighting map.

In this context, we propose to modify the weighting map in equation 2.5, so that it includes the information contained in the final binary motion map, similar to the approach presented in [5].

The selection of the reference image is imperative at this level, as the manipulation of the weighting maps differs for the reference and the non-reference images. Nonetheless, the weighting values corresponding to dynamic pixels in the reference image are set to 1, since these pixels are excluded from the weighting map of the non-reference image. This can be expressed in the following 2 equations:

$$W_{Ref}(p) = \begin{cases} 1, & \text{if } \hat{M}(p) = 0 \\ C_{Ref}(p)^{\omega_C} \times S_{Ref}(p)^{\omega_S} \times E_{Ref}(p)^{\omega_E}, & \text{otherwise} \end{cases} \tag{2.6}$$

$$W_{NonRef}(p) = C_{NonRef}(p)^{\omega_C} \times S_{NonRef}(p)^{\omega_S} \times E_{NonRef}(p)^{\omega_E} \times \hat{M}(p), \tag{2.7}$$

where $\hat{M}(p)$ is the previously computed binary map. $W_{Ref}(p)$ and $W_{NonRef}(p)$ are the weights assigned to pixel p in the weight maps of the reference and non-reference images.

In the next section, a brief discussion of the proposed de-ghosting approach for more than 2 images is provided.

2.1.3 De-ghosting for Multiple Non-Reference Images

The steps of the proposed de-ghosting approach for more than 2 input images are similar to the previously described steps for the case of a single non-reference image. The underlying idea is to detect the *inconsistencies* between the reference and non-reference images. Understandably, the detection of these inconsistencies, which are mostly induced by dynamic pixels, is preceded by the initial selection of the reference image from the input stack of LDR images. Several approaches for the selection of the reference have been proposed in previous works, which differ in the quality measure used for making the selection, such as in [6] and [5]. Alternatively, the choice of the reference can be relinquished to the user.

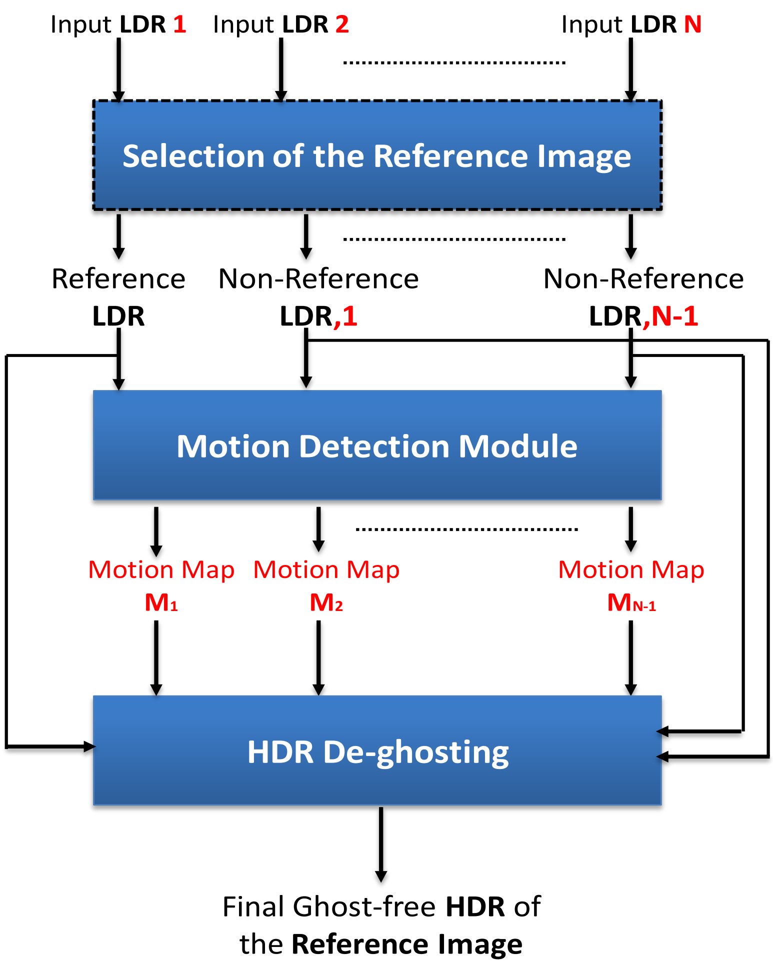


Figure 2.8: Illustration of the different stages of the proposed HDR de-ghosting for the case of N input images.

Figure 2.8 presents the different stages of the proposed de-ghosting approach for N input images ($N > 2$). The reference image selection step is succeeded by the motion detection module. In this stage, the algorithm computes for each non-reference image i the corresponding binary motion map $\hat{M}_i$ separately. This is done by comparing it to the reference image using the various steps described in the case of 2 input images. This results in $N-1$ binary motion maps, which describe the inconsistencies of each individual non-reference LDR image in comparison to the reference image.

Using the generated binary motion maps, the weighting map W_i of the non-reference image i is computed according to equation:

$$W_i(p) = C_i(p)^{\omega_C} \times S_i(p)^{\omega_S} \times E_i(p)^{\omega_E} \times \hat{M}_i(p), \tag{2.8}$$

where $\hat{M}_i$ is the binary motion map of the image i. Accordingly, the weighting map of the reference image W_{Ref} is expressed as

$$W_{Ref}(p) = \begin{cases} 1, & \text{if } M_{NonRef}(p) = 0 \\ C_{Ref}(p)^{\omega_C} \times S_{Ref}(p)^{\omega_S} \times E_{Ref}(p)^{\omega_E}, & \text{otherwise} \end{cases} \tag{2.9}$$

with M_{NonRef} representing the combined non-reference motion map computed according to

$$M_{NonRef}(p) = \sum_{i=1}^{N-1} \hat{M}_{NonRef,i}(p), \tag{2.10}$$

where $\hat{M}_{NonRef,i}(p)$ is the binary motion map of the image i at pixel p. The final HDR image corresponding to the reference LDR image is computed based on the EF algorithm and using the previously modified weighting maps.

2.2 Experimental Results

In this section, we provide the experimental evaluation of the proposed HDR de-ghosting method. To this end, we tested the performance of our approach on several sequences captured under various circumstances, such as varying exposure ratios between input LDR images, different numbers of input images and different types of object motion inside the captured scenes.

As discussed in earlier sections, the accuracy of the motion map is important for the ensuing de-ghosting step, since pixels marked as dynamic in the final binary motion map(s) are excluded from the rendered HDR image. This also includes probable false positives (wrongly detected pixels). As a result, unwanted artifacts might be visible in the final HDR image, especially in the case of 2 input LDR images, where the weighting values of the EF step for the dynamic regions is extracted only from the reference image.

The improvement brought by accurate motion maps is illustrated in Fig. 2.9, where 2 examples of HDR de-ghosting are shown. The first set (scene 1, first row) depicts a scene with large color differences and only 2 inputs. In addition, the depicted scene encloses abrupt motion, as well as objects present in only 1 frame. As shown in the figure, the final HDR is ghost- and artifact-free, and the computed motion map accurately detects the motion-related areas, with very few false positives.

The second scene (second row) is composed of 4 LDR images with significant scene motion. Likewise, the depicted scene comprises objects which appear in 1 or 2 images of the input stack, as shown in Fig. 2.10. Nevertheless, our algorithm accurately detects the dynamic regions and pixels in comparison to the designated reference image, thus resulting in a ghost-free HDR image with no artifacts or additional noise.

Figure 2.9: Motion detection and HDR results of our approach on sequences from [7] (scene 1) and [6] (scene 2). The tested sequences present large color differences between the input LDR images as well as complex motion due to several dynamic objects. The accurate binary maps improve the quality of the final HDR image. This is visible when we compare our results to the case where no de-ghosting is performed, as shown in the areas marked by the red boxes. Note that these results were generated using $k_1 = 0.09$, $w_1 = 3$ and $w_2 = 3$.

Furthermore, we compare the results of our approach with the results gained from the approach proposed by An *et al.* in [5]. Apart from being one of the most recent approaches for HDR de-ghosting, the method proposed in [5] is of a particular interest to us, as it is based on the computation of binary motion maps indicating the locations of the dynamic objects, as well as the modification of the EF algorithm.

To this end, we used the recently released LDR dataset by Karaduzovic-Hadziabdic *et al.* in [1], which comprises several multi-exposure scenes. In each scene, Karaduzovic-Hadziabdic *et al.* create a stack of 5 differently exposed LDR images, ranging from dark to bright. Besides, several types of motion between the LDR images are included. The exposure ratio between each consecutive pair of images is set to 2. In addition, each scene contains an additional stack of differently exposed LDR images aligned to the view of LDR image 3. This setup is shown in Fig. 2.10.

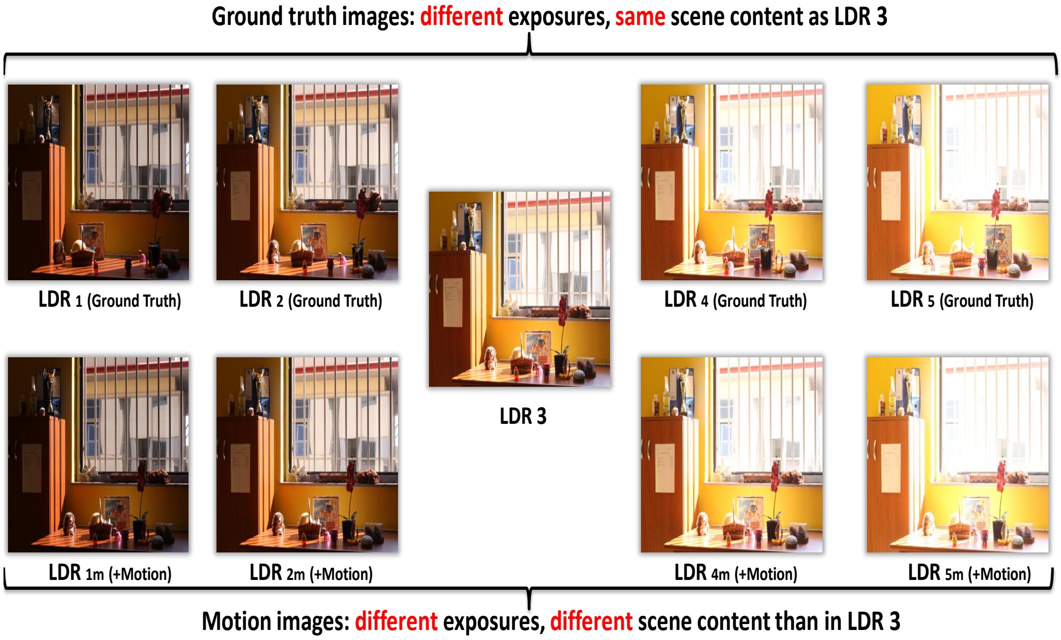


Figure 2.10: Illustration of a scene containing complex motion from the dataset presented in [1]. The scene contains 5 differently exposed LDR images depicting the same scene as LDR image 3 (first row). In addition, the scene includes 4 more additional LDR images, which are differently exposed and depict different scene content than LDR image 3, due to the introduced scene motion (second row).

Using this dataset, we create 10 subsets, each composed of 3 images: LDR_{1m}, LDR_3 and LDR_{5m}, as shown in Fig. 2.10. Furthermore, each subset contains 2 additional images, namely LDR_1 and LDR_5, which are fused with LDR_3 using EF for the purpose of creating the *Ground-Truth HDR* image of the scene. The ground-truth HDR image is used for the evaluation of the results of our approach as well as the results of the method proposed by An *et al.* in [5]. The results of this comparison are presented in Table 2.1, where the PSNR values are gained from the comparison of the final HDR image from [5] as well as ours, with the ground-truth HDR images rendered from LDR_1, LDR_3 and LDR_5 and using the EF algorithm. Additionally, the table contains the execution times of both approaches for the computation of the binary motion maps.

As shown in the table, the PSNR values generated using our approach are constantly higher than the PSNRs from the approach of An *et al.* [5], achieving an average lead of $11.15\ dB$ across the 10 subsets used for the comparison. Furthermore, our method not only improves in terms of PSNR, but also requires considerably less time to compute the binary motion map. As shown in Table 2.1, the average run time improvement over the 10 subsets is $241.61\ sec.$. Note that these tests were conducted on a computer with standard configuration and using *Matlab* implementations of both approaches.

The enhancement in terms of de-ghosting performance is also shown in Fig. 2.11, which

Subset	Approach proposed in [5]		Our Approach	
	PSNR (dB)	**Execution Time** (sec.)	**PSNR** (dB)	**Execution Time** (sec.)
(1)	20.023	251.66	**26.239**	**4.28**
(2)	22.463	255.84	**27.065**	**4.31**
(3)	17.992	267.36	**26.084**	**4.68**
(4)	13.343	251.46	**27.067**	**4.81**
(5)	14.346	263.32	**34.525**	**4.41**
(6)	13.828	229.05	**32.559**	**4.66**
(7)	22.848	247.21	**28.971**	**4.41**
(8)	19.028	224.83	**28.601**	**4.63**
(9)	16.743	225.96	**34.377**	**5.09**
(10)	23.522	245.29	**30.319**	**4.67**

Table 2.1: PSNR values and run-time in seconds of the method proposed by An *et al.* in [5], alongside the results generated using our proposed approach. Note that the execution times presented here describe the required time to generate the **binary motion maps**. For these tests, k_1 was set to 0.09.

contains the final HDR results generated using the approach described in [5] as well as our results. Similar to the conclusions drawn based on the objective evaluation presented in Table 2.1, the improvement in the accuracy of the gained binary motion maps impacts positively the de-ghosting performance. The final HDR images generated using our de-ghosting technique contain less artifacts.

2.3 Discussion

So far we described a novel de-ghosting approach based on the detection of dynamic objects. The motion maps are constructed based on the manipulation of the difference image between the input LDR images. The high accuracy of the gained motion maps allows for a ghost-free final HDR. Aside from its ease of implementation and low complexity, our approach obtains good quality even on extreme cases with few input LDR images.

The proposed approach experiences performance limitations in certain challenging cases, apart from the common scenarios such as large exposure ratio between the input LDR images or complex motion in the depicted scene. Tests have shown that saturated areas represent the most challenging conditions to the proposed de-ghosting approach. These limitations arise when the selected reference image contains large saturated areas, where the motion occurs. In this particular situation, even if the dynamic object in the saturated area is correctly detected, the final HDR image in this region is based solely on the saturated parts in the reference image. Consequently, artifacts can be noticed on the final

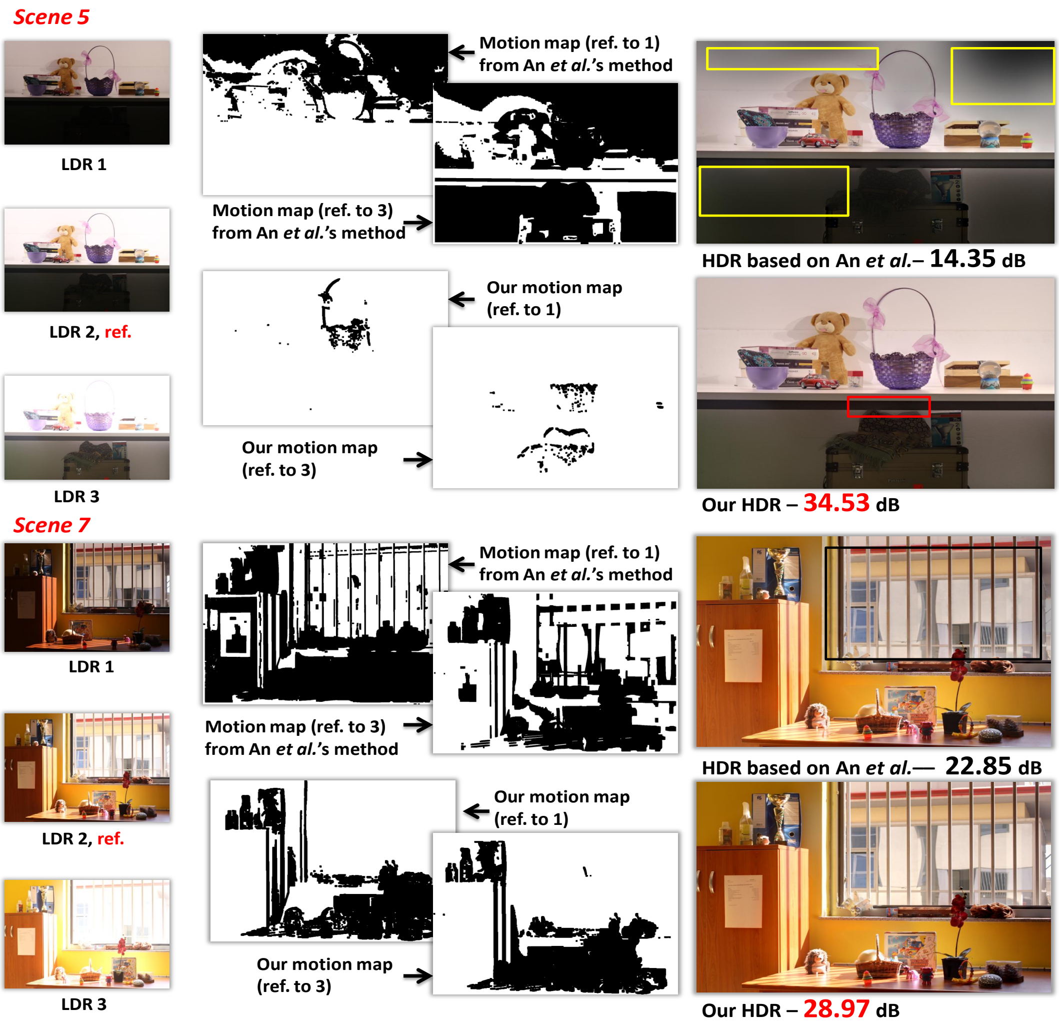


Figure 2.11: Comparison of the results generated using the approach described in [5] together with our results. The accurate and relatively noise-free binary motion maps computed using our method significantly improve the visual quality of the final HDR. This explains the large difference between both approaches in terms of PSNR values. Areas under the yellow boxes represent regions where the inaccurate motion detection from [5] negatively impacts the final HDR. Red boxes indicate regions where our algorithm creates artifacts. Input images are courtesy of [1].

HDR image, especially in the case of 2 input LDR images. An example of these limitations is provided in Fig. 2.12. Despite the accurate detection of the dynamic objects in both scenes, artifacts can be seen in the final HDR image (regions under the yellow boxes). These artifacts can be avoided either by correctly selecting the reference image, or by extending the stack of input LDR images to more than 2 images, especially with scenes where the corresponding dynamic range is very large.

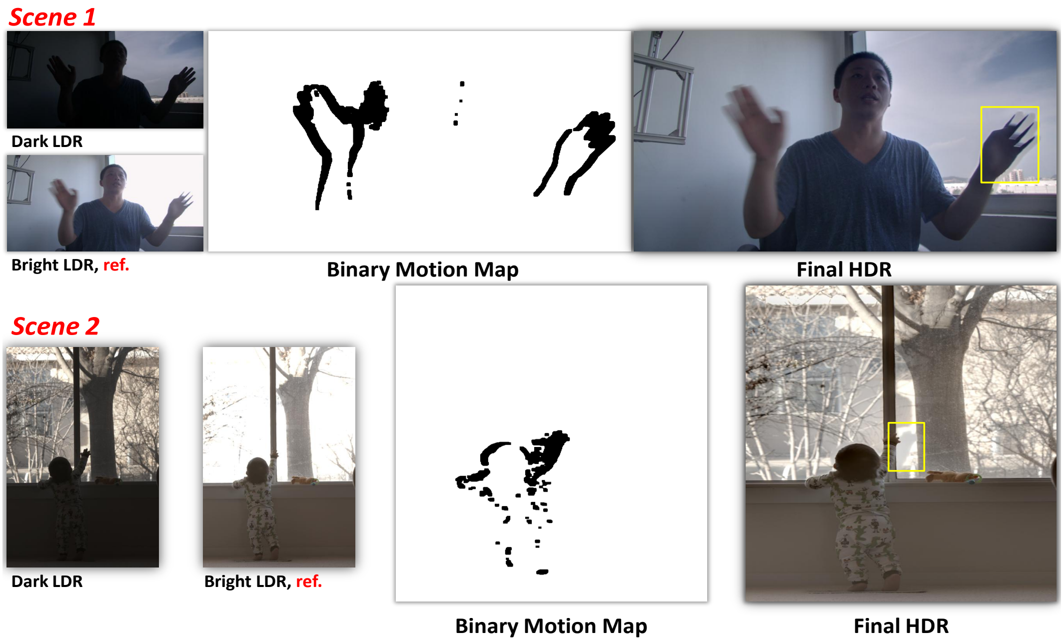


Figure 2.12: Illustration of two challenging cases where the motion occurs in the saturated areas of the selected reference image. Although the dynamic objects where successfully detected as shown in the binary motion maps, the final HDR still presents small color artifacts (areas under the yellow boxes), as the HDR image is constructed from only 1 image (reference image) in these areas. Images of scene 2 are courtesy of [8].

Furthermore, the color mapping step based on the HM algorithm is particularly prone to noise. As a consequence, the accuracy of the detection stage and the quality of the de-ghosting procedure are both negatively affected. In fact, HM yields limited mapping results when the input images are noisy. Additionally, in case the input images contain large homogeneous areas and less texture, equal intensities in the source image corresponding to the homogeneous regions will be mapped to different target intensities, which creates visible artifacts in the final result. Accordingly, if the color distribution of the input images is limited, larger parts in the source histogram will be mapped to smaller parts in the target histogram. An example of these artifacts is shown in Fig. 2.13.

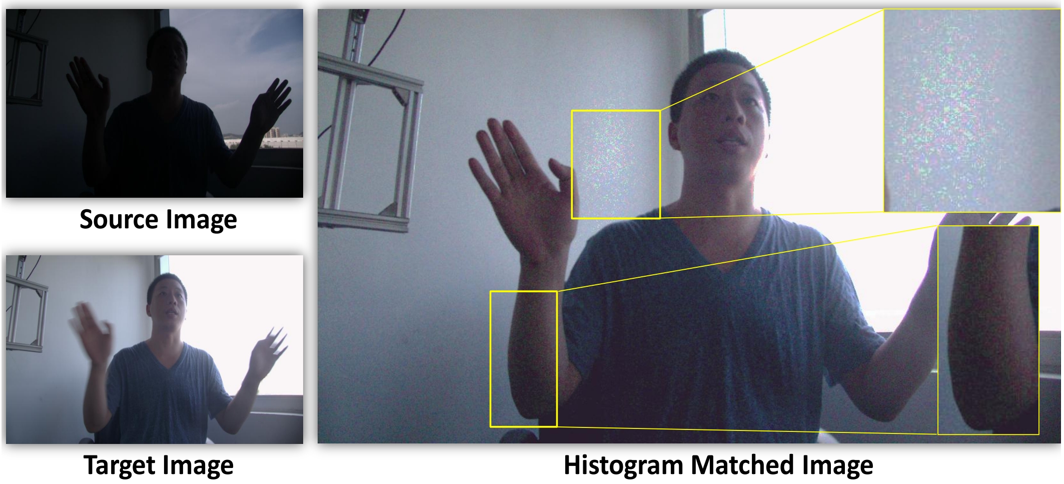


Figure 2.13: Illustration of a scenario where HM yields poor results. Clear artifacts are visible in the histogram matched image, especially in the homogeneous areas next to the head (areas encompassed in the yellow boxes).

In the next Chapter, we propose a low-complexity approach for the detection and correction of HM-related mismatches and artifacts. Furthermore, we assess the impact of the introduced modifications on the HM performance as well as on the de-ghosting approach discussed in this Chapter.

3 Histogram Matching Processing

As discussed in the previous chapter, HM [4] restrains the quality of the proposed deghosting approach, despite its relative low-complexity. Therefore, improving the performance of HM while preserving its low-computational cost represents the goal of this chapter. However, in order to understand the limitations of HM and develop a strategy to tackle them, a look at the general scope which encompasses the HM algorithm seems to be imperative at this stage.

For a wide range of applications related to image processing and computer vision, the input pair of images to process have different color properties. This difference is mostly related to the exposure settings of the capturing devices or the illumination conditions of the captured scene. In addition, the scene content described by these images might be different mainly due to camera or scene-related motion or different capturing times. Accordingly, for numerous computer vision and image processing applications, the illumination consistency assumption between the input images plays a central role. In this context, color mapping enables the reduction of these color differences.

In various computer vision applications, color mapping approaches based on color statistics are more favorable, especially in case low-complexity is a priority. HM is an efficient and well-known technique for color mapping, where the histogram of a source image is matched to the histogram of a target image. However, in cases where the images are not perfectly aligned due to camera or object motion, HM can result in visible color artifacts. Furthermore, HM typically involves brightening dark regions of the source image, which at the same time enhances the image noise. As a result, there is a need for detecting and correcting color artifacts and strong image noise.

To the best of our knowledge, there has been only few prior works dealing with HM on only two images with color differences. In [54], Rolland *et al.* propose fast implementations in order to increase the computational speed of HM. In [55], Shapira *et al.* propose to combine spatial dependencies between local regions in the source image. This approach aims at finding an optimal monotonic color mapping which fits the set of calculated histogram pairs. It performs well if the input images have small color differences and it is limited to the case where there is no camera motion while capturing the image pair.

To deal with HM-related limitations, we present a novel post-processing method for improving the quality of histogram matched images using statistical properties of the target image. The proposed method is based on ensembles of neighboring pixels. We learn the statistics of such ensembles within the target image, which exhibits content difference in comparison to the source image. Based on these statistics, we detect and reconstruct faulty intensities (outliers), by following a *naive Bayesian framework* [56] based on likelihood maximization. A shortened version of our approach for HM enhancement is presented in [57].

3.1 Proposed Approach

We propose a novel post-processing method for detecting and correcting HM-related color mismatches using a naive *Bayes approach*. In a nutshell, after applying HM, we perform edge detection on the histogram matched image and the target image, respectively. Then, we compute color statistics on the target image. Finally, on the histogram matched image, we detect outliers (noise, artifacts) and correct them. These steps are illustrated in Fig. 3.1 and will be examined in details in the following sections.

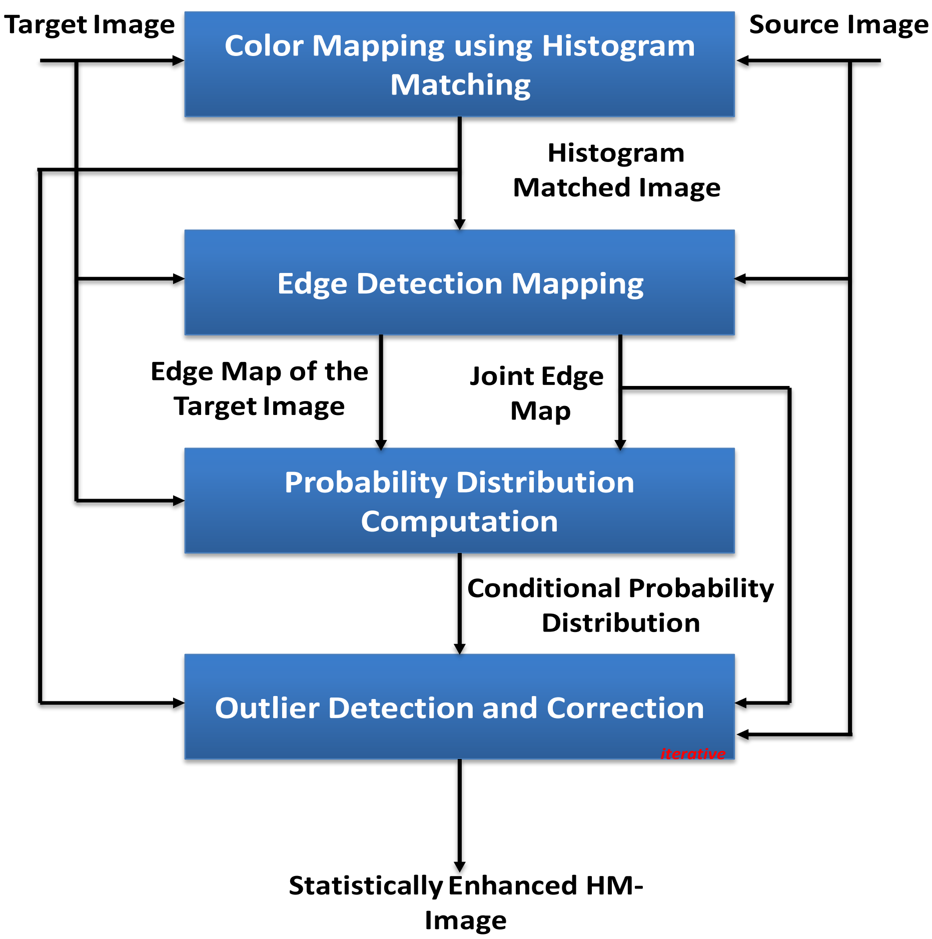


Figure 3.1: Illustration of the stages composing the proposed statistics-based approach for the detection and correction of HM-related artifacts.

We denote the location of the central pixel by (x,y) and the location of a neighbor pixel by $(x+i, y+j)$, where i, j denote the displacement coordinates from the central pixel coordinates. We work on 3 images:

- I^t: target image.
- I^s: source image.
- I^h: histogram matched image (prior to post-processing).

The target image I^t comprises the targeted color distribution. Therefore, for each color channel c, we estimate the conditional probability distribution

$$p(I_c^t(x,y)|I_c^t(x+i,y+j)) \tag{3.1}$$

of the central pixel intensity $I_c^t(x,y)$ given its neighbor pixel intensity $I_c^t(x+i,y+j)$ of the target image. We define the set of all neighbor pixel intensities $\theta_c^r(x,y)$ by

$$\theta_c^t(x,y) = \{(i,j) \in \mathscr{H} : I_c^t(x+i,y+j)\}, \tag{3.2}$$

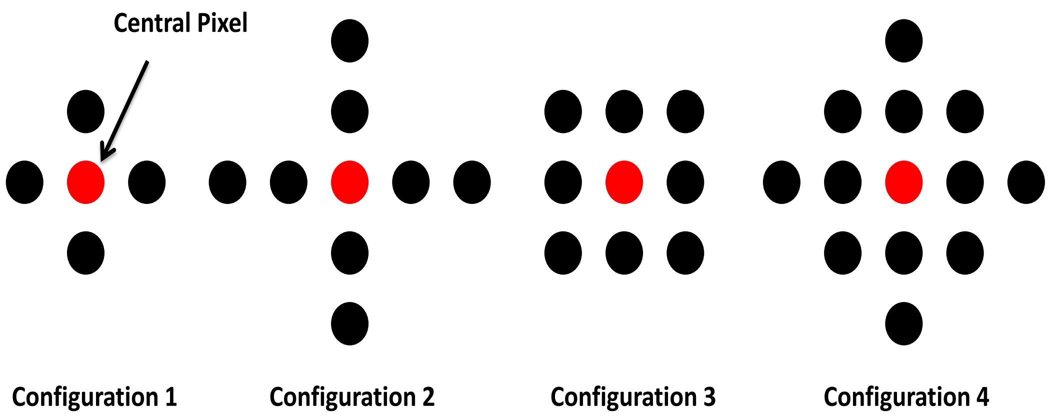


Figure 3.2: Examples of neighborhood configurations.

where (i,j) is a displacement vector and $\mathscr{H}$ is the set of all displacement vectors corresponding to a neighborhood configuration. For example, for the 4-neighbors configuration as shown in Fig.3.2 (first left side) we have

$$\mathscr{H} = \{(1,0),(0,1),(-1,0),(0,-1)\}. \tag{3.3}$$

Our basic assumption is that the conditional color distributions in the target image should be the same as in the histogram matched image, except for certain areas such as occluded regions and new areas introduced by camera motion. The detection and correction steps can be iterated several times in order to improve the performance of the algorithm. The proposed approach deals with different exposure ratios between the input pair of images, therefore applicable to different scenarios with various exposure ratio.

Due to the sparsity of the distributions $p(I(x,y)|I(x+i,y+j))$, the naive Bayes approach of choice, which aims at maximizing the product $\prod_{(i,j)\in\mathscr{H}} p(I(x,y)|I(x+i,y+j))$ requires proper smoothing of the distributions. The distribution-smoothing operation is performed prior to the outlier detection step. In the following, we will go through the building blocks of the proposed method, as illustrated in Fig. 3.1.

3.1.1 Edge Detection and Mapping

The proposed approach starts with performing HM on the source image I^s. This creates an image I^h whose color properties are ideally close to the target image I^t. Similar to the strategy deployed in the previously proposed de-ghosting approach, we set the under-exposed image (dark input image) as the source image, and consequently the over-exposed image as the corresponding target image. The assignment of source and target images as described is motivated by the fact that under-exposed images usually contain less saturated regions, in contrast to over-exposed bright images. This results in a better color mapping operation using HM.

Prior to the computation of the probability distribution of the target image, empirical tests have shown that edges in both images require special attention due to the fact that they comprise large gradients. This leads to sparse support for the conditional distributions $p(I^t_c(x,y)|I^t_c(x+i,y+j))$. As a consequence, edges are mainly detected as erroneous regions, thus causing a significant decrease of the overall image quality. To prevent this, we apply *Canny Edge Detector* [9] on both I^h and I^t. The purpose of this step is to exclude pixels labeled as *edge* from subsequent processing, such that neither the estimation of the conditional distribution nor the correction will be done on these pixels.

Furthermore, we detect the edges on the source image I^s. This allows us to find edges or object borders that were not detected on I^h, especially in saturated regions in I^h. In addition, we fuse both edge maps from I^s and I^h to create a final edge map with improved quality. However, in order to efficiently merge both edge maps, we compute the *well-exposedness* values of each edge pixel from I^s and I^h, as suggested in [2]. More specifically, the well-exposedness value E_p of the pixel intensity $I_p \in \{0,1\}$ is computed as follows:

$$E_p = \exp\left(-\frac{(I_p-0.5)^2}{2\sigma^2}\right). \tag{3.4}$$

The well-exposedness values computed for each pixel in I^s and I^h are used to select the edge pixel which is *best-exposed* (having the higher well-exposedness value) to be part of the final edge map.

Moreover, we restrain edge pixels from being part of the set of neighbor pixels, in order to prevent the corruption of the conditional probabilities. This allows us to exclude pixels in the immediate vicinity of edges from further computations. Accordingly, the size of the expansion depends on the depth of the chosen configuration. For example, in case configuration 4 is selected (see Fig. 3.2), the width of the expansion is set to 2 in each direction.

3.1.2 Detection of Color Mismatches

In this section, we describe the detection of the erroneous pixels in the histogram matched image I^h. As shown in Fig 3.1, a variety of possible configurations is available for selection, prior to the computation of the conditional probability distribution.

For a selected neighborhood configuration θ and for each neighbor defined by its coordinates (i,j), we estimate the conditional probability distribution from the target image $p(I_c^t(x,y)|I_c^t(x+i,y+j))$ by estimating the joint probability distribution $p(I_c^t(x,y),I_c^t(x+i,y+j))$ and the probability distribution $p(I_c^t(x+i,y+j))$. This is achieved by a simple calculation of the frequency $f(I_c^t(x,y),I_c^t(x+i,y+j))$ of the joint occurrences for $(I_c^t(x,y),I_c^t(x+i,y+j))$, and the frequency $f(I_c^t(x+i,y+j))$ for the occurrences of $I_c^t(x+i,y+j)$, respectively.

According to Bayes theorem [56], it is

$$p(a|b) = \frac{p(a,b)}{p(b)}. \tag{3.5}$$

A probability distribution is a normalized frequency distribution, i.e.,

$$p(a) = \frac{f(a)}{\#\text{total occurences}}. \tag{3.6}$$

There are as many non-gradient pixels N as specific neighbors, so we can write

$$p(I(x,y)|I(x+i,y+j)) = \frac{p(I(x,y),I(x+i,y+j))}{p(I(x+i,y+j))} \tag{3.7}$$

$$= \frac{f(I(x,y),I(x+i,y+j))}{N}\frac{N}{f(I(x+i,y+j))} \tag{3.8}$$

$$= \frac{f(I(x,y),I(x+i,y+j))}{f(I(x+i,y+j))}, \tag{3.9}$$

The inherent sparsity of the computed probability distributions (discrete distributions) constraints the subsequent MAP estimation during the correction step. This explains the need for a smooth and continuous representation of the distribution curve. This representation is obtained through a *kernel smoothing function* with a bandwidth value σ. After smoothing, we obtain a continuous probability distribution function, which we re-sample to obtain again a smoothed discrete probability distribution. For this purpose, we smooth the discrete probability distribution $p(I(x,y),I(x+i,y+j))$ using a Gaussian-shaped $2D$

smoothing function:

$$p_s(I_c, I_n) = (1/M) \sum_{\tilde{I}_c=0}^{255} \sum_{\tilde{I}_n=0}^{255} p(\tilde{I}_c, \tilde{I}_n) \exp\left(\frac{-((I_c - \tilde{I}_c)^2 + (I_n - \tilde{I}_n)^2)}{2\sigma^2}\right) \tag{3.10}$$

where I_c and I_n are variables representing the intensities of the central and neighboring pixels respectively, and $\tilde{I}_c$ and $\tilde{I}_n$ are the corresponding samples giving support in the discrete distribution. Accordingly, $1/M$ is a normalization factor such that $\int\int p_s(I_c, I_n) dI_c dI_n = 1$. The bandwidth value σ is set empirically to 2 using several tests conducted on the available datasets.

The smoothed probability distribution becomes:

$$p(I(x,y)|I(x+i,y+j)) = \frac{p_s(I(x,y), I(x+i,y+j))}{p_s(I(x+i,y+j))}. \tag{3.11}$$

Note that $p_s(I(x+i,y+j))$ is easily computed out of $p_s(I(x,y), I(x+i,y+j))$ using the formula

$$p_s(I(x+i,y+j)) = \int p_s(I(x,y), I(x+i,y+j)) dI(x,y). \tag{3.12}$$

Once the smoothing operation of the distributions is finished, a tuple $(I_c^h(x,y), I_c^h(x+i,y+j))$ in I^h can be verified to be valid or a mismatch using the proposed naive Bayes approach. This is done by comparing the product $\prod_{(i,j)\in\mathscr{H}} p(I_c^h(x,y)|I_c^h(x+i,y+j))$ to the term $\Gamma = (1/\gamma)^N$. The parameter γ varies empirically between 2 and 12 and regulates the detection tolerance of the proposed approach. A low γ value allows for more outliers to be detected. Accordingly, a higher γ means that less pixels will be labeled as HM-related mismatches, which is helpful against possible false positives. The comparison is performed in the logarithmic space. The transformation to the logarithmic space eases the subsequent computation of the MAP estimate during the correction step:

- $\sum_{(i,j)\in\mathscr{H}} \log(p(I_c^h(x,y)|I_c^h(x+i,y+j))) > \log(\Gamma) \Rightarrow$ likely to be a correct match, no HM outlier detected.
- $\sum_{(i,j)\in\mathscr{H}} \log(p(I_c^h(x,y)|I_c^h(x+i,y+j))) < \log(\Gamma) \Rightarrow$ likely to be a mismatch, the central pixel is marked as an outlier.

The detection operation is performed on all pixels in the histogram matched image and separately for each color channel. The operation results in 3 different binary outlier maps of the *Red*, *Green* and *Blue* color channels. Examples of these binary maps is shown in Fig. 3.3.

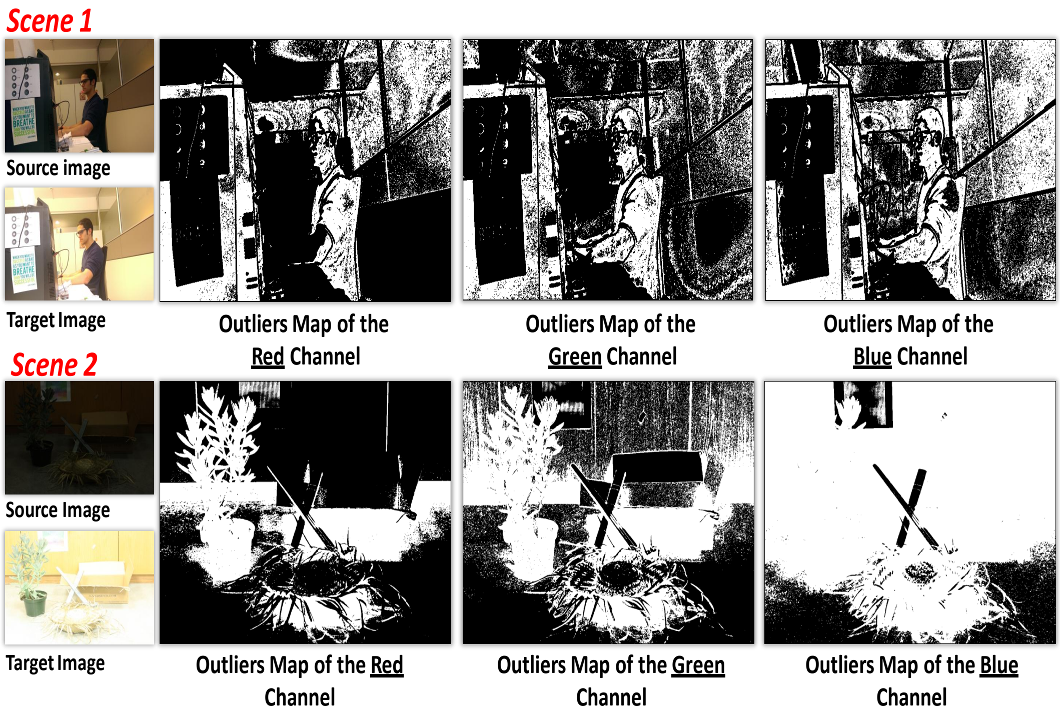


Figure 3.3: Examples of binary outlier maps indicating the locations of detected HM-related mismatches. Note that **Black** pixels indicate **Inliers**. **White** pixels indicate **detected mismatches**. For these scenes, the edge maps were created using *Canny Edge Detector* [9] and a threshold γ equal to 12. Images of scene 2 are courtesy of [10].

Using the computed binary outlier maps, we perform the correction of the detected HM-mismatches for each color channel separately. For this purpose, we compute the MAP estimate $\hat{I}_c^h(x,y)$ of the sum of the logarithmic probabilities, i.e.,

$$\hat{I}_c^h(x,y) = \underset{I\in\{0,255\}}{\arg\max} \sum_{(i,j)\in\mathscr{H}} \log(p(I|I_c^h(x+i,y+j))). \tag{3.13}$$

The underlying idea is to estimate the central pixel intensity I_c^h which maximizes the sum of the logarithmic distributions.

The detection and correction procedures can be optionally iterated in order to improve the performance of the algorithm. Empirically, performance saturation is reached after 2-3 iterations. This depends of course on the nature of the input images, and the performance of the original HM. However, tests have shown that more iterations tend to blur the final results, which decreases the perceptual quality of the results especially for images with much texture.

In addition, in case of multiple iterations, we update the edge map of I^h after each iteration. In fact, the updated image I^h comprises less noise and artifacts, therefore allowing for a better edge detection accuracy.

Finally, we apply a *Gaussian* low-pass filter on the corrected pixels. This allows for a smoother merging of the newly corrected pixels with the surrounding regions.

3.2 Experimental Results

We test the performance of the proposed approach on several image pairs, in order to assess the enhancement introduced by the proposed method. Understandably, each image pair has different properties, namely different exposure ratio between the input dark and bright images. In addition, we test the proposed method on different types of motion (slow/fast motion, complex motion) and on stereo images.

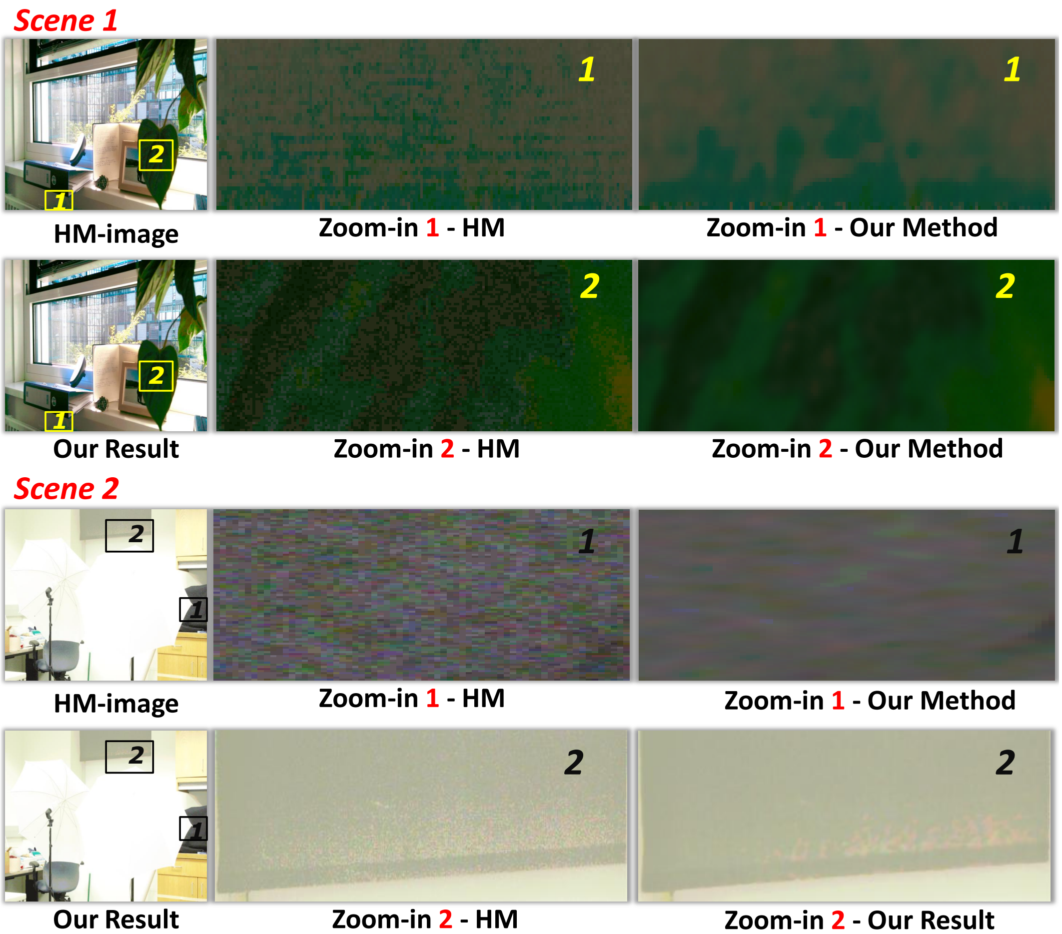


Figure 3.4: Illustration of the enhancement of the HM results using our proposed approach. Our results present less artifacts, while maintaining sharp edges. For both scenes we set the threshold γ equal to 12. Images of scene 2 are courtesy of [10].

Examples of the visual enhancement introduced by our approach are shown in Fig. 3.4. It can be seen that most artifacts caused by HM are fixed, such that the overall visual quality is improved. Note that our method effectively smooths uniform areas with less texture while maintaining edges. This way, we obtain selective denoising.

In addition to the visual assessment of the proposed HM enhancement method, we conduct experimental evaluations by comparing the enhanced HM-image to its corresponding ground-truth image. Typically, this comparison requires a second source image which depicts scene content identical to the source image I^s, but has the same color properties as the target image I^h. To this end, we test the performance of the approach on images from the *Middlebury Stereo* dataset [11], which offers the required setup (second source image for the evaluation). Additionally, we created new scenes using a smartphone camera as capturing device. To this end, we developed a camera application for this particular purpose, which allows us to capture 10 consecutive frames of the same scene automatically, alternating each time between under-exposure (dark) and over-exposure (bright). This operation results in 5 under-exposed and 5 over-exposed images. The entire process is therefore hand-free and does require any user input to change the exposure settings after each captured frame.

In order to gain the needed ground-truth image, we fix the camera and capture a static scene during the first 5 frames. This way, we obtain our source image I^s as well as the ground-truth image I^{gt}. Next, we introduce motion during the final 5 frames. The introduced motion can be the result of either camera-motion or due to a dynamic object in the scene. As consequence, we obtain the target images I^t, which has the same color properties as I^{gt} but depicts a different scene content. Our goal is to perform the proposed enhancement algorithm on the HM-image (resulting from applying HM on I^t and I^s) and to compare it the ground-truth image I^{gt}. The described capturing procedure is illustrated in Fig. 3.5.

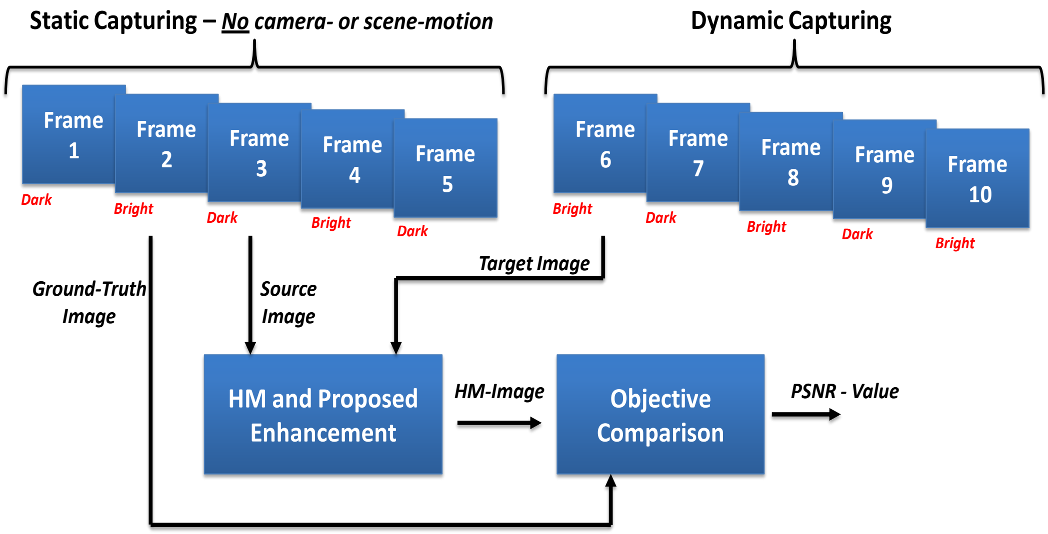


Figure 3.5: Graphical representation of the capturing of the target, source and ground-truth images using the proposed approach. Notice that the source image is chosen from the frames 1, 3 or 5. The ground-truth images is selected from frames 2 or 4. Accordingly, the target image is either frame 6, 8 or 10.

The images used for the subsequent experimental evaluation of the performance of our approach are shown in Fig. 3.6. The presented image pairs comprise significant color and content differences. Furthermore, the testing sequences exhibit different types of motion (camera and/or object motion) and were taken under different illumination conditions. This creates the requested diversity needed to test the performance of the proposed approach.

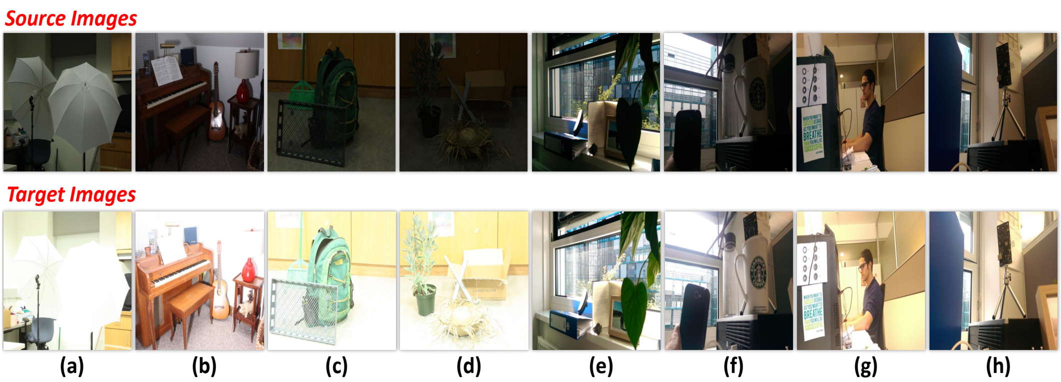


Figure 3.6: Scenes used for the quantitative evaluation of the proposed approach. The input images comprise significant color and content differences due different exposure settings and camera or object motion. Scenes (a), (b), (c) and (d) are courtesy of [10]. Scenes (e), (f), (g) and (h) were gained using the approach described previously and illustrated in Fig. 3.5.

Scene	**Original HM** (dB)	**Our Method** **1** Iteration (dB)	**Our Method** **3** Iterations (dB)	**Our Method** **1** Iteration **No Edge-removal** (dB)	**NLM** [16] (dB)
(a)	30.603	31.629	**31.636**	30.644	30.904
(b)	27.699	**28.021**	27.864	27.479	27.848
(c)	26.550	27.410	**27.614**	26.779	26.373
(d)	28.249	**30.015**	29.693	29.101	29.931
(e)	26.212	**26.531**	26.452	24.756	26.509
(f)	32.744	**33.279**	33.045	29.854	32.998
(g)	35.125	**35.550**	35.286	28.929	35.377
(h)	29.829	**30.792**	30.706	29.354	30.540

Table 3.1: PSNR values resulting from the comparison of the original HM, our method (1 and 3 iterations, with- and without edge-removal) as well as the denoising algorithm NLM (Non-local Means) [16] using optimal parameter settings, against available ground-truth images (scenes are shown in Fig. 3.6). For these tests we used a Desktop PC with a Core I5 CPU. Results were gained based on the 12-neighbors configuration, Canny edge detector and a threshold $\gamma = 12$.

The PSNR results gained from the comparison are presented in Table 3.1, which contains the results generated using our approach (1 iteration and 3 iterations), as well as the results gained without the proposed edge removal procedure. In addition, Table 3.1 contains the PSNR results using the *Non-Local-Means* [16] denoising algorithm applied on all tested scenes. This enables us to compare the performance of our approach against low-complexity yet well-performing denoising algorithm.

Considering the results presented in Table 3.1, 3 main conclusions can be drawn:

- There is no clear winner between single and multiple iterations. Empirically, the choice of the number of iterations depends on the nature of the scene. For scenes rich with texture, 1 iteration delivers better results, as multiple iterations tend to over-blur the HM-image. Understandably, scenes with less texture and large plain areas can benefit from multiple iterations of the enhancement algorithm. This observation explains the results for scenes **(a)** and **(c)**, where 3 iterations deliver better results. These 2 scenes contain relatively less texture in comparison to the remaining scenes. Accordingly, 1 iteration yields the highest PSNR values for the remaining texture-rich scenes.
- The evaluation shows the positive impact of the edge removal stage on the proposed approach. The PSNRs for the proposed method **without** edge removal are in some cases lower than the PSNRs of the original HM-images.
- Our approach outperforms the results gained using the denoising algorithm NLM. The average PSNR improvement is around $0.37\ dB$.

In addition to these experimental evaluations, we assess the performance of our enhancement approach on the scenes introduced earlier in Section 2.2 (see Fig. 2.10). As described earlier, each subset used for evaluation is composed of 3 LDR images, ranging respectively from under-exposure to over-exposure, with the middle image LDR_3 appointed as the corresponding reference image for the de-ghosting algorithm. However, since we only have *ground-truth* images of LDR_3 (LDR images containing the same scene but captured under different exposure settings), we split each subset into 2 scenes composed respectively of LDR_3 and LDR_{1m}, as well as LDR_3 and LDR_{5m}. Understandably, we perform the proposed enhancement algorithm on each scene separately. In both operations, we set LDR_3 to the source image I^s. As a consequence, the HM-enhancement approach is applied for the case of color mapping from *dark* to bright (LDR_3 as source image I^s and LDR_{5m} as target image I^t), as well as for the case of color mapping from *bright* to *dark* (LDR_3 as source image I^s and LDR_{1m} as target image I^t). An example of this setup is shown in Fig. 3.8.

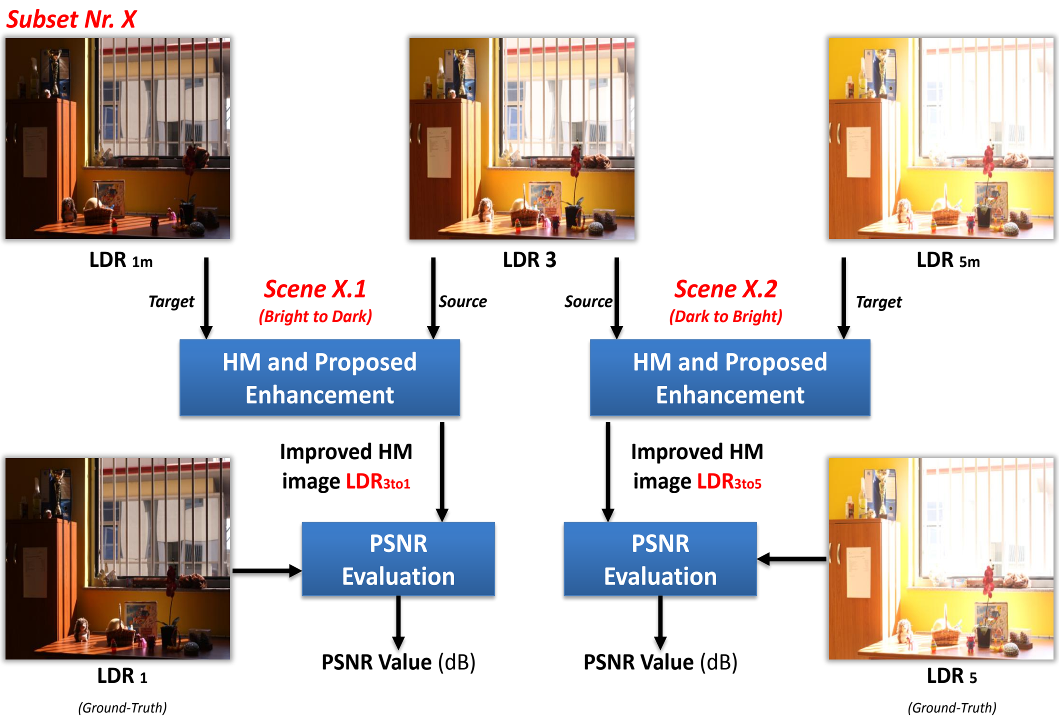


Figure 3.7: Example of the 2 scenes created from a subset as described previously. The proposed HM enhancement algorithm is applied on both scenes. PSNR results are gained using the available ground-truth images. Images courtesy of [1].

The improved HM-images will be subsequently compared to the available ground-truth LDRs, namely images LDR_1 and LDR_5. The PSNR comparisons resulting from the previously described evaluation setup are presented in table 3.1.

Scene	**Original HM** (dB)	**Our Method** **1** Iteration (dB)	**Our Method** **3** Iterations (dB)
(**1**.1)	32.831	32.884	**32.901**
(**1**.2)	30.152	**30.199**	30.022
(**2**.1)	37.482	37.587	**37.611**
(**2**.2)	34.787	**34.943**	34.611
(**3**.1)	23.886	**31.712**	31.707
(**3**.2)	35.138	**35.537**	35.342
(**4**.1)	22.55	**31.608**	31.598
(**4**.2)	36.617	**37.168**	36.861
(**5**.1)	22.56	22.915	**22.962**
(**5**.2)	38.423	**39.2**	38.566
(**6**.1)	23.192	**32.251**	32.242
(**6**.2)	35.861	**36.405**	36.272
(**7**.1)	22.709	**22.859**	22.829
(**7**.2)	**31.933**	31.929	31.812
(**8**.1)	25.965	26.065	**26.066**
(**8**.2)	32.947	**33.002**	32.692
(**9**.1)	23.225	23.299	**23.307**
(**9**.2)	32.512	**32.518**	32.334
(**10**.1)	23.651	**23.832**	23.765
(**10**.2)	31.904	**32.031**	31.825

Table 3.2: PSNR values resulting from the comparison of the original HM as well as our enhancement approach (1 and 3 iterations) on the scenes gained using the previously described strategy (see Fig. 3.8). Results were gained based on the 12-neighbor configuration, Canny edge detector and a threshold $\gamma = 12$. Note that the denotation in column 1 (scene) is composed of 2 digits, where the first digit indicates the original subset, and the second digit points to the scene number related to the chosen setup (dark to bright or bright to dark).

As shown in Table 3.2, the proposed HM enhancement approach almost constantly improves the quality of the HM image, except for scene 7.2. The increase in PSNR values is more noticeable on scenes 3.1, 4.1 and 6.1, which correspond to the cases where the HM is performed from bright (over-exposed source image) to dark (under-exposed target image). The result of such HM operation is typically limited in terms of matching quality, as the source image (bright) generally contains more saturated areas (over-exposed). As a consequence, reconstructing texture in these areas through HM is an ill-posed problem. This explains the relatively low PSNR values resulting from HM images, where the source image is brighter than the target image (corresponds to scenes ending by the digit 1 in Table 3.2).

The proposed approach successfully detects and corrects the artifacts related to the mentioned HM-related limitations in scenes 3.1, 4.1 and 6.1. This explains the sharp increase in terms of PSNRs. In addition, Fig. 3.8 contains a comparison of the results from scenes 4.1 and 6.1. The proposed approach improves the quality of the HM image by accurately detecting strong HM-related mismatches, and accurately predicting the correct intensity values in these locations. On the other hand, the proposed approach achieves for certain scenes a relatively small increase, such as for 3.1, 4.1 and 6.1.

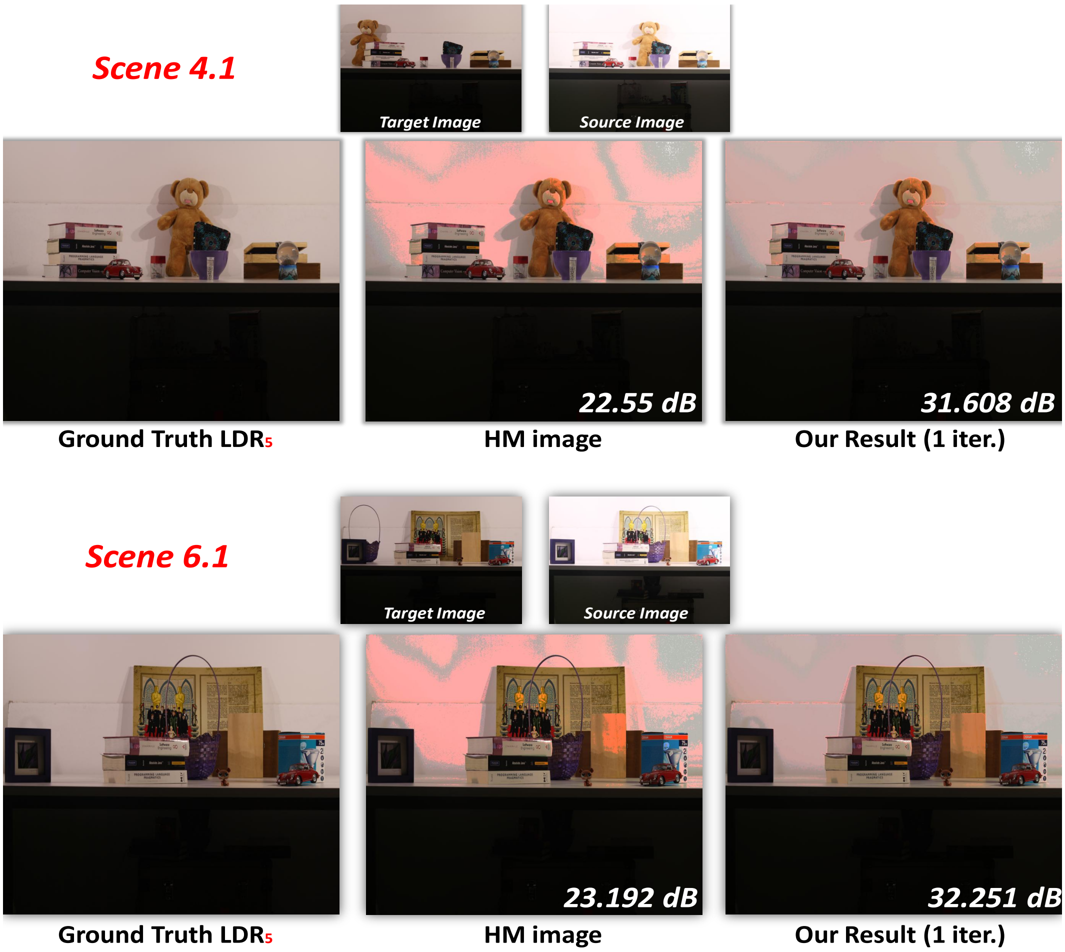


Figure 3.8: Examples of the enhancement of the HM images using our proposed approach. The resulting images are significantly improved in comparison to the HM images. Images courtesy of [1].

Furthermore, we are interested as well in assessing the impact of the HM enhancement algorithm on the previously proposed de-ghosting method. To this end, we apply the de-ghosting approach on the same sequences used in Section 2.2 and explained in Fig. 2.10. The idea behind these tests is to include the HM enhancement algorithm after each HM operation. Understandably, in case of an input stack composed of 3 LDRs (for HDR rendering), there will be a total number of 2 HM operations followed respectively by the enhancement step. We run tests over the stack of subsets and compare the gained final HDR image to the available ground-truth HDR image.

Subset	**Original De-ghosting** (dB)	**Original De-ghosting Including HM Enhancement** **1** Iteration (dB)
1	26.239	**26.242**
2	27.065	**27.079**
3	26.084	**26.094**
4	**27.067**	27.034
5	**34.525**	34.324
6	32.559	**32.634**
7	28.971	**29.026**
8	28.601	**28.622**
9	34.377	**34.448**
10	30.319	**30.344**

Table 3.3: Comparison of the PSNR results gained from the de-ghosting approach introduced in Chapter 2, alongside results generated using the same approach but with the HM enhancement algorithm included. For these tests, k_1 was set to 0.09.

From the PSNR values shown in Table 3.3, it is clear that the HM enhancement has a very limited influence on the quality of the final HDR image. This in turn means that the relative improvement of the HM images has a constrained impact on the accuracy of the motion detection step in the de-ghosting algorithm introduced in Chapter 2. In addition, the HM enhancement step significantly increases the complexity of the de-ghosting algorithm as a whole, as it contains time consuming steps such as the *Gaussian Smoothing* operation. Therefore, the reached quality improvement does not justify the increase in terms of computational cost.

These observations incite us to further investigate the topic of color mapping between images presenting content and color differences. However, since gradually improving the performance of the HM algorithm did not have the intended impact on de-ghosting, the logical course of action in this case would be to look for an alternative approach, in order to completely replace HM and provide and color mapping algorithm with a better mapping quality.

However, it is important to investigate and understand the constraints related to color mapping, in order to be able to improve upon exiting approaches such as HM. To this end, the next chapter will be dedicated to the analysis of exiting methods for color mapping, which fall under the scope of low-complexity approaches. This allows us to have a better understanding for their limitations in terms of mapping quality and subsequently guides us towards developing a novel and better performing mapping approach, such that we can improve beyond current state-of-the-art methods.

4 Non-Local Color Mapping

In this chapter, we seek to investigate the topic of color mapping which not only constitutes an essential element of HDRI, but also represents a fundamental building block for a wide range of computer vision applications such as *Stereo Matching*, *Optical Flow*, Camera Calibration and various other tasks.

Color consistency between the set of input images is a crucial precondition for a variety of applications in computer graphics and image processing, such as object recognition and detection, image and panorama stitching, motion estimation and compensation, disparity map computation, inter-frame color consistency and numerous additional tasks. For these applications, the color dissimilarity is in general caused by different illumination conditions during the capturing process, different camera exposure settings or simply different capturing times. This unintentional color difference is typical for *multi-camera* systems such as stereo and multi-view setups.

However, the nature of the application in certain scenarios imposes inherent radiometric variation between the input images. This is especially the case for HDRI, where the input LDR images are differently exposed, ranging from under-exposure (dark images) to over-exposure (bright with saturated areas). The input LDR images are subsequently merged into one single HDR image with a greater dynamic range. This technique, known as *Exposure Bracketing* requires the input LDR images to be aligned, in order to recover the *Camera Response Function* (CRF) or perform *Exposure Fusion* (EF) [2] directly. However, motion introduced by the capturing device or the scene itself violates this assumption. This calls for motion compensation, which in turn depends on the initial color mapping between the input LDR images. In this chapter we shift our focus to the latter scenario.

With this in mind, the main goal of color mapping is to transform the color properties of a *source image* so that they fit those of a *target image*. In other words, the resulting color mapped image typically depicts the same content information as the source image, but has the same color properties and hence the "exposure" of the target image. Accordingly, the task of color mapping is challenging, especially when the source and target images present large difference in terms of content and exposure.

In the next section, we provide a review of existing state-of-the-art approaches for color mapping. The investigation of these methods and their mapping performances allow us to identify their weaknesses, so that we know how to improve the overall quality of the mapping approach. Note that our proposed color mapping approach is outlined as well in our work presented in [58].

4.1 Related Work

4.1.1 Color Mapping

Color mapping is a well-covered research topic due to its relevance to various other tasks in image processing and computer vision. The bulk of existing approaches can be roughly split in 2 main categories, depending on the proposed trade-off between mapping quality and the ensuing complexity of the supporting solution.

The first category of approaches takes advantage of the resemblance between the input images for the purpose of modeling the color mapping function. Therefore, pixel correspondences are needed. Generally, methods which fall under the scope of geometry-based approaches differ in the way correspondences are computed, as well as the strategy used to model the color mapping and compensate the color differences. These correspondences can be either sparse [59, 60, 61, 62, 63] (using common algorithms for features detection and extraction such as *SIFT* [64] or *SURF* [65]), region-based [66, 67, 68] or dense [69, 70, 71]. The matched points or regions are used to model a mapping function, which typically assigns each intensity level (in case of 8-bit images, each level from 0 to 255) to a target intensity, using for example a look-up table similar approach.

This set of methods suffers from limitations related to the computational cost. This is a direct result of the costly detection and matching steps of feature vectors from the input images. Furthermore, performance-related issues are quite common, especially if source and target images present large scene and/or color differences. Consequently, less reliable feature points will be detected and matched, which decreases the robustness of the mapping operation.

The second category of the approaches omits pixel correspondences and relies on information provided by the statistical properties of the input images. The goal is to reshape the distribution of the source image so that it fits the distribution of the target image. In the experimental section of this Chapter, we compare the results of our approach with methods from this category, as we completely discard the target image in the application phase of our color mapping method, as explained later.

The first set of methods in this category takes advantage of color space transformations for performing color mapping, such as the uncorrelated *L'a'b* color space. This transformation decreases the complexity from a 3D color mapping to 3 separate 1D mappings. This was initially introduced by Reinhard *et al.* in [72]. In this work, the authors propose to transform the initial images to the *L'a'b* color space using a rotation matrix. This is followed by a translation and a scaling using the means and standard deviations of both source and target images. Similarly, Xiao and Ma propose in [73] to transform the input images to the axes of their *principal components*.

Moreover, alternative forms of distributions such as *histograms* represent a valuable statistical tool in the context of color mapping. The most notable algorithm in this category is the HM algorithm. As thoroughly explained in the previous chapter, HM exhibits performance decrease when dealing with images presenting large color and scene differences. Alternatively, Xiao and Ma propose in [74] to include HM in a global optimization framework, which aims at minimizing not only the difference between the source and target histograms, but also the difference between the gradient distributions of the source and the resulting color mapped image. Likewise, Pitie *et al.* suggest in [75] to model the 3D color distribution (such as *RGB*, *L'a'b*, etc.) altogether during the matching process, followed by a post-processing step where the gradient field of the output image is adapted to the gradient field of the source image. The approaches proposed in [74] and [75] tend however to over-blur the final images. Alternatively, Pouli *et al.* [76, 77] propose an approach which offers more control over the amount of mapping between source and target distributions. This is achieved by computing histograms in different scales and extracting persistent features in each scale, which correspond to low and high frequencies. It is important to notice most of these methods were designed to handle cases where the source and target images share few semantic information.

4.1.2 Limitations of Previous Works

The majority of color mapping approaches based on image statistics create a mapping which ideally assigns, for each intensity value I_s ($I_s \in [0,255]$), a single target intensity I_t, or 3 target intensities for each color channel. The final mapping is therefore a sort of a look-up table, which maps each input intensity to the corresponding target intensity designated for it. However, the generalization of the color mapping model as a *one-to-one* mapping is not valid in the real-world case. In other words, when we observe the true color mapping using ground-truth images (multiple instances of a LDR image captured with different exposure settings), each intensity I_s might have several target intensities I_t. Likewise, different intensities in the source image might be mapped to the same intensity in the target image. This observation is shown in Fig. 4.1.

Therefore, restraining the model space of color mapping to a one-to-one mapping explains the reason why most of the previously mentioned algorithms are limited in terms of mapping quality, especially when confronted with situations where the exposure ratio between input and source images is large, such as in HDRI. The main cause of this deviation from the general mapping model is image noise. There are many types and models of noise which are introduced at various stages during the image formation process chain. The most commonly investigated type is *Gaussian Noise* (a.c.a amplifier noise). This also explains the Gaussian shape of the intensity distributions presented in Fig. 4.1. In the context of HDRI, noise might corrupt the mapping, in the sense that pixels with the same intensities in the under-exposed image are matched to different target intensities in the over-exposed image, and vice versa.

In addition to image noise, the reconstruction of the color information, known as *Demosaicing*, further discredits the one-to-one mapping model. Typically, each cell of the *Color Filter Array* (CFA) is able to capture a single color channel (red, green or blue) according to a pre-defined pattern. The subsequent recovery of the missing color channels takes

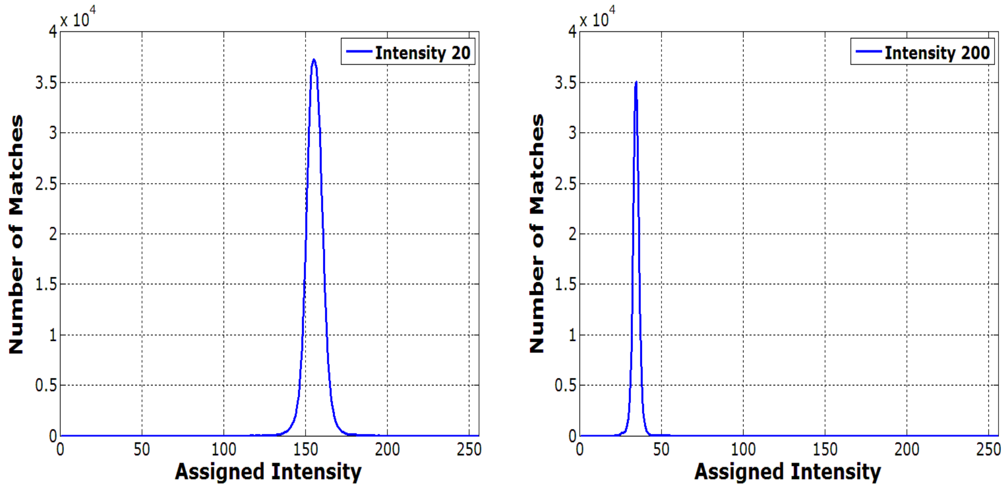


Figure 4.1: Illustration of different intensity mappings using ground-truth images (I_s and I_t depict the *same* scene but were captured with different exposure times). Left curve shows the mapping of intensity **20** from *dark* (I_s) to *bright* (I_t). Right curve represents the mapping of intensity **200** in the case of *bright* to *dark* color mapping.

advantage of the spatial correlation between pixels in the same region. This means that the observed pixel color intensities are strongly correlated to its immediate environment.

Finally, we must take into consideration saturation due to over- and under-exposure in the context of HDRI. In this context, the one-to-one mapping is no longer valid, since pixels in saturated areas with values close to 0 or 255 are typically mapped to different intensities, which represent the details of the scene lost in these saturated areas.

These observations point to the fact that high-quality color mapping is achieved by extending the one-to-one mapping approach to a more comprehensive model, which acknowledges the previously mentioned image formation circumstances while estimating the real mapping model as shown in Fig. 4.1. This suggests that a robust color mapping approach must consider these factors holistically by incorporating additional prior information for an accurate mapping model. These priors are inherently found in the immediate vicinity of each pixel, and as we show later, they enable the accurate estimation of the target intensity. In order to achieve this, we rely on *Convolutional Neural Networks* (CNN), which through the adoption of convolutions using sliding window kernels for the representation of higher dimensional features, include the color distribution of the environment of the pixel under investigation.

In the next section, we provide a brief overview of the theory behind convolutional networks, as well as CNN-based approaches, in particular methods falling under the scope of image processing.

4.1.3 Convolutional Neural Networks

Inspired by the advances made in the field of *neuroscience* for the purpose of unraveling the lack of clarity around complex biological systems such as the brain and its functions, neural networks in the context of machine learning aim at taking advantage of these discoveries in order to imitate complex tasks such as object detection and classification, speech and voice recognition systems and many more. Such artificial intelligence systems, known as *Artificial Neural Networks* (ANN) are typically used in the context of classification applications by echoing the basic component of the human brain, namely neurons. This is done by interconnecting a set of neurons which are organized in *layers*. A conventional ANN is composed of input *data layers*, *hidden layers* and a corresponding *output layer*. The idea behind ANNs is to learn a set of weights and biases for each neuron in order to determine whether it will be activated or not, and the respective output in case of activation. In the context of *supervised learning*, the simulated function represented by the set of learned parameters seeks to approximate the relationship between the presented input and output data (mostly known as labels) in the training phase. Accordingly, *unsupervised learning* is more challenging in the sense that no labels are available to guide the learning of the parameters.

However, for computer vision and image processing tasks, where the input data are typically images, learning the weights and biases in a multi-layer ANN is a computationally tedious task, as the number of parameters is very high, which puts a huge burden on the computational efficiency of the desired system. This is related to the fact that in ANNs, all neurons in hidden layers are interconnected.

To deal with these limitations, *Convolutional Neural Networks* allow to learn *shared* weights and biases due to the introduction of spatial convolutions. As thoroughly explained by O'Shea and Nash in [78], each convolutional layer in a CNN is composed of a set of n *filters*, with a kernel size $k_h \times k_w \times k_d$, where k_h, k_w and k_d are respectively the kernel height, width and depth. The convolution consists in the scalar product between each filter kernel and a local region under consideration in the input data. The input data can be either the image data or feature maps, depending on the location of the convolutional layer in the network. Sliding the filter kernel over the input data results in the so called *feature maps*, which typically depict high-dimensional abstractions of the data, thus the name *features*. The sharing of parameters in CNNs and in contrast to ANNs, is a result of the kernel sliding procedure, where the same filter is used for each region of the image. Accordingly, each convolutional layer composed of n such filters generates n features maps. The weights and biases corresponding to these filters are learned during the training phase. Note that the height and width k_h and k_w of the kernel, as well as n, *zero-padding* and the convolution-stride are considered as **hyperparameters**.

There exist several types of layers which can be used in CNNs. However, the most significant ones can be narrowed down to 5 layers:

- **Input Layers:** Layers containing the input data, specifically images. Input layers have neither parameters nor hyperparameters.

- **Convolutional Layers:** As mentioned earlier, convolutional layers represent the core component of CNNs. Convolutional layers extract higher dimensional representations from the data, which are refined through the network so that specific class scores can be assigned according to these extracted features.
- **Rectified Linear Unit ReLU:** Acts as an activation function to the output of the previous convolutional layer. ReLUs include the needed non-linearity to the estimated function describing the relationship between input and output data. Similar to the input layers, ReLUs have neither parameters nor hyperparameters.
- **Fully Connected Layers FC:** Necessary Component for classification tasks. FC layers compute the class scores based on the activations from previous layers [78].
- **Pooling Layers:** Usually used to perform spatial down-sampling for the purpose of parameters reduction, as explained in [78].

Recently, CNN-based approaches were proposed for low-level image processing tasks such as *Image Denoising*. In [79], Jain and Seung describe a method which recovers an underlying natural image from noisy input data, using a CNN-trained model learned from distinct noise models (unsupervised learning). In [80], Karbasi *et al.* suggest to combine image denoising with the task of learning redundant patterns. Moreover, the work presented in [81] by Junyuan *et al.* describes a convolutional network capable of performing image denoising and blind image in-painting simultaneously. Recently, Dong *et al.* introduced in [12] a new framework for image *Super-Resolution* (SR) based on a learned mapping between low- and high-resolution images. The architecture of the proposed convolutional network is simple, yet delivers state-of-art results. Our network developed for the task of color mapping is based on this work.

4.2 Color Mapping Using CNNs

4.2.1 Dataset

In this work, we use 2 different datasets for the purpose of training and testing the performance of our approach. Typically, each sequence from these datasets comprises 3 different images: Source and target images I_s and I_t and ground-truth image I_{gt}. We set the under-exposed image in each sequence to I_s. Respectively, I_t is the over-exposed image of the sequence. This means that we will perform color mapping from **dark to bright**.

The images I_s and I_t can be distinguished as having different scene and color information, with a corresponding exposure ratio $R = \frac{e_t}{e_s}$, where e_s and e_t are the corresponding exposure times of I_s and I_t. The image I_{gt} depicts the same scene information as I_s. The ground-truth image I_{gt} was taken with the exposure time e_t. I_{gt} is required for the training phase as well as for later performance evaluation purposes. Note that designating the under-exposed image of the each sequence as the source image is explained by the fact that under-exposed images contain generally less saturated areas than over-exposed images, thus offering more texture and details which in turn allows for a straightforward mapping operation. The same observation is valid for all methods we compare with, as it is harder to reconstruct lost information in over-exposed (saturated) areas.

The first dataset we use for our CNN-based color mapping model is the *Middlebury stereo dataset* [11]. This dataset offers the required source, target and ground-truth images alongside different exposure ratios between source and target images. Using the Middlebury set, we compose 2 different subsets with exposure ratios 4 and 16. The regrouped subsets are composed of respectively 17 images for training and 6 images for validation (testing), as shown in the examples in Fig. 4.2. The sequences from the ratio 16 subset are very challenging in the sense that they present large color differences between target and source images. In the following, we will refer to both subsets as *Ratio4* and *Ratio16* datasets.

Figure 4.2: Samples scenes from the *Ratio16* dataset used for training the proposed CNN-based color mapping model. The *target images* shown in the second row contain the desired target color distribution. Images courtesy of [11].

Additionally, we created a second dataset using an *IDS uEye* camera. Likewise, the captured dataset contains the required source, target and ground-truth images. For each sequence, we first capture I_s. Next, using the *SDK* of the camera, we externally increase the exposure time in order to capture I_{gt}. Finally, we capture the target image by fixing the corresponding exposure time and introducing motion in the scene. The captured dataset is in turn composed of 18 sequences for the training of the color mapping model, and 6 sequences for validation/testing purposes. We refer to this dataset as the *uEye* set. Samples from the training set are shown in Fig. 4.3.

An interesting feature of the uEye camera dataset is the fact that the source images were

Figure 4.3: Samples scene from the *uEye* dataset used for training the proposed CNN-based color mapping model. The *target images* shown in the second row contain the desired target color distribution.

captured with different exposure times (and respectively target images), unlike the *Ratio4* and *Ratio16* datasets. However, the exposure ratio between each pair of source and target images is fixed to 3. This implies that for this particular sequence, we specifically train a deep learning model for the ratio 3, independently on the exposure time of the source images.

4.2.2 Implementation and Network Details

As indicated in the previous sections, the CNN-based color mapping approach does not require any target image during the application phase. The proposed convolutional network takes I_s as input and applies a series of convolutions, in order to render the final color mapped image I_m. We set the number of the corresponding convolutional layers to 3, as suggested by [12]. We tested the performance of our network using additional convolutional layers, but we came to the conclusion that increasing the network size increases the execution time, with no proven improvement on the final results. A graphical illustration is provided in Fig. 4.4.

The first convolutional layer L_1 renders a total number of n_1 *feature maps* by applying n_1 convolutions on the input RGB image I_s, using n_1 filters with kernel size $c \times f_1 \times f_1$, where c is the number of color channels (3 in case of RGB). Next we apply a *Rectified Linear Unit* (ReLU) for the activation of the output of the first layer. These two operations can be written as

$$I_{s_1} = L_1(I_s) = max(0, W_1 * I_s + B_1), \tag{4.1}$$

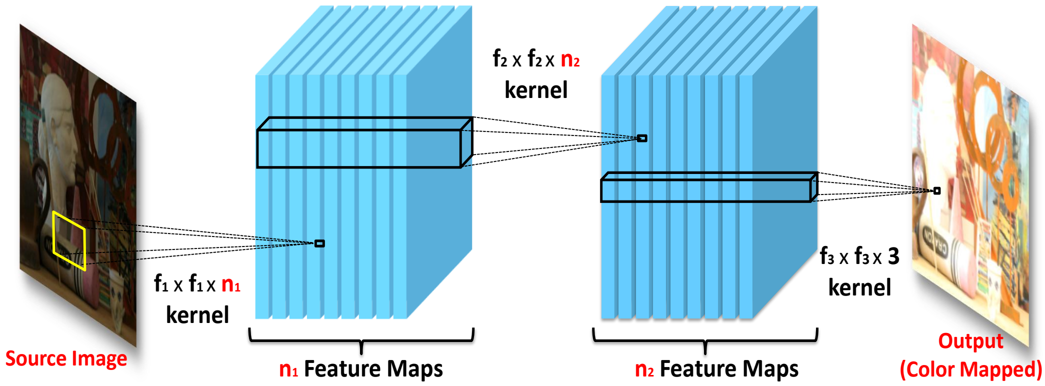


Figure 4.4: Representation of the CNN architecture used for the proposed color mapping approach. Note that this architecture is inspired from the work presented by Dong *et al.* in [12]. The shown source and output images are courtesy of the Middlebury stereo set [11].

where I_{s_1} represents the n_1 dimensional output of the first layer (feature maps), W_1 represent the set of $c \times f_1 \times f_1 \times n_1$ filter weights and B_1 is an n_1 dimensional vector of biases each corresponding to a filter. The first convolutional layer represents therefore each $c \times f_1 \times f_1$ patch in I_s through an n_1 dimensional vector. Note that we use a stride of 1 for the convolutions and we do not use zero-padding.

The second convolutional layer L_2 applies a non-linear transformation on the n_1 feature maps. This is achieved by performing a set of n_2 convolutions on each $n_1 \times f_2 \times f_2$ patch (or vector), which corresponds to the kernel size of each filter in the second layer. This is followed by the ReLu activation. In case f_2 is set to 1, the convolutions are applied in this case on a $n_1 \times 1 \times 1$ feature *vector*, which corresponds to the initial $3 \times f_1 \times f_1$ initial patch. However, in case f_2 is higher than 1, the second layer indirectly combines adjacent patches during the non-linear transformation, which is beneficial for the color mapping operation, as shown later. The second layer can be expressed through:

$$I_{s_2} = L_2(I_{s_1}) = max(0, W_2 * I_{s_1} + B_2), \tag{4.2}$$

where I_{s_2} is the rendered n_2 feature maps, W_2 and B_2 represent respectively the filter weights and biases of L_2. Again we use a stride equal to 1 and no zero-padding is applied.

The final layer L_3 renders the color mapped image I_m. This is done by applying n_3 convolutions on each $n_2 \times f_3 \times f_3$ patch from the n_2 feature maps I_{s_2}. Logically, n_3 is equal to 3 as it represents the number of output color channels in I_m. This can be written as:

$$I_m = L_3(I_{s_2}) = W_3 * I_{s_2} + B_3, \tag{4.3}$$

with W_3 and B_3 representing respectively the filter weights and biases of L_3. The third layer can be interpreted as an averaging step where each convolution on a $n_2 \times f_3 \times f_3$ patch from I_{s_2} yields the final RGB values of one pixel.

Understandably, the main task of the training stage is to learn the set of network parameters, namely the filter weights and biases $\mathscr{S} = \{W_1, B_1, W_2, B_2, W_3, B_3\}$ corresponding to each convolutional layer. During the training phase, these parameters are updated after each iteration according to the computed loss value between the intermediate mapped image I_m and I_{gt}. This enables to minimize the loss value of the next iteration. In our case we use the *Mean Squared Error* as loss function. The set of learnable parameters is updated by means of stochastic gradient (back-propagation). For this purpose, we set the corresponding learning rate to a relatively small value (10^{-4}), in order to avoid convergence to a local minimum. Note that for the training procedure, we provide the input source (data) and ground-truth (labels) images to the network as a set of cropped sub-images with a stride of 14, as suggested in [12]. The size of the data sub-images is set to 64×64, whereas the size of the labels varies according to the network architecture and is generally smaller than the size of the data sub-images.

4.2.3 Network Settings

In this section, we conduct several experiments in order to effectively select the most convenient network settings to the application at hand. These settings consist in the filter numbers n_1, n_2 and n_3 and the corresponding filter spatial sizes f_1, f_2 and f_3 of each convolutional layer. To achieve this, we first set the filter spatial sizes to $f_1 = 9$, $f_2 = 1$ and $f_3 = 5$, as suggested in [12], and compare the results gained from the various networks with different filter numbers on the *Ratio16* dataset. The gained results are presented in Table 4.1, where we compare the average PSNR values of each architecture over the full dataset, namely the training as well as the testing sets (total number of 23 images). Note that for these particular tests we down-size the images by a factor of 2, due to memory related issues especially for networks with large filter sizes and numbers.

Settings		**32-16-3**		**48-24-3**		**64-32-3**		**96-48-3**	
PSNR (dB)	**Time** (Sec.)	31.425	**0.73**	31.973	1.01	**31.956**	1.32	31.848	1.78

Table 4.1: Average PSNR values resulting from the comparison of several network settings with different *filter numbers*.

As shown in Table 4.1, using 64-32 settings provides the highest average PSNR value. Obviously, enlarging the number of filters induces a small increase of the average execution time. The choice of the network settings depends on the requirement of the application related to color mapping (short execution time or higher mapping quality). In this work, we select the network settings yielding the highest average PSNR, as the execution times of different configurations are very close.

In the second round, we fix the filter numbers to $n_1 = 64$ and $n_2 = 32$ and modify the spatial sizes f_1, f_2 and f_3 of the filters. As presented in Table 4.2, the combination of 9-3-5 achieves the highest average PSNR on the *Ratio16* dataset.

Settings		9-1-5		9-3-5		9-5-5		9-7-5		11-1-7	
PSNR (dB)	**Time** (Sec.)	31.956	**1.32**	**32.187**	1.84	31.975	2.33	31.761	4.75	31.679	2.10

Table 4.2: Average PSNR values resulting from the comparison of several settings with different *filter sizes*.

The slight gain of the average PSNR in favor of the combination 9-3-5 in comparison to 9-1-5 can be explained by the fact that increasing the spatial support of the non-linear mapping layer represents in indirect increase of the patch spatial size, which introduces further priors to the color mapping operation.

4.3 Experimental Results

In order to get a better understanding of the enhancement achieved by the introduced CNN-based approach and the support theory behind it, we compare the gained mapping results with state-of-the-art color mapping approaches. This includes the approaches of Reinhard *et al.* [72], Xiao and Ma [73], Pitie *et al.* [75], Pouli *et al.* [76] and the HM algorithm.

As illustrated in Fig. 4.5, our approach yields the best approximation to the ground-truth image despite the relatively high exposure ratio (16). The resulting image is artifact-free, unlike the images gained from other methods. This explains the $3.051\ dB$ lead in terms of PSNR of our approach over the second best result (HM). Note that the results gained from the approach of Pitie *et al.* [75] were generated without the suggested graining step, since it over-blurs the final image.

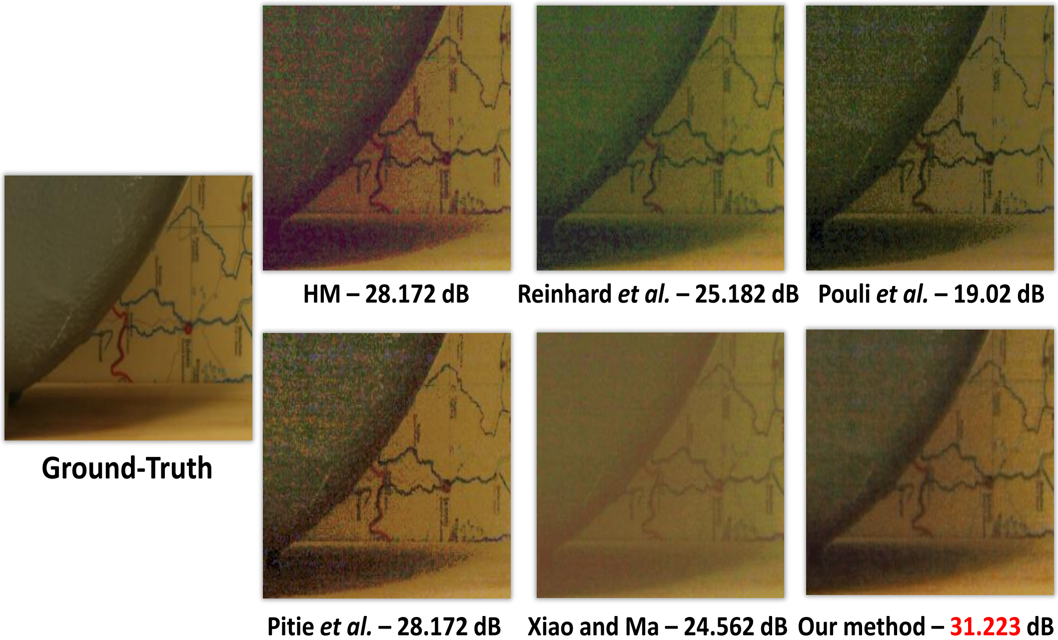


Figure 4.5: Zoom-ins on color mapping results from various methods including our approach. The results correspond to the 1^{st} *testing* sequence from the *Ratio16* dataset. Input images of the scene are courtesy of [11].

The visual as well as quantitative advantage in favor of our approach is also perceptible in Figures 4.6 and 4.7. Our color mapped image achieves a PSNR lead of respectively $5.003\ dB$ and $7.338\ dB$.

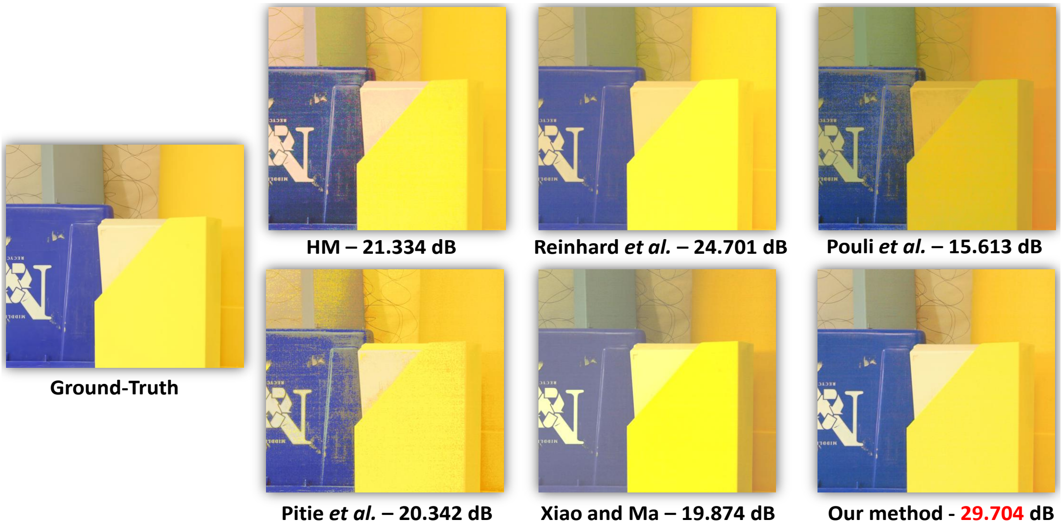


Figure 4.6: Color mapping results from various methods including our approach. The results correspond to the 6^{th} *testing* sequence from the *Ratio16* dataset. Input images of the scene are courtesy of [11].

Figure 4.7: Zoom-ins on color mapping results from various methods including our approach. The results correspond to the 5^{th} *testing* sequence from the *uEye* dataset. The yellow boxes indicate areas of faulty color mapping results.

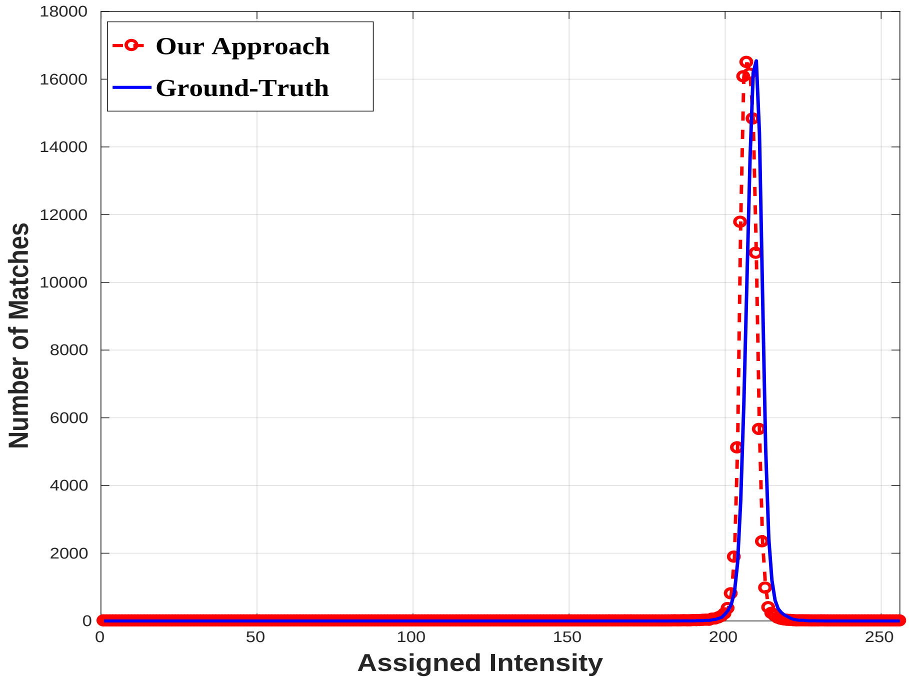


Figure 4.8: Mappings of intensity 40 (green color channel) generated using image pairs I_s and I_{gt} from *testing* sets of the *Ratio16* dataset (blue curve) as well as I_s and results of our approach I_m (red curve).

The noticeable color mapping quality of our approach is explained by the fact that our CNN-based model is able to approximate the Gaussian mapping model introduced earlier (see Fig. 4.1). An example of the accurate approximation is presented in Fig. 4.8, which illustrates the mapping of the intensity 40 (green color channel) using ground-truth (blue curve) as well as our results (red curve). The shown mappings are gained using the full set of *testing* sequences of the *Ratio16*.

These observations are confirmed by the results of the quantitative evaluation of the proposed approach as well as various additional methods, which are shown in Fig. 4.9. The PSNR results gained from the full-size *testing* images of the *Ratio16*, *Ratio4* and *uEye* sequences confirm the fact that our color mapping algorithm successfully handles the mapping problem. In fact, our approach yields better PSNR over almost all testing sequences except for sequence 5 in *Ratio4* set, where HM achieves a slightly higher PSNR value ($38.471\ dB$ using HM and $38.004\ dB$ using our approach). In addition, the average PSNR leads over the next best PSNR score for every sequence are $3.05\ dB$ for *Ratio16*, $3.68\ dB$ for *Ratio4* and $4.1\ dB$ for the *uEye* dataset.

Furthermore, the analysis of the results from Figures 4.5, 4.6 and 4.7 suggests that the CNN-based approach includes a denoising aspect as well. In order to verify this observa-

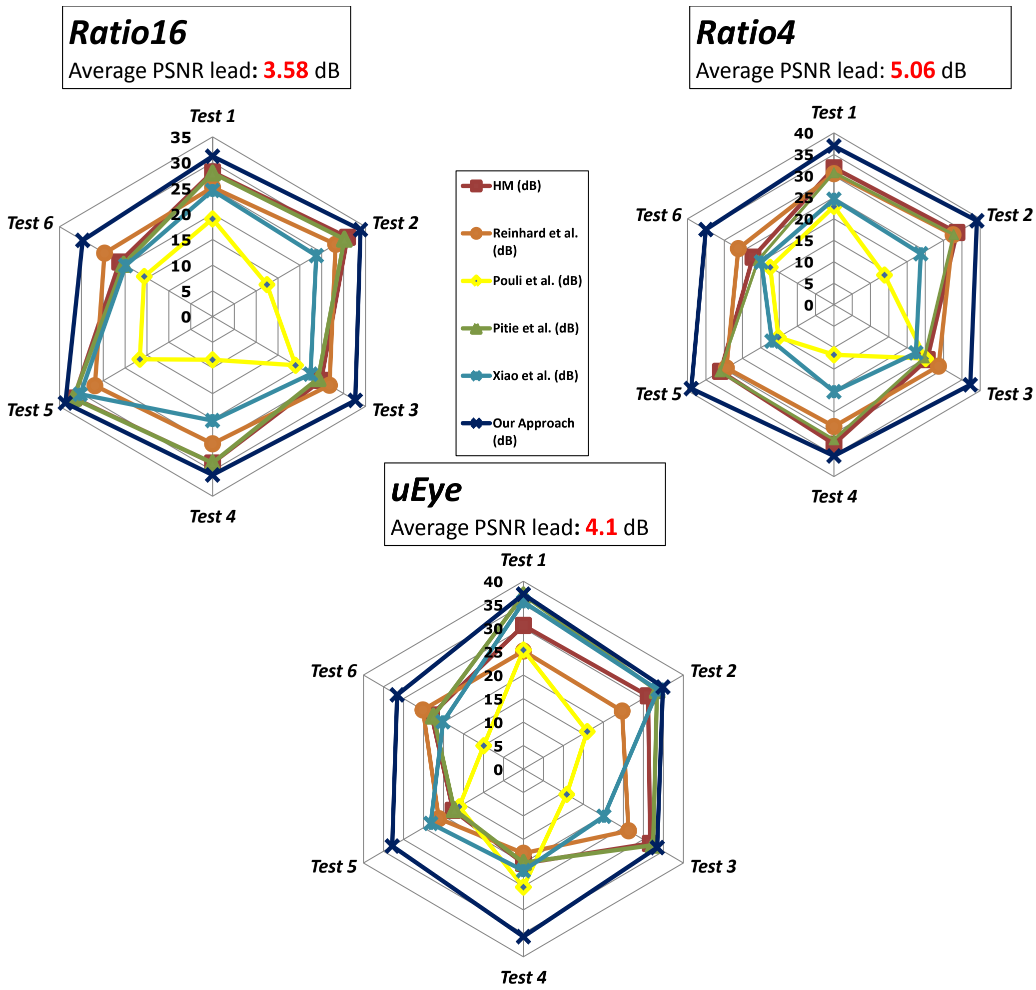


Figure 4.9: PSNR results of the testing sets corresponding to the *Ratio16*, *Ratio4* and *uEye* datasets.

tion, we apply the *BM3D* [18] denoising on the *second best* mapping results of each testing sequence in all datasets and compare the average PSNR values with our color mapping results (without denoising). The comparison results are presented in Table 4.3. Our approach still outperforms the denoised color mapping results of the second best approach.

Dataset	Second Best + BM3D, $\sigma = 5$	Second Best + BM3D, $\sigma = 10$	Our Approach
Ratio16 (dB)	31.720	31.800	**31.989**
Ratio4 (dB)	34.020	33.612	**36.929**
uEye (dB)	30.657	30.636	**34.283**

Table 4.3: Average PSNR values resulting from the comparison of denoising the second best color mapping approach with our mapping results.

4.4 Applications

4.4.1 Single Image HDRI

The improved color mapping quality has a particular significance for *single image HDRI*, since it directly influences the visual quality of the final output. In this context, our proposed CNN-based mapping model represents the perfect fit for single image HDRI, since no target image is needed for the mapping task. In order to assess the impact of enhanced color mapping on single image HDRI, we use the exposure fusion EF algorithm. The comparison starts with rendering the ground-truth HDR using I_s and the corresponding ground-truth image I_{gt} for each *testing* scene in all datasets. Next, we replace the ground-truth image by the color mapping results of the tested approaches (I_m), and generate the corresponding HDR, which we consequently .

Method	HM	Reinhard *et al.*	Pouli *et al.*	Pitie *et al.*	Xiao and Ma	Our Approach
Ratio16 (dB)	27.623	27.715	20.497	26.522	22.211	**31.151**
Ratio4 (dB)	32.826	34.928	26.927	32.399	27.334	**39.929**
uEye (dB)	28.802	28.257	21.750	30.492	27.857	**35.502**

Table 4.4: Average PSNR values resulting from the comparison of single image HDRs rendered using various methods with the rendered ground-truth HDR.

As shown in Table 4.4, the average PSNR values from the HDR images generated using our color mapped images are clearly higher than the average PSNRs of the remaining approaches. These results suggest that our color mapped images enable an accurate and high-quality expanding of the dynamic range of I_s with no additional priors from the target image such as color distribution.

4.4.2 Stereo Matching

One further application which benefits from high quality color mapping is *stereo matching*. Best performing state-of-the-art stereo matching algorithms are based on the brightness-constancy assumption between the input stereo images. In our tests we use the *Displets* stereo matching algorithm introduced by Guney *et al.* in [82], which is ranked first on the *KITTI Stereo Evaluation 2015* benchmark[1].

For each testing sequence from the stereo datasets *Ratio16* and *Ratio4*, we generate the left and right disparity maps between the color mapped image I_m and I_t, using the color mapping approaches under testing, including ours. The gained disparity maps are compared to the ground-truth maps provided by the *Middelbury* dataset. In case the ground-truth maps are not available, we render the missing maps using I_t and I_{gt} and label these maps as ground-truth. For the evaluation, we compute the average of the *Percentage of Bad Pixels* using 3 different thresholds (0.5, 1 and 2). The results of these comparisons are shown in Tables 4.5, 4.6, 4.7 and 4.8.

The disparity maps rendered using our color mapped image almost consistently contain the lowest average percentage of bad pixels, except for threshold 0.5, where we achieve very close results to the maps generated using the method of Reinhard *et al.* [72].

Method	**HM**	**Reinhard *et al.***	**Pouli *et al.***	**Pitie *et al.***	**Xiao and Ma**	**Our Approach**
Threshold 0.5 (%)	54.94	**52.78**	58.27	56.03	55.47	53.17
Threshold 1 (%)	26.56	22.58	30.37	26.84	25.7	**21.95**
Threshold 2 (%)	22.14	17.82	24.87	21.21	19.93	**17.56**

Table 4.5: Average percentage of bad pixels in the generated disparity maps – **Left** Disparity Map, **Ratio4** dataset.

Method	**HM**	**Reinhard *et al.***	**Pouli *et al.***	**Pitie *et al.***	**Xiao and Ma**	**Our Approach**
Threshold 0.5 (%)	54.39	**52.01**	57.49	55.48	54.89	52.45
Threshold 1 (%)	25.48	21.45	29.84	25.79	24.33	**20.85**
Threshold 2 (%)	20.92	16.68	24.35	20.34	18.59	**16.18**

Table 4.6: Average percentage of bad pixels in the generated disparity maps – **Right** Disparity Map, **Ratio4** dataset.

Method	**HM**	**Reinhard *et al.***	**Pouli *et al.***	**Pitie *et al.***	**Xiao and Ma**	**Our Approach**
Threshold 0.5 (%)	61.54	**59.85**	65.9	62.09	66.06	60.34
Threshold 1 (%)	35.28	32.48	40.74	35.41	40.6	**31.36**
Threshold 2 (%)	29.57	27.16	34.19	29.38	34.75	**25.93**

Table 4.7: Average percentage of bad pixels in the generated disparity maps – **Left** Disparity Map, **Ratio16** dataset.

[1] `http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo`

Method	**HM**	**Reinhard *et al.***	**Pouli *et al.***	**Pitie *et al.***	**Xiao and Ma**	**Our Approach**
Threshold 0.5 (%)	61.88	**59.66**	65.56	62	66.37	60.66
Threshold 1 (%)	35.6	31.56	41.41	35.52	40.54	**31.2**
Threshold 2 (%)	29.74	26.02	34.96	29.81	34.45	**25.26**

Table 4.8: Average percentage of bad pixels in the generated disparity maps – **Right** Disparity Map, **Ratio16** dataset.

4.5 Discussion

So far we have seen that CNN-based approaches can be successfully applied to low level computer vision and image processing tasks such as color mapping. The results gained based on the proposed CNN approach improve significantly upon state-of-the-art methods in terms of mapping accuracy. These results and the related observations pave the way for further investigations, which focus on two main questions:

- Is it possible to develop a CNN-based approach for HDR rendering on dynamic scenes, instead of color mapping? In other words, we are interested in creating an end-to-end HDR rendering model, which typically renders an HDR image based on a limited number of differently exposed LDR images with content difference due to motion. Evidently, the inherent difference between both applications (HDRI and color mapping) must be taken into consideration. For example, the HDR rendering model we seek to create must not be restrained by a specific exposure ratio between the input LDR images. Furthermore, the rendering model must exhibit strong generalization capabilities in order to make it relevant to as many scenarios as possible, such as different motion types and extreme exposure differences between the input LDR images. This explains the need for a more adequate architecture to the application at hand.
- It is possible to go beyond the CNN architecture we used for color mapping, so that it fits the HDR rendering task?

In order to provide meaningful answers to these questions, we first need to understand the limitations of the CNN architecture we used for color mapping, which was inspired from the work of Dong *et al.* introduced in [12]. In this architecture, the estimation of the final intensity of a specific pixel is based on an aggregation operation, which typically occurs in the final layer of the network. Therefore, the estimated intensity of a pixel results from the processing of several patches, which are spatially linked to that specific pixel. In other words, the intensity estimation does not account for global properties of the input images, and is subsequently computed based on the direct neighboring pixels. This was the intention behind using CNNs for the color mapping task, namely to take the immediate vicinity of each pixel into account while estimating its new intensity.

However in the context of HDR rendering, accounting for the global relationships between the pixels is very useful for the rendering of the final HDR, as the extension of the dynamic range should be consistent in all regions of the designated reference LDR image. In addition, global properties between the pixels over the stack of available LDR image are needed when dealing with motion-related constraints such as occlusions. In such cases, local dependencies do not provide enough priors for an accurate rendering of the HDR image in these areas. Finally, applying convolutions on full-size feature maps in every layer of the network creates a computational burden, especially during the training phase.

Based on these facts, there is a need for CNN architectures which concurrently provide the desired per-pixel predictions and enable the integration of global properties between the pixels. In this context, Fischer *et al.* introduced in [13] a novel CNN approach called *FlowNet* for the estimation of optical flow motion vectors between a pair of images. Similar to our HDR rendering application, optical flow computation requires a per-pixel estimation of the motion vectors. Furthermore, an accurate optical flow model needs to take into consideration the local and global properties of the input images. In order to achieve this, the proposed *FlowNet* architecture is composed of 2 main parts, namely a *contractive* part and a *refinement* part (also called *expanding* part). The contractive part of the network is typically composed of a succession of convolutional layers, each followed by an activation layer (ReLU). The refinement part is on the other hand composed of a sequence of de-convolutional layers. Likewise, each de-convolutional layer is followed by an activation layer. In addition, both parts of the *FlowNet* network are connected such that feature maps from the contractive part are forwarded to the refinement part.

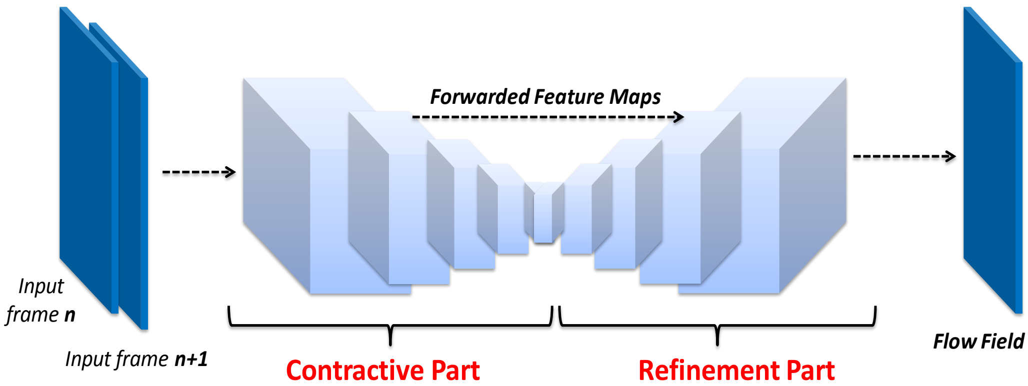


Figure 4.10: Representation of the *contractive* as well as the *refinement* parts of the *FlowNet* architecture, as introduced in [13].

The contractive part of the network extracts high-level features and spatially shrinks the feature maps. The down-sampling helps to effectively aggregate information over large areas of the input images. In order to recover the original resolution, the ensuing expanding part simulates a coarse-to-fine refinement operation by up-sampling to corresponding input feature maps and concatenating them with the forwarded feature maps from the contractive part. The approach of using *de-convolutions* for spatially up-sampling feature maps is inspired from the works presented previously in [83, 84, 85]. This allows a reliable recovery of the details lost during the contractive phase, which in turn results in a more accurate coarse-to-fine refinement operation. The final output is typically a dense per-pixel representation with the same 2D resolution as the input images. Fig. 4.10 contains an illustration of the topology of the previously described concept of the *FlowNet* architecture.

As shown in Fig. 4.10, the contractive and refinement parts of the *FlowNet* resemble pyramid representations. This not only enables the simultaneous integration of local and global information into the final refinement operation, as explained in [13], but also decreases the computational cost related to the learning of the network parameters.

In the following chapter, we aim at developing a CNN-based framework for the entire HDR rendering process chain. To achieve this, we initially experiment with a simple architecture inspired by *FlowNet*, in order to assess the applicability of machine learning on the task of HDR rendering. Based on the evaluation of the results gained from the initial tests, we propose several modifications to the CNN architecture so that it fits the requirements of the task at hand.

5 CNN-based HDR Rendering

Heretofore we approached the HDR rendering task for dynamic scenes from a ”conventional“ point of view, namely by proposing a low-complexity de-ghosting approach. However, and as we have seen from the evaluation of the results of our de-ghosting approach as well as other state-of-the-art methods, we are still looking for an HDR rendering framework which proposes the perfect fit between the computational cost of the enabling algorithm and the quality of the resulting HDR in terms of dynamic range extension, details recovery and most importantly suppression of motion-induced artifacts.

In this chapter, we propose to investigate the topic of HDRI for dynamic scenes from a completely different and novel point view, namely by taking advantage of latest advances achieved by CNNs in classification and segmentation topics as well as low-level image processing tasks such as super resolution processing and image de-noising. Additionally, we have seen in the previous chapter that CNNs can be successfully applied to the topic of color mapping, which represents an important building block of HDR rendering. Ultimately, the aspired HDR rendering model should be able to successfully handle the following scenarios:

- The input LDR images present large exposure differences. The HDR rendering model must handle cases where the input stack of LDR images is composed of only 2 images, which contain large saturated areas due to under-exposure and/or over-exposure. Furthermore, no pre-condition should be set on the exposure ratio between the input LDR images. This is crucial for the generalization aspect of the HDR model. In other words, the aspired HDR model needs to be able to handle cases where no prior information on the capturing settings of the input LDR images are available.

- In addition to the constraint related to the exposure difference between the input LDR images, the HDR rendering model is expected to handle various types of motion, be it camera-related (e.g. stereo capturing devices or handheld camera) or object-related motion, or simply a combination of both. In the context of object motion there exist several types ranging from rigid/non-rigid motion, slow/fast complex motion with several objects moving in the scene, objects appearing in a single LDR or a combination of these types.

With this in mind, we start by evaluating the performance of the *FlowNet*-inspired [13] architecture on the topic of HDR rendering for dynamic scenes. Based on the subsequent analysis of the gained results, we propose a new architecture that is more adequate to the application at hand. The proposed network is able to learn a function that maps multiple LDR images to an HDR image, thus enabling the extension of the dynamic range while successfully handling motion-related artifacts.

In the following sections we provide a detailed explanation of our approach and subsequently test it on several indoor and outdoor scenes, where we show that the quality of our results improves upon state-of-the-art approaches. We show as well that our approach is capable of handling extreme cases in terms of motion and exposure difference between the input images, while maintaining a very low execution time. This makes it suitable for average to low-end capturing devices. Finally, we assess the generalization performance of our model with respect to different types of motion. We show that our model can indeed handle free-motion scenes.

5.1 HDR Rendering Using FlowNet

5.1.1 Dataset

One critical component of CNN-based approaches is the dataset used for training. In this regard, an important attention should be given to the size of the training set, as well as the encompassed diversity in terms of the depicted scenes and exposure differences between the input LDR images.

In order to learn the mapping between the input LDR images and the corresponding HDR image, the corresponding training set typically consists of several scenes. Each scene comprises differently-exposed LDR images depicting various scene contents together with the corresponding HDR image, which is rendered from aligned but differently-exposed instances of one reference LDR image. The required setup for a *free-motion* dataset where all types of motion are included, is shown in details in Fig. 5.1.

Capturing differently exposed but aligned images in a sequential manner is a tedious task, especially in a non-controlled environment. This explains the reason why there are only a handful of datasets offering the required setup. Alternatively, stereo datasets are easier to create, especially when captured in a controlled indoor environment in order to avoid motion. The arrangement of stereo datasets is slightly different from that of free-motion datasets. This difference is illustrated in Fig. 5.2.

In our work, we combine 2 different datasets. The 1^{st} dataset is a combination of the 2005, 2006 as well as 2014 *Middlebury Stereo-sets* proposed by Scharstein and Szeliski in [11] and Scharstein *et al.* in [10]. The 2005 and 2006 sets contain several scenes composed of 3 differently exposed LDR images of the left view as well as 3 additional exposures of the right view. The exposure ratio between each consecutive pair of LDR images is 4. The 2014 set offers 6 different exposures of each view, where the minimum exposure ratio is 2. Fig. 5.3 shows 2 examples of the previously described scenes from the *Middlebury* stereo sets.

The *Middlebury* datasets enclose challenging scenes, especially in terms of exposure

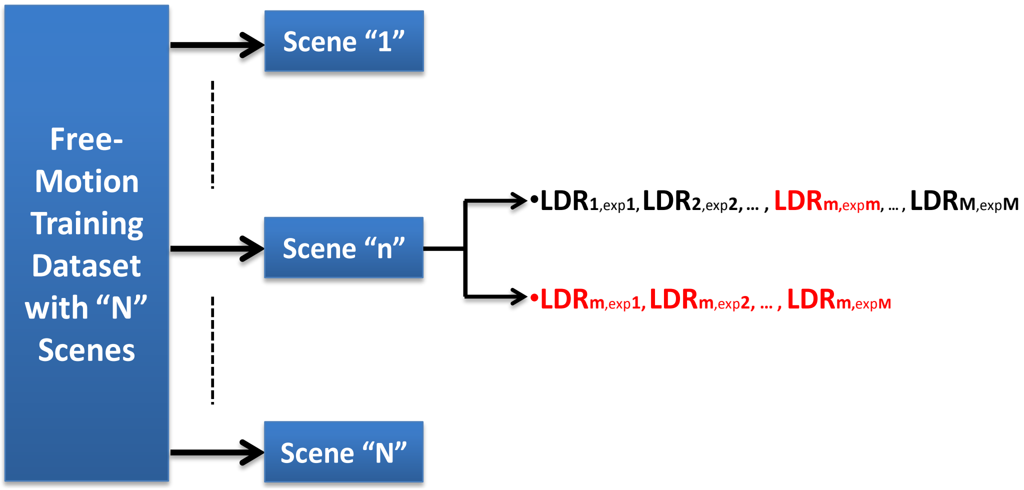


Figure 5.1: Illustration of the arrangement of a *free-motion* dataset as required for training an HDR rendering model. For example, an input pair of LDR images consists of LDR_{m,exp_1} and LDR_{2,exp_2}. The corresponding ground-truth HDR is composed from LDR_{m,exp_1} and LDR_{m,exp_2}. This means that LDR_m represents the reference LDR image.

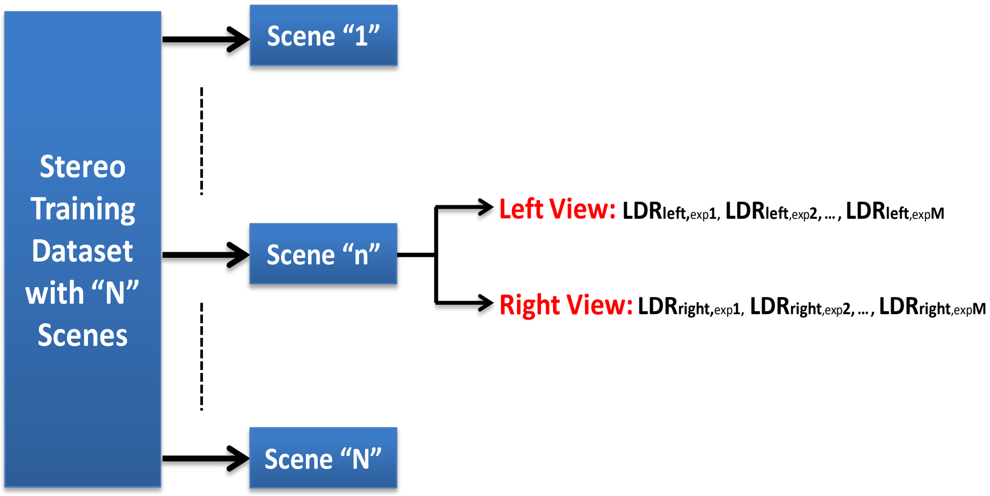


Figure 5.2: Illustration of the arrangement of a *stereo* dataset. For example, an input pair of LDR images consists of LDR_{left,exp_1} and LDR_{right,exp_2}. The corresponding ground-truth HDR is composed from LDR_{left,exp_1} and LDR_{left,exp_2}. This means that the reference LDR corresponds to the **left** view.

differences and saturated images (over and/or under-exposed), but lacks the desired diversity in terms of scene content as all the images in the dataset depict indoor scenes. However, a proper HDR rendering model needs to cover different scenarios including outdoor scenes. These are usually challenging for most HDRI systems as saturation due to over-exposure is very common. In addition, capturing differently exposed instances of the same scene is a problematic procedure, especially when the exposure settings need to be changed manually after each capture.

Figure 5.3: Sample scenes from the stereo datasets proposed by Scharstein and Szeliski in [11] (scene 1) and Scharstein *et al.* in [10] (scene 2).

To compensate for the lack of outdoor scenes, we created the second dataset. To this end, we used 2 *IDS uEye* cameras with identical parameters and settings (focal length, ISO sensor sensitivity, etc. ...). Using the **stereo** mounted cameras which are controlled via a software developed for this purpose, we are able to create several scenes. This is achieved by simultaneously capturing several LDR images using each camera (thus each view), while rapidly varying the exposure times from under-exposure to over-exposure, thus allowing us to capture differently-exposed instances of each view. The exposure settings are controlled by the software, so that updating the exposure time after each pair of images is done automatically. This way we avoid motion between consecutive frames of each view. Accordingly, each captured scene contains 5 LDR images of the left view and 5 additional LDR images of the right view. An illustration of the stereo setup used to create the outdoor dataset is presented in Fig. 5.4. In addition, Fig. 5.5 shows 2 different scenes captured using the *IDS uEye* as described previously.

We combine both datasets for the creation of the training set. This enables us to learn a

Figure 5.4: Stereo setup composed of 2 *IDS uEye* cameras used for the purpose of creating the outdoor dataset.

mapping on indoor and outdoor scenes simultaneously. Considering the designated setup where the reference LDR image corresponds to the left view and is darker than the non-reference LDR image, the upper bound for the number of possible combinations is equal to $(M_{scene_n} - 1)!$, where M_{scene_n} represents the number of LDR images in each view contained in scene n. Accordingly, the minimum possible ratio between each pair of left dark and right bright LDR images is equal to 2. This however can be problematic as there will be cases where both LDR images are either too dark or too bright at same time. Consequently, we only select LDR image pairs where the exposure ratio between the left dark and right bright images is at least 8. This guarantees that the trained model is capable of handling LDR image pairs with large exposure ratios.

Finally, we apply data augmentation on the combined training set in order to increase its size. The data augmentation consists of a vertical flip for each image. This way, we obtain 770 training pairs of *Dark Left* and *Right Bright* LDR images and the corresponding HDR version of the left view, which we render using the EF algorithm. This means that we label the left view as the corresponding reference. Additionally, 176 pairs of dark and bright LDR images are available for validation (testing) purposes. Evidently, images from the validation set are not included in the training phase of the network and are created for the sole purpose of evaluating the performance of the HDR rendering model. In this context, we make sure that the training and validation sets do not have any common scenes in order to ensure a reliable evaluation of the HDR model.

Scene 1 - Left Dark LDR (reference) | Scene 1 – Right Bright LDR

Scene 2 - Left Dark LDR (reference) | Scene 2 – Right Bright LDR

Figure 5.5: Sample scenes from the outdoor stereo datasets which we created using 2 *IDS uEye* cameras.

5.1.2 FlowNet-Based HDR

We experiment in this section with a basic *FlowNet*-inspired architecture composed of 3 convolutional layers in the contractive part, and 3 deconvolutional layers for the refinement part. This in turn implies that there exist 2 additional concatenations as output feature maps from layers 1 and 2 of the contractive part, which are forwarded as an additional input to the layers 5 and 6 in the refinement part. This architecture and the forwarded feature maps are shown in Fig. 5.6.

Note that we set the filter number for all convolutional and deconvolutional layers to 16, with a 4×4 convolutional (and deconvolutional) kernel. The stride s for all convolutions and deconvolutions is set to 2. This enables the down-sampling and up-sampling of the feature maps by a factor of 2, acting therefore as a pooling layer.

The input pairs of images are re-sized to 800×480 and presented to the GPU-training of the model as a concatenated input tensor. We use the *Caffe* framework [86] for the implementation of the layers in the *FlowNet* network. Furthermore, we set the initial learning rate to 10^{-3} with a "*Multistep*" learning policy. The learning is done based on the "*Adam*" optimization method, which is more suitable for machine learning tasks with a large number of network parameters, as thoroughly explained in [87].

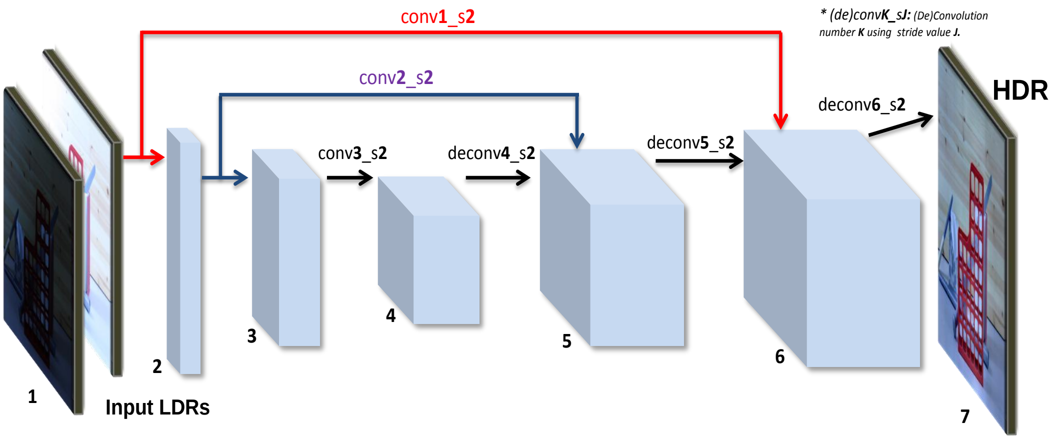


Figure 5.6: Illustration of the *FlowNet*-inspired [13] architecture used for creating the end-to-end mapping for the purpose of HDR rendering.

We evaluate the performance of the learned HDR model during the training phase, using for this purpose the validation set. This is done by averaging the PSNR values from all 176 testing sequences, resulting from the comparison of the HDR images rendered using the current model against available ground-truth HDR images. Note that during the selection procedure of the scenes, we made sure that the training and validation sets do not have any common scenes, in order to avoid parameter over-fitting. The values from the comparison are presented in Fig.5.7.

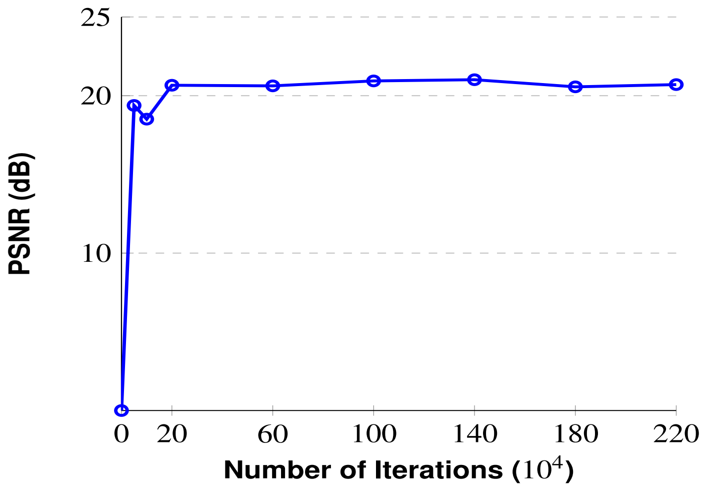


Figure 5.7: Average PSNR values resulting from the progression of the HDR model trained using the *FlowNet*-inspired architecture.

As shown in Fig.5.7, the training of the model converges rather fast. The average PSNR value settles around $20.5\ dB$. However, these values are relatively low over all tested iterations. This can be noticed as well when visually evaluating the resulting final HDR images, as presented in Fig. 5.8.

Although the HDR model significantly extends the dynamic range of the input left LDR image, clear *square*-shaped artifacts can be seen. These artifacts are related to a faulty reconstruction of the details from the down-sampled feature maps during the refinement stage of the network. This is in turn explained by the loss of details during the extraction of high-level abstractions in earlier stages of the network.

Moreover, the ambiguity of the HDR rendering task hinders the estimation of a comprehensive model. The HDR images used as labels (ground-truth images) differ in terms of dynamic range and details representation. They are also strongly dependent on the EF algorithm, whose performance is in turn determined by various factors such as the number, well-exposedness and saturation of the input LDR images, as well as the amount of motion in the depicted scenes.

Added to that, the quality of the ground-truth HDR images is limited in terms of details representation as well as freedom from artifacts. This can be seen on the ground-truth HDR image of the third scene in Fig. 5.8. The limited dynamic range and the related artifacts represent a typical consequence to the large exposure difference between the input LDR images, where EF struggles to reconstruct details in areas which are simultaneously under-exposed in the dark LDR image and over-exposed in the bright LDR image.

Considering all these observations, we propose several modifications to the *FlowNet* architecture used previously. The main goals of these modifications are:

- Reduction of the square-shaped artifacts in the rendered HDR images by providing more priors to the convolutional as well as deconvolutional layers.
- Decreasing the complexity of the HDR rendering problem by decomposing it to smaller sub-problems that are easier to model.
- Improving the HDR quality in terms of details and contrast for all possible scenarios including the cases where the input images present large content and exposure differences.

In the next section, we will discuss the modifications brought to the *FlowNet*-inspired approach, in order to effectively handle the previously mentioned points.

5.2 Proposed Modifications

5.2.1 Reducing Reconstruction Artifacts

One of the limitations noticed on the results using the *FlowNet*-inspired architecture is related to the loss of details in the contractive part, thus hindering an accurate reconstruction of these details during the subsequent refinement part. To deal with this limitation, we

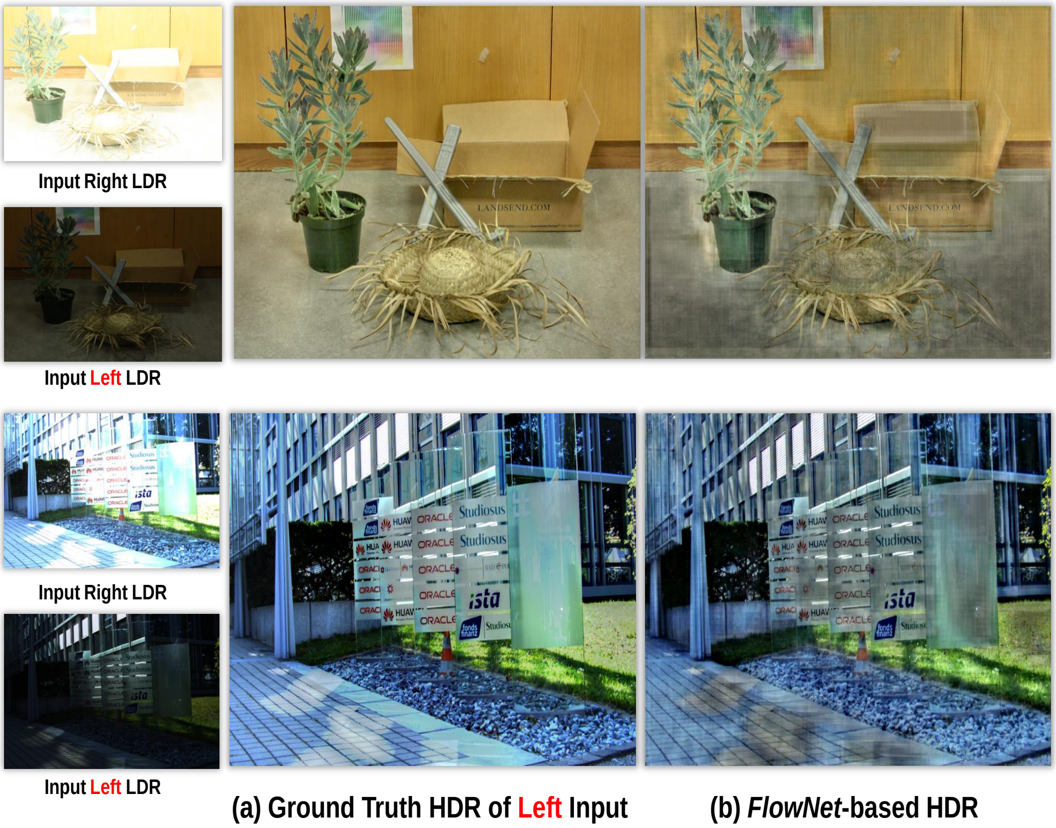


Figure 5.8: HDR rendering results on scenes from the validation set using the *FlowNet*-inspired architecture [13] (column (**b**), together with the corresponding ground-truth HDR image (column (**a**)). Although the rendered CNN-based HDR images have a greater dynamic range, they contain visible artifacts. Images from scene 1 (first row) are courtesy of [10].

propose to modify the *FlowNet*-inspired architecture by extending the range of concatenations, as suggested in our previous work presented in [14]. The proposed architecture, as shown in Fig. 5.9, is called *Fully-Connected-FlowNet* (FC-FlowNet). Note that the name "*Fully-Connected*" is not related to the known fully-connected layers.

Similar to *FlowNet*, *FC-FlowNet* is based on the symmetrical configuration comprising a contractive part and a subsequent refinement part. However, the difference to the original *FlowNet* lies in the additional links which reinforce the connection between both parts of the network. At each convolutional or deconvolutional layer, the feature maps representing different high-level abstractions from previous layers are combined (concatenated) into a single input chunk. This procedure is illustrated in more details in Fig. 5.10. In order to create multiple outputs with different spatial sizes in each layer, we vary the stride used for the convolutions and deconvolutions. This is necessary in order to match the dimensions of the target layers. The resulting feature maps are concatenated to the input of

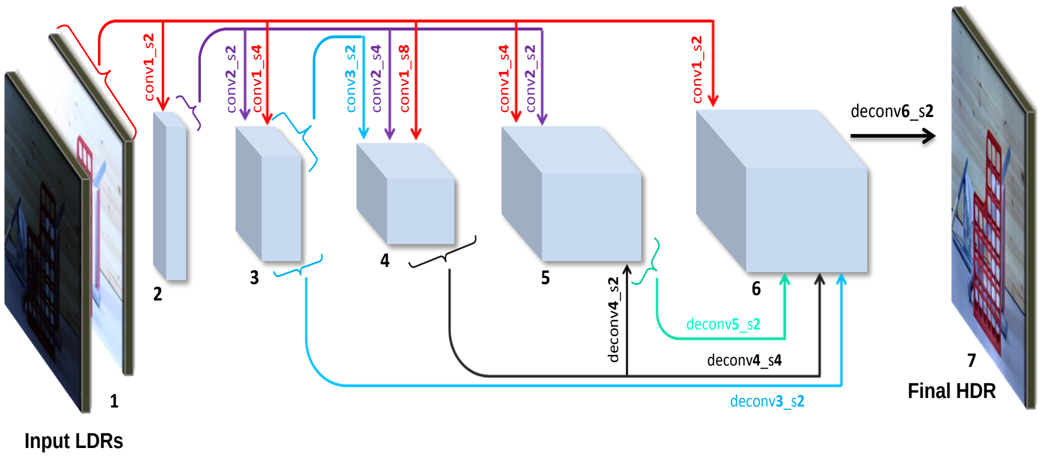


Figure 5.9: Depiction of the *Fully-Connected-FlowNet* architecture as proposed in [14]. The illustrated input LDR images are courtesy of [10].

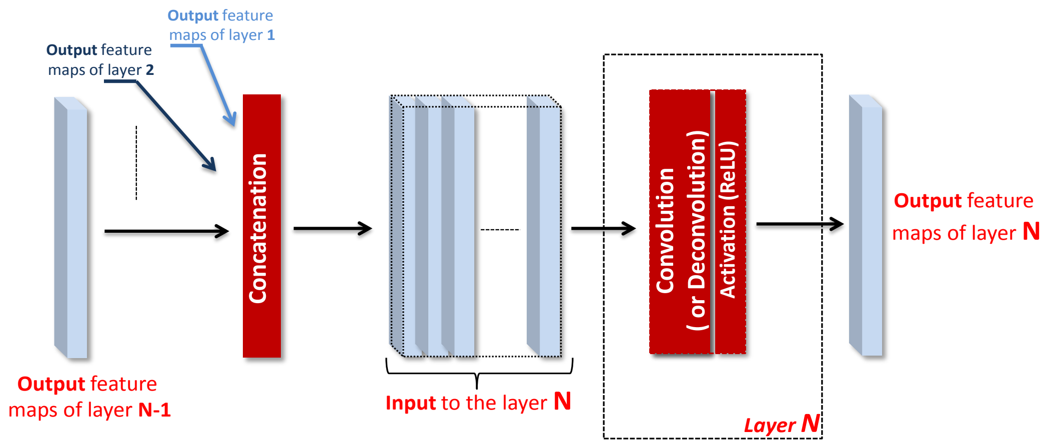


Figure 5.10: Representation of the approach used to concatenate different feature maps as a single input to the corresponding destination layer.

the corresponding destination layer. This results in that the depth of the input to each layer increases gradually when progressing through the network, as shown in Figures 5.9 and 5.10. Accordingly, the forwarded output feature maps enforce the redundancy of inherent information before each layer, which in turn enables a better recovery of the details at the output.

The additional information asserting redundancy improves the accuracy of the final coarse-to-fine step, since details are preserved and diffused through the network. In fact, forwarding feature maps from early layers, where few details were lost on processing, and connecting them to corresponding destination layers has the advantage of guiding the learning process to keep track of the very fine details, which in turn results in a more accurate reconstruction procedure. As a consequence, forwarding output feature maps representing different levels of abstractions enables a better integration of global as well as local properties of the input images, especially during the reconstruction operation.

5.2.2 Decreasing the HDR Ambiguity

As noticed on the results provided by the *FlowNet*-inspired architecture, learning a direct end-to-end mapping between the input LDR images and the HDR image of the selected reference image is a challenging task. Typically, The HDR model covers all possible scenarios. These scenarios might differ in terms of exposure ratio between the input images, type and amount of the depicted motion as well as the nature of the scene (indoor or outdoor scene, day or nature capture, etc...). In this context, the relationship between the input data composed of the LDR images and the corresponding output HDR image cannot be modeled in a unique function. Consequently, there exist the need to reduce the complexity related to the processing pipeline.

From this angle, numerous state-of-the-art HDRI approaches initially reduce the color difference between the input LDR images as a basis for further processing. With this in mind, there is a need for an approach which breaks down the desired end-to-end mapping between the input LDR images and the final HDR image, into representable functions, one of them typically taking care of reducing the color difference between the input images, hence performing color mapping. In this sense, we propose to split the CNN-based rendering task into 2 main sub-operations: **color mapping** and **HDR merging**. Each sub-operation is represented through a *FC-FlowNet* sub-network, where the output of the first sub-network is forwarded as an input to the second sub-network. In the following, we will refer to the proposed architecture as *Double-Loss FC-FlowNet*, as it contains 2 loss layers (for the training phases) for both connected sub-networks. A graphical illustration of the proposed setup is presented in Fig. 5.12.

The color mapping part learns the mapping model between the input LDR images (left dark and right bright). This results in an estimate of the **left bright** LDR image. Training such a model is possible since our dataset contains the differently exposed instances of each view, which we originally used to create the ground-truth HDR images. Next, the estimate of the *left bright* and the original input LDR images are concatenated and forwarded to the second sub-network. The goal of the HDR merging sub-network is to estimate the HDR image with respect to the left view based on these inputs. In this case, learning the rendering model is easier since 2 instances of the left LDR (under- and over-exposed) are available. In addition, tests have shown that providing the original right bright LDR image as an additional input to the second part of the network, together with both instances of the left image, improves the quality of the final HDR image.

However, we are interested in examining the feasibility of the color mapping task using the *FC-FlowNet*-based architecture, prior to integrating it into the framework of the HDR

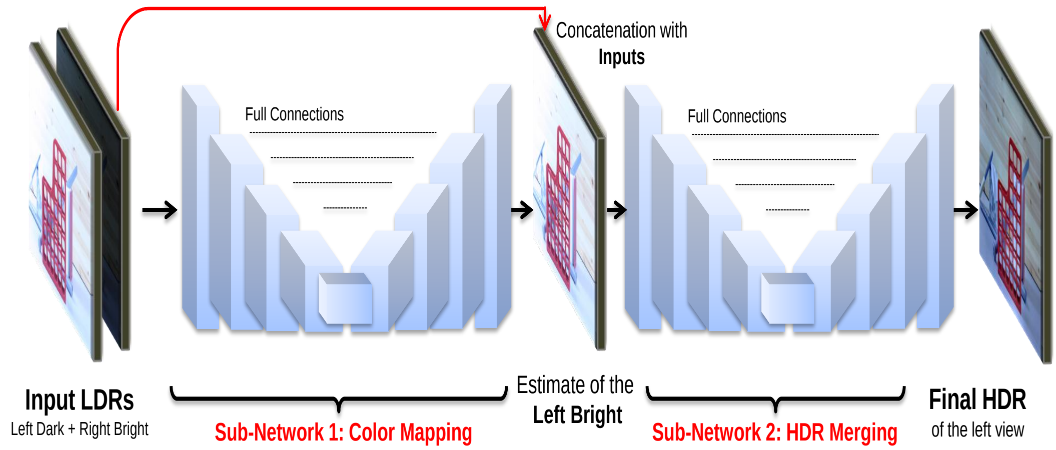


Figure 5.11: Illustration of the proposed ***Double-Loss FC-FlowNet*** architecture, which is composed of **color mapping** sub-network and a subsequent **HDR merging** sub-network. Input images are courtesy of [10].

rendering. For this purpose, we conduct several color mapping experiments in order using several datasets and configurations, as explained in the next section.

FC-FlowNet-based Color Mapping

In this section, we investigate the applicability of the *FC-FlowNet* architecture on the topic of color mapping, since it represents an important building block of the proposed *Double-Loss* network configuration. As thoroughly explained in Chapter 4, the main purpose of color mapping is to change the color properties of a *source image* so that it fits those of a *target image*. In this context, we aim at learning a proper end-to-end mapping between the source and target images, which simulates the desired color mapping operation. To achieve this, we conduct several experiments in order to cover all possible scenarios in conjunction with HDR rendering.

With this in mind, we start with 2 straightforward experiments where we seek to learn a model simulating a "*Dark to Bright*" color mapping. For these experiments, we use the *2014 Middlebury Stereo* dataset [10]. Table 5.1 contains a brief description of both experiments.

Experiment	**Exp. 1**	**Exp. 2**
Source Image	Left Dark	Left Dark
Target Image	Not Included	Included - Right Bright
Label Image (Ground-Truth)	Left Bright	Left Bright
Configuration	Dark to Bright	Dark to Bright
Exposure Ratio	Fixed Ratio (4)	Fixed Ratio (4)

Table 5.1: Details of experiments 1 and 2. In both experiments, our goal is to learn a model capable of performing color mapping on a **dark source** image (under-exposed), in order to estimate a **brighter** version of the source image.

In both experiments, we use the *FC-FlowNet* architecture in order to learn a color mapping between a dark source image and a corresponding brighter version of the same image (ground-truth used for the training of the model), as explained in Table 5.1. However, the difference between both experiments revolves around the *target image* (Right Bright). For instance, the target image is not included in experiment 1. On the other hand, we provide the target image as an additional input to the network in the second experiment. Accordingly, the target image is concatenated together with the source image. Hence, the input data to the network is a tensor comprising both source and target images.

The underlying idea behind using the target image is to explore the possibility of using information concerning its color properties. These information represent a valuable prior for learning the mapping function. Nevertheless, we are interested in assessing the impact of the perspective difference (parallax) related to the stereo nature of the input images.

The training set corresponding to both experiments is composed of 180 samples. In addition, 64 samples are available for validation/testing purposes.

We use an *FC-FlowNet* architecture with 5 convolutional layers in the contractive part, and 5 deconvolutional layers in the refinement part. We set the filter number of all layers to 32, and use a 4×4 kernel. In addition, we select a declining learning rate with an initial value equal to 10^{-3}, which decreases during the training following on a "*Multistep*" learning policy. The implementation of the networks is based on the *Caffe* framework [86].

In addition, we compare the results gained from both experiments against color mapped images from the state-of-the-art approach of Hu *et al.* [15]. As explained earlier, the method introduced in [15] aims at rendering an HDR image by mapping the colors of the non-reference LDR images to the selected reference LDR frame. The patch-based approach is based on the *PatchMatch* algorithm [44] for performing the color mapping operation. The comparison is based on the *Matlab* implementation provided by Hu *et al.*.

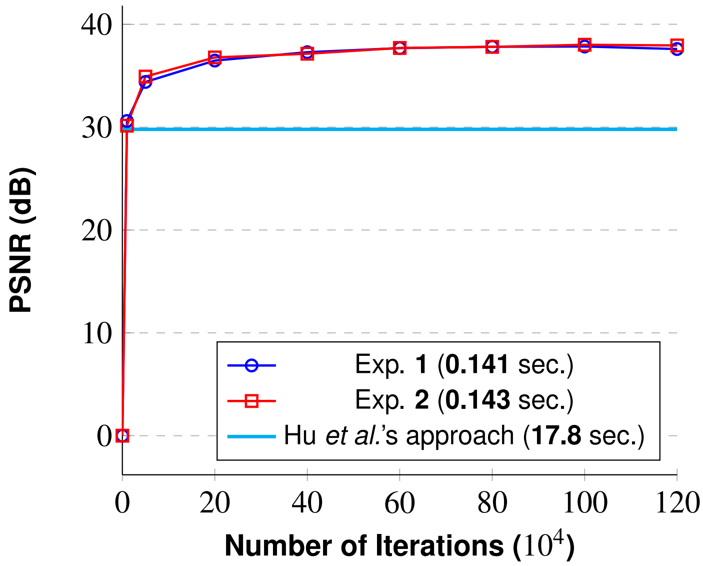


Figure 5.12: Evolution of the average PSNR values of experiments 1 and 2 on the validation set, together with the average PSNR value from the color mapping results as proposed by Hu *et al.* in [15]. In addition, we provide the corresponding average execution time. Note that we used the available *Matlab* implementation for the generation of the color mapping results of Hu *et al.*'s method.

The average PSNR values resulting from these experiments are shown in Fig. 5.12, together with the average PSNRs corresponding to mapping method proposed by Hu *et al.* in [15]. In both experiments, the learning of the model converges relatively fast, and settles around $37.5\ dB$. However, the difference in terms of performance between both experiments is small, with a slight lead in favor of the second experiment. In addition, the *FC-FlowNet*-based approaches clearly improve upon the results of Hu *et al.* by almost $8\ dB$ on average for the entire validation set.

Furthermore, Fig. 5.12 shows that the average execution times of experiments 1 and 2 (0.141 and 0.143 secs.) are considerably lower than the average execution time of Hu *et al.*'s color mapping approach (17.8 secs.). This can be explained by the fact that the approach of Hu *et al.* includes several building blocks which are time consuming, such as the patch-based correspondence search using *PatchMatch* or the multi-scale nature of the algorithm, using a pyramid-like approach [15]. Note that we conducted the execution time evaluation of Hu *et al.*'s approach based on their *Matlab* implementation using a computer with standard configuration.

The observations concerning the improvement in terms of color mapping accuracy based on Fig. 5.12 can be also confirmed when visually evaluating the resulting color mapped images, as shown in Fig. 5.13. The color mapping approaches of experiments 1 and 2, which are based on the *FC-FlowNet* network design, yield an accurate estimation of the corresponding target intensities. However, the color mapped image gained using Hu *et al.*'s method present blurring artifacts especially in the saturated areas, where the algorithm is seeking to reconstruct the corresponding details during the mapping operation based on similar regions in the target image.

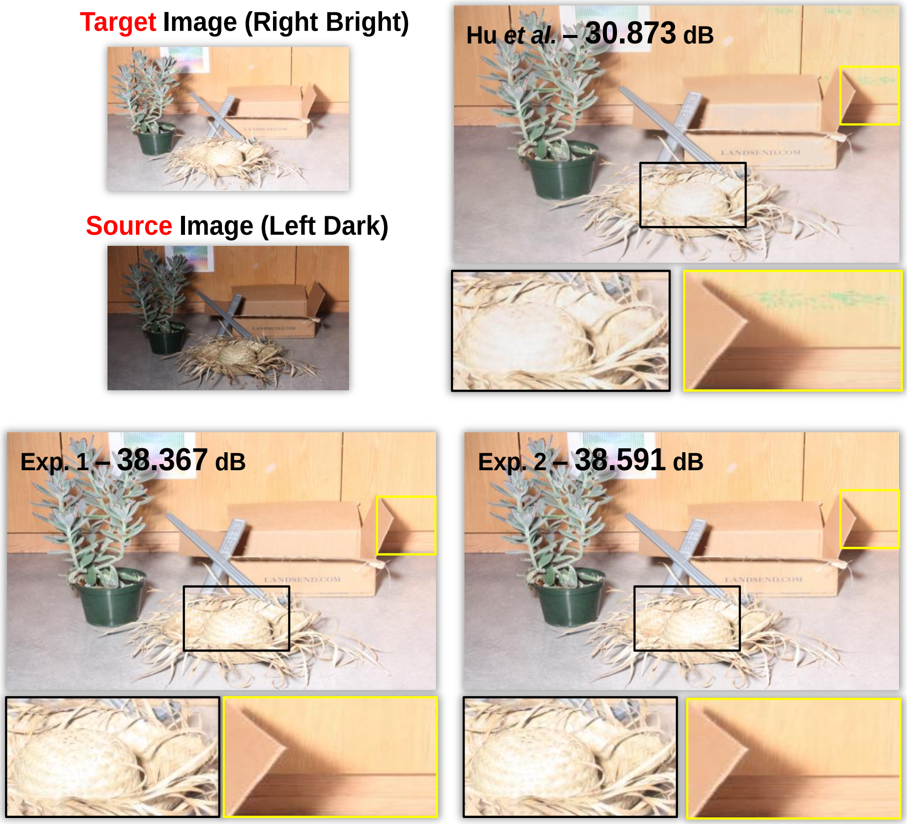


Figure 5.13: Example of color mapping results from experiments 1 and 2, together with the corresponding result of Hu *et al.* [15]. Our color mapping results achieve significant visual improvement in comparison to Hu *et al.*'s approach, where artifacts in saturated areas are visible. This improvement can be also perceived in the included PSNR values, where we achieve an 8 dB PSNR lead. Input images courtesy of [10].

Considering these results, we also are interested in examining the performance of the *FC-FlowNet*-based approach on the case where the source image is bright (over-exposed) and the target image is dark (under-exposed). As discussed previously, performing color mapping from bright to dark is a challenging task, as bright images typically contain large over-exposed areas. Accordingly, the reconstruction of the details in such areas is an ill-posed issue, as no prior information are available to guide the estimation. With this in mind, we perform 2 additional experiments to test the "*bright to dark*" color mapping operation.

Experiment	**Exp. 3**	**Exp. 4**
Source Image	Left Bright	Left Bright
Target Image	Not Included	Included - Right Dark
Label Image (Ground-Truth)	Left Dark	Left Dark
Configuration	Bright to Dark	Bright to Dark
Exposure Ratio	Fixed Ratio (4)	Fixed Ratio (4)

Table 5.2: Details of experiments 3 and 4. In both experiments, our goal is to learn a model capable of performing color mapping on a **bright source image** (over-exposed), in order to estimate a **darker** version of it.

The details of both experiments are shown in Table 5.2. Likewise, we aim at evaluating the impact of using the target image as an additional input (experiment 4) on the learned mapping model. We use similar network settings as the first set of experiments, in terms of number of filters, kernel size and other parameters.

The average PSNR values resulting from experiments 3 and 4 on the validation set are presented in Fig. 5.14, together with the values gained from Hu *et al.*'s algorithm. Evidently, experiment 3 yields the lowest PSNR values as it does not include the target image in the color mapping operation, as opposed to experiment 4 and Hu *et al.*'s approach. On the other hand, experiment 4 achieves the highest average PSNR values, with a lead over Hu *et al.*'s result by almost $2\ dB$. This confirms the assumption that in the case of "*Bright to Dark*" color mapping operation, providing the target image alongside the source image to the network enables a more accurate learning of the mapping model between the source bright image and the corresponding source dark image (label).

This observation can be also seen on the results shown in Fig. 5.15. The target image enables a more accurate estimation of the corresponding pixel intensities for areas which are very bright (over-exposed). This explains the large lead in terms of PSNR value in favor of experiment 4. In addition, the results gained from experiment 4 have no parallax-induced artifacts typically related to the stereo nature of the source and target images.

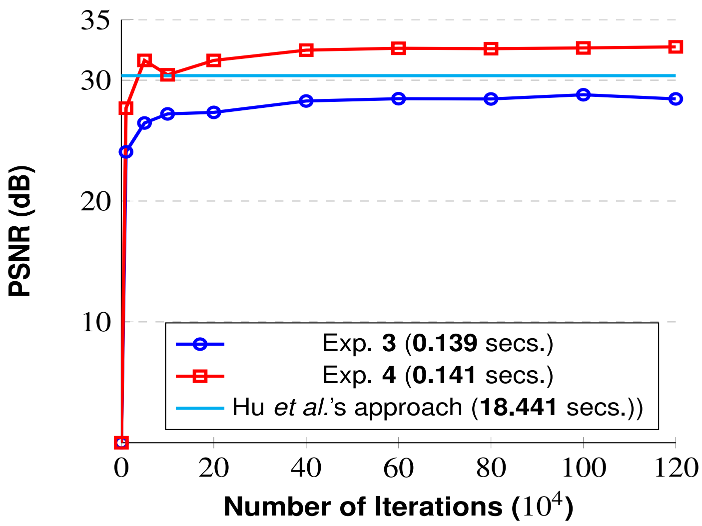


Figure 5.14: Evolution of the average PSNR values of experiments 3 and 4 on the validation set, together with the average PSNR value from the color mapping results as proposed by Hu *et al.* in [15]. In addition, we provide the corresponding average execution time. Note that we used the available *Matlab* implementation for the generation of the color mapping results of Hu *et al.*'s method.

In addition to these results, there is a need to investigate one final important case, namely where the exposure ratio is **variable**. So far we conducted our experiments on **fixed** exposure ratios between the source and target images. It is however crucial to make sure that the HDR rendering model we seek to develop is independent of the exposure ratio between the input LDR images for the sake of generalization performance. As a matter of fact, learning a color mapping model for variable exposure ratio cases is feasible as a result of including the *target image* into the processing framework. The target image acts therefore as a *guide* to the color mapping model, by providing information concerning the target color properties. Understandably, in case the target image is not included (e.g. experiments 1 and 3), the learned model is unable to guess the target exposure time.

To demonstrate this, we perform a final experiment using the same dataset we described previously in Section 5.1.1. Evidently, the corresponding label images in this case are the left bright versions of the source image. As mentioned earlier, this dataset offers the desired diversity in terms of scenes (indoor and outdoor scenes) as well as the corresponding exposure ratio between the input pairs of images. Accordingly, the minimum exposure ratio is set to 8. The dataset offers a total number of 770 image pairs with the corresponding labels which we use for learning the model parameters (training), in addition to 176 sequences for validation purposes. The configuration details of the last experiment are shown in Table 5.3.

the comparison of the average PSNR values is shown in Fig. 5.16. Once it converges

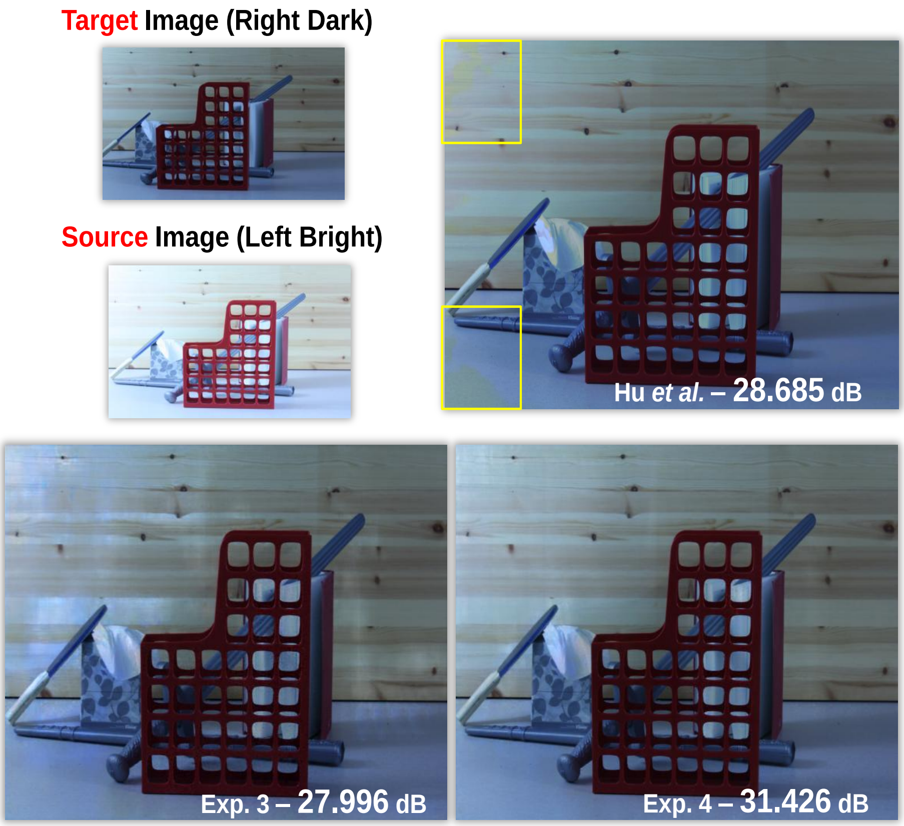


Figure 5.15: Color mapping results from experiments 3 and 4, alongside results from HM. The results gained from experiment 4, where the target image is provided as an additional input, evidently improves upon the results from experiment 3. This is especially the case for the over-exposed bright areas. The included PSNR values confirm this observation. Input images courtesy of [10].

Experiment	**Exp. 5**
Source Image	Left Dark
Target Image	Included - Right Bright
Label Image (Ground-Truth)	Left Bright
Configuration	Dark to Bright
Exposure Ratio	Variable

Table 5.3: Details of experiment 5. The main goal of this experiment is to learn a mapping model capable of performing color mapping on multiple exposure ratios simultaneously.

and settles around $32\ dB$, our mapping model constantly outperforms Hu *et al.*'s color mapping approach by approximately $5\ dB$. In addition to the improved color mapping accuracy, the average execution time of our method is relatively low (0.202 secs.), especially

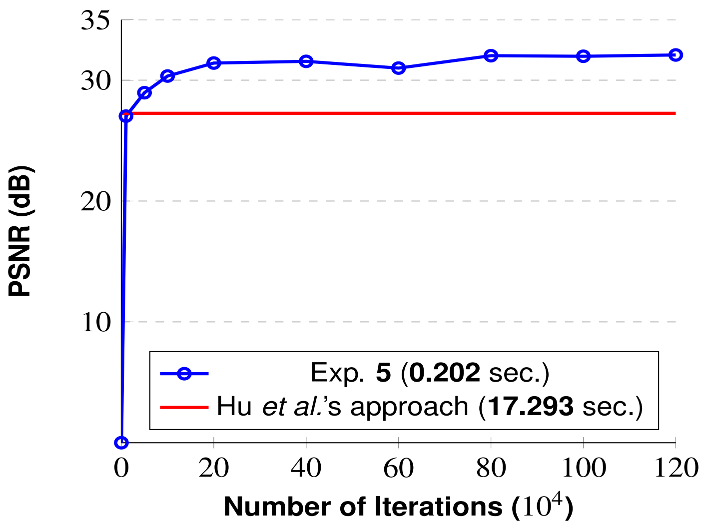


Figure 5.16: Evolution of the average PSNR values of experiment 5 on the validation set, together with the average PSNR value from the color mapping results of Hu *et al.*'s approach introduced in [15]. In addition, we provide the average execution times needed to perform color mapping on the validation set composed of 176 image pairs.

when compared to the required average computation time by the approach of Hu *et al.* (17.293 secs.).

These observations can be confirmed as well when visually evaluating the quality of the color mapped images. Fig. 5.17 contains 2 sample results from of our approach, together with the results based on Hu *et al.*'s method. These results confirm the assumption that the included target image guides the mapping model to the target exposure time. In addition, our results do not show any artifacts related to the stereo nature of the scenes and the induced perspective shift between the source and target images. On the other hand, the results of Hu *et al.* present large blurred areas especially in the second scene (areas highlighted by the yellow boxes). This is indeed a direct result to the fact that the method of Hu *et al.* aims at reconstructing under-exposed areas in the source image based on the target image, using for this purpose the *PatchMatch* algorithm. However, in cases where the exposure ratio between the source and target images is large, this strategy results in clear artifacts, as the PatchMatch-based reconstruction fails at finding regions in the target image which are similar (in terms of texture) to the under-exposed areas in the source image.

Figure 5.17: Illustration of color mapping results using the *FC-FlowNet* architecture (experiment 5) as well as the approach proposed by Hu *et al.* in [15], together with the corresponding ground truth (brighter version of the source left dark image). Results from the 5^{th} experiment exhibit enhanced accuracy in terms of pixel intensity estimation, while being almost artifact-free. The yellow boxes highlight areas where the results of Hu *et al.* contain visible artifacts. Input images of scene 1 are courtesy of [10].

5.3 Experiments

In this section, we conduct several experiments in order to evaluate the impact of the modifications introduced earlier and summarized in the *Double-Loss FC-FlowNet* network architecture.

In all subsequent HDR experiments, we use the same network settings. Based on the *Caffe* framework [86], we set the depth of the *color mapping*-related sub-network to 5 convolutional layers and 5 corresponding deconvolutional layers, using the *FC-FlowNet* design. For this sub-network, we set the filter numbers of all layers to 32. The HDR-merging sub-network is in turn composed of 3 convolutional and 3 deconvolutional layers, according to the *FC-FlowNet* configuration, with a corresponding number of filters equal to 16. Furthermore, we set the initial learning rate to 10^{-3} which decreases using a *multistep* learning policy. In addition, we fix the *Weight-Decay* value to 4.10^{-4}, the momentum to 0.9 and the gamma value to 0.5. We select the *Euclidean Loss* as the designated loss-layer for both networks and use the "*Adam*" optimization method. The input images are provided to the *Double-Loss FC-FlowNet* network as batch of 2 images with a resolution of 480×800. The final HDR image is expected to be of same spatial resolution as the input images.

HDR Experiment 1

The purpose of the first experiment is to compare the performance of the proposed *Double-Loss FC-FlowNet* architecture against the *FlowNet*-inspired variant, in order to establish a clear evaluation of the modifications introduced earlier. This means that during the first HDR experiment, we seek to develop an HDR rendering model capable of estimating the HDR image of a pair of stereo input LDR images, where the reference LDR image is set to be the **left dark LDR**. Accordingly, the corresponding non-reference input image is the right bright LDR image.

Furthermore, we generate the corresponding HDR images using the approach proposed by Hu *et al.* in [15]. The underlying idea is to align the input LDR images colorwise. Once the alignment through color mapping is done, the final HDR image of the reference is rendered using the EF algorithm by merging the reference LDR and the corresponding color mapped image which have the view of the reference LDR and the color properties of the non-reference LDR image.

With this in mind, we monitor the progress of the average PSNR values during the learning phase of the network parameters (training). The PSNR values are gained from the comparison of the results of each method against the ground-truth HDR, which we generate using left dark as well as left bright LDR images and based on the EF algorithm, similar to the generation of the label HDR images for the training set. Evidently, the evaluation of the HDR model is based on the validation set. The results of this evaluation are presented in Fig. 5.18. The HDR rendering model trained using the *Double-Loss FC-FlowNet* architecture converges relatively fast, and stabilizes around $25.3\ dB$. This represents an approximately $4\ dB$ improvement in comparison to the average PSNRs using the *FlowNet*-inspired architecture. On the other hand, The HDR images gained by performing EF on the aligned LDR images using the approach of Hu *et al.* achieve an average PSNR value of $30.215\ dB$ for an average execution time of 17.293 seconds. Note that the average execution time corresponds to the LDR alignment step (color mapping), excluding therefore the EF part. Despite achieving better average PSNR value than HDR experiment 1, the comparison is not equitable. In theory, performing EF on the outcome of the first sub-network responsible for color mapping (see Fig. 5.12), would achieve an average PSNR value equal to $30.471\ dB$, which is higher than the average PSNR values corresponding to the HDR images gained using Hu *et al.*'s approach, while achieving a relatively low average execution time. This can be can be explained by the fact that performing color mapping using an *FC-FlowNet* network architecture yields better mapping results than the state-of-the-art approach of Hu *et al.*, as shown previously in Figures 5.16 and 5.17. However, the main purpose of this work is to provide a comprehensive and inclusive end-to-end mapping, where we are able to directly generate an HDR image from an input pair of differently exposed LDR images. In addition, we are interested in investigating whether the HDR merging sub-network and the subsequent gained final HDR image would benefit from the CNN framework we set for this application, as we will see in the following section.

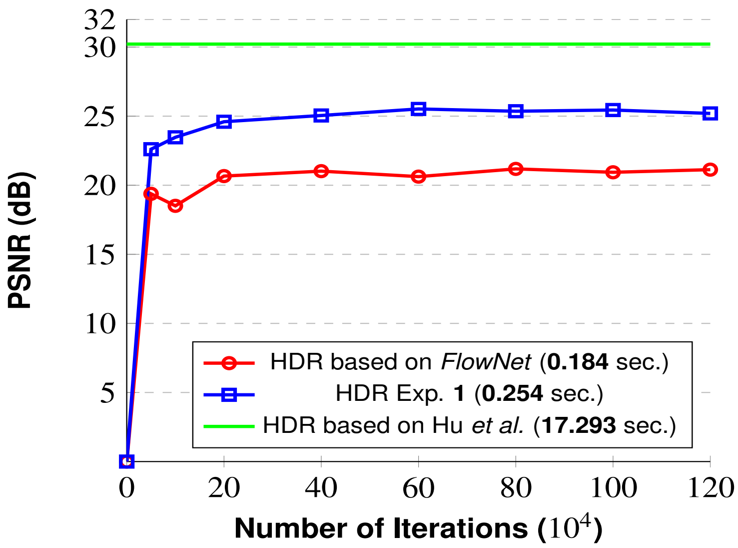


Figure 5.18: Average PSNR values resulting from the comparison of the progression of the HDR model trained using the *Double-Loss FC-FlowNet* architecture (HDR experiment 1), together we the results gained previously using the *FlowNet*-inspired architecture as well as the HDR images rendered using Hu *et al.*'s method. In addition, the figure contains the average execution times of the tested approaches. Note that the average processing time of Hu *et al.*'s approach concerns only the alignment step (color mapping), excluding therefore the EF step.

Furthermore, we aim at assessing the visual enhancement related to the *Double-Loss*

FC-FlowNet architecture as well. In Fig. 5.8, we presented 2 examples where the HDR model trained using the *FlowNet*-inspired architecture creates visible artifacts, which significantly decrease the visual quality of the gained HDR image. Fig. 5.19 shows the updated results on the scenes presented previously in Fig. 5.8. The HDR gained using the *Double-Loss FC-FlowNet* architecture does not contain any artifacts related to the recovery of details during the refinement part of the network. This in turn impacts positively the overall quality of the HDR images, as perceived also in the enclosed PSNR values.

Nevertheless, a closer look at the results presented in Fig. 5.19 brings us to the conclusion that the overall dynamic range in the gained HDR images is limited. In fact, the generation of the ground-truth HDR images used for validation and training is based on the EF algorithm. Apart from its relative straightforwardness and performance stability, EF does not require any priors on the input stack of LDR images, such as the exposure ratio. However, in the case of 2 input LDR images with significant exposure difference and large saturated areas, EF performs poorly.

Considering this observation, we seek to go beyond the dynamic range available in the input LDR images in order to render a high quality final HDR image. This is a critical feature of the HDR rendering model we aim at developing, as it has to be able to deal with extreme cases in terms of exposure ratio between the input LDR images. Such cases are very challenging as conventional methods for HDR rendering yield unsatisfactory results. In the next experiment, we propose a modification to our process chain, which enables us to achieve better HDR quality even in extreme cases.

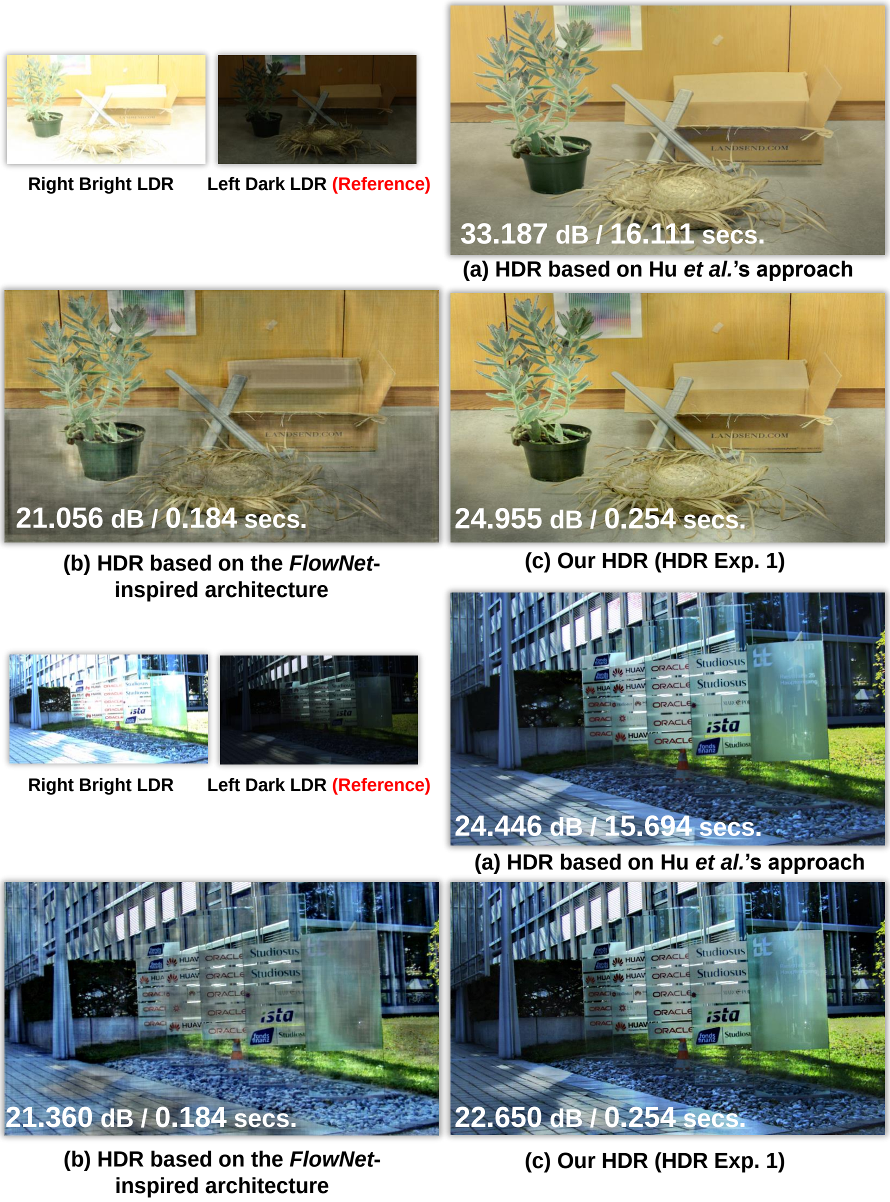


Figure 5.19: HDR rendering results on scenes from HDR experiment 1 using the *Double-Loss FC-FlowNet* architecture (**c**) together with results of the *FlowNet*-inspired architecture [13] (**b**) and the corresponding HDR images rendered using Hu *et al.*'s approach [15] (**a**). The HDR images gained from HDR experiment 1 significantly improve upon the results gained using the *FlowNet*-inspired architecture, as the square shaped reconstruction artifacts were successfully eliminated. Images of scene 1 are courtesy of [10].

HDR Experiment 2

As noticed in the results gained so far, training on low-quality HDR images with limited dynamic range impacts negatively the performance of the HDR rendering model. To solve this problem, we propose in the second HDR experiment to use the **full stack of available LDR images** for the generation of the corresponding label images (ground-truth HDR images used for training). For example, in the case of the outdoor dataset that we created using the *uEye* cameras, the ground-truth HDR image for each scene is gained from merging the 5 differently exposed instances of the left view (reference). This way, our model does not only render a ghost- and artifact-free HDR, but also simulates the case where more than 2 input LDR images are available. This allows us to deal with very challenging cases in terms of exposure differences and number of input images. Fig. 5.20 shows the quality difference between the 2-images ground-truth HDR and the 5-images ground-truth HDR.

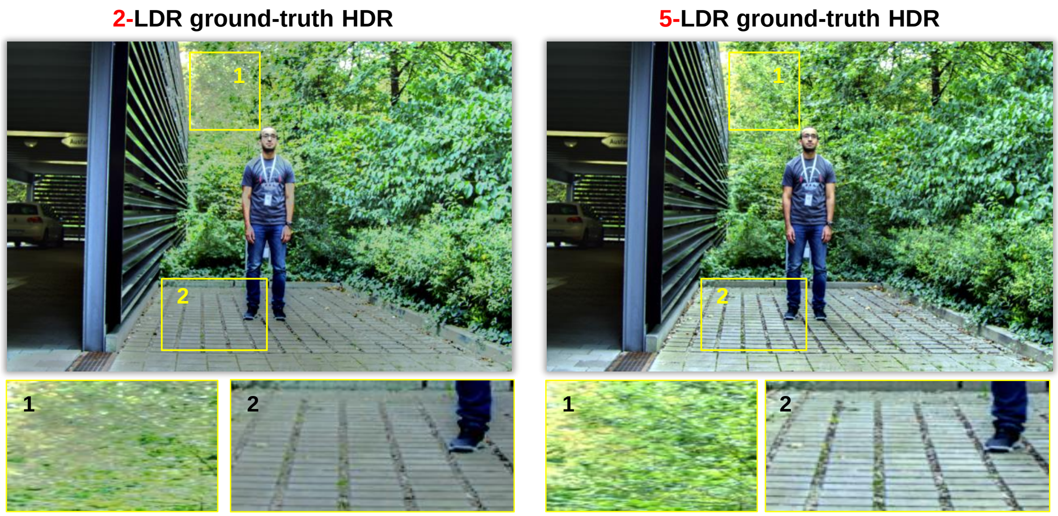


Figure 5.20: Illustration of the difference between the 2-LDR based ground-truth HDR and the 5-LDR based ground-truth HDR on an outdoor scene. Clearly, the 5-LDR based HDR image is more suitable to train the desired HDR rendering model as it disposes of a larger dynamic range. This means that it contains more details of the depicted scene, notably in the saturated regions (due to under- or over-exposure) as shown in the zoomed-in areas.

We use the same network settings as the first HDR experiment. Note that at this stage, an objective comparison by means of PSNR computation between the outcome of HDR experiment 2 and Hu *et al.*'s approach is no longer possible, as the ground truth HDR images needed to compute the corresponding PSNRs are different in both cases (full stack based HDR in experiment 2 and 2-LDR based HDR in the case of Hu et al.'s approach). Accordingly we rely on the subjective (visual) evaluation of the gained results.

Figure 5.21: Updated results on the scenes showed previously in Figures 5.8 and 5.19. The images resulting from experiment 2 depict a larger dynamic range than the results of experiment 1 as well as the HDR images gained based on Hu *et al.*'s approach. This results in an enhanced quality, especially in terms of details, all while keeping the required average execution times very low. Images of scene 1 are courtesy of [10].

Using the learned model in HDR experiment 2, we evaluate the impact of last modification on the same scenes shown previously in Fig. 5.19. The updated results are illustrated in Fig. 5.21. The HDR images gained using the rendering model from HDR experiment 2 achieve a higher visual quality despite the relatively large exposure difference between the input LDR images, as well as the inherent difference in perspective. In fact, our **Double-Loss FC-FlowNet**-based HDR model enables the depiction of a wider range of details of the scenes for well-exposed regions as well as saturated regions. In addition, the extension of the dynamic range is very realistic, as the general contrast of the gained HDR images is well balanced. In other words, these images look like as if they were captured using an HDR-capable camera sensor. Evidently, the improvement of the HDR rendering quality is achieved without impacting the corresponding execution time, as the average computation time is equal to $0.262\ sec$.

To confirm these observations, Fig. 5.22 contains a second set of comparisons between the HDR results of experiment 2 as well as the HDR images gained from the *FlowNet*-inspired architecture and the approach of Hu *et al.*.

Likewise, the examples shown in Fig. 5.22 support the observations made previously. The HDR images rendered using the learned model from the second experiment depict a much wider dynamic range than the corresponding reference image. This is achieved while no artifacts related to the difference in perspective between the input LDR images can be seen in the final image. The model gained from HDR experiment 2 allows to use the well-exposed parts of both input images for the recovery of the details in the under- and/or over-exposed areas. Consequently, our model is capable of handling extreme cases with large exposure and scene differences, as opposed to existing state-of-the-art approaches such as the method of Hu *et al.*, which exhibits limited performance in such cases. Additionally, the run-time requirement of the developed HDR model is very low, which makes it suitable to applications and devices with limited computational capabilities.

In this context, we are getting closer to fulfilling the requirements mentioned in the introduction to this Chapter concerning the aspired HDR rendering model. So far, the *Double-Loss FC-FlowNet* architecture trained on the full stack of the available LDR images is able to render an HDR image from an input pair of LDR images, which have various degrees of differences in terms of exposure settings (thus amount of depicted details and saturated regions) and content (indoor and outdoor scenes), all in a very low amount of time and with no prior knowledge of the capturing parameters. Nevertheless, the HDR experiments conducted so far are focusing on a single type of motion which can be categorized as camera motion due to the stereo nature of the capturing setup. Accordingly, there exist the need to investigate other types of motion, in order to evaluate the generalization performance of the learned HDR model.

With this in mind, we seek in the next HDR experiment to extend the range of tested examples to the case of *free-motion*, which includes all possible types of camera and/or scene motion.

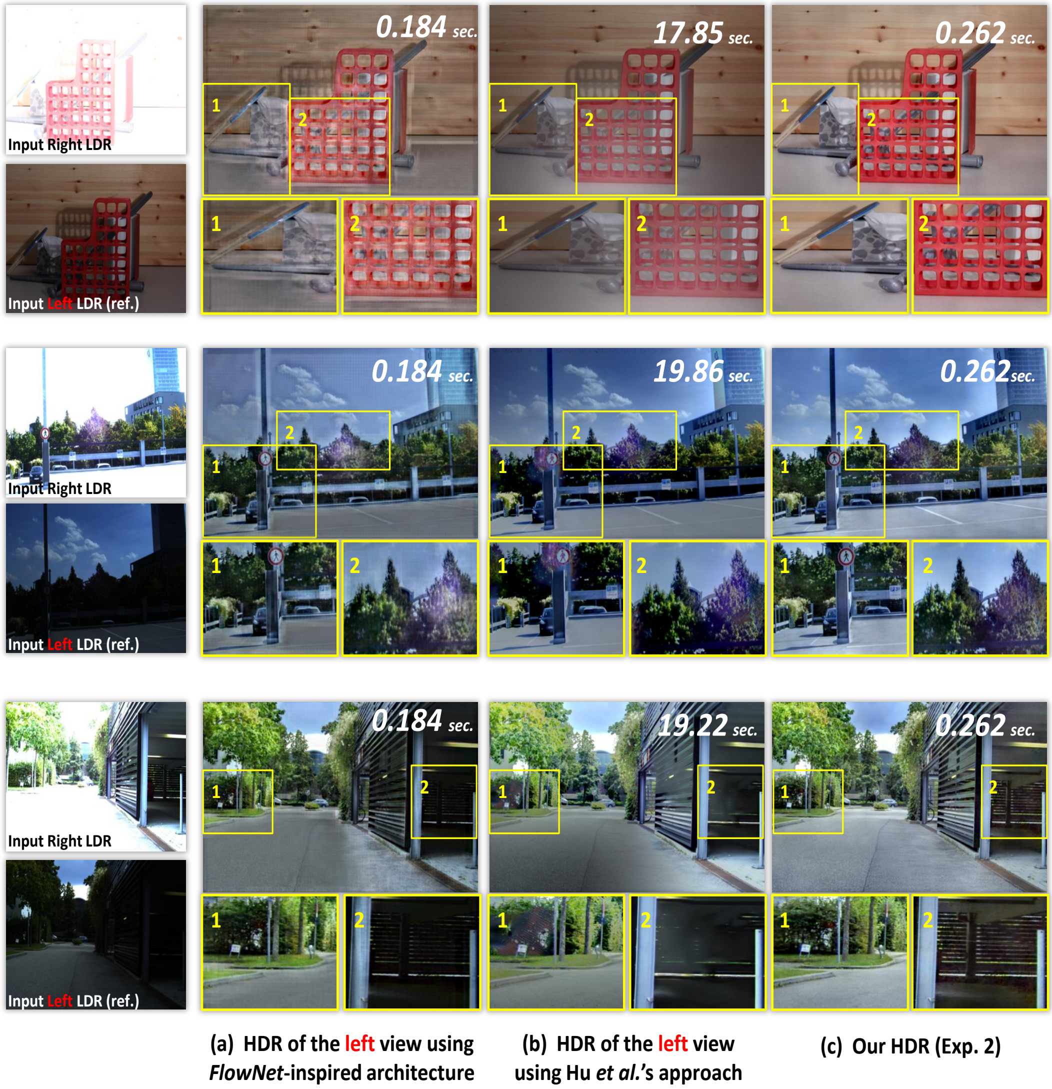


Figure 5.22: Visual Comparison between the rendered HDR images using the *FlowNet*-inspired approach (column **a**), the method of Hu *et al.* [15] (column **b**) and the results from the second HDR experiment (column **c**). In addition, we provide the corresponding execution times. Note that the computation time of Hu *et al.*'s approach apply only to the alignment part. Clearly, our HDR is artifact-free and yields the highest expansion of the dynamic range, despite the large color and scene differences between the input LDR images. This is more noticeable on the zoomed-in areas. Images of scene 1 are courtesy of [10].

HDR Experiment 3

The purpose of the third experiment is to examine the *relevance* of the previously outlined *Double-Loss FC-FlowNet* approach trained on the full of available LDR stack, to *free-motion* scenarios. This represents an important feature of the desired HDR rendering scheme. Evidently, the main goal is the extension of the dynamic range of the reference LDR while successfully removing any motion related artifacts.

One limitation concerning free-motion scenarios is related to the dataset. As explained earlier, the training of the aspired HDR model requires a specific cluster of the input LDR images. This includes the reference and the non-reference LDR images, together with the differently exposed instances of the reference image. These instances are necessary for the purpose of generating the corresponding "Ground-Truth" HDR image of the reference, which is deployed either as *label image* in the training set or for sake of performance evaluation in the validation set. In this context, the task of creating such a dataset with large number of scenes is challenging. The main limitation is related to capturing the differently exposed versions of the reference image, which is a task carried sequentially in time. Accordingly, capturing such a set of images in an uncontrolled environment (e.g. outdoor scenes) is arduous, as the captured images need to be "motion-free" which in turn implies that the capturing device as well as the depicted environment must be static .

To circumvent the lack of such datasets, we resolve to *fine tuning* in the third experiment. To this end, we start with extending the previously used stereo dataset (see Section 5.1.1) to remove the constraint on the *view* of the reference LDR. This means that in this experiment, we allow for the reference LDR to be taken from both **left and right** views, as opposed to the previous HDR experiments where we constantly set the reference LDR to be the **left dark** image. The underlying idea is to initially break the dependence of the trained HDR model on the *view* of the reference LDR, in order to facilitate the ensuing fine-tuning step. Accordingly, the only retained dependency with respect to the reference image is that it needs to be "darker" (under-exposed) in comparison to the non-reference LDR.

As a result, further data augmentation operations can be added in order to increase the size of the training set. These operations consist in a horizontal flip and a rotation along the diagonal axis. Consequently, the total number of training sequences amounts to 3080. Moreover, 352 sequences are available for validation purposes. Understandably, no data augmentation was performed on these sequences.

Using the trained HDR model on the extended dataset, we propose to fine-tune the learned model on a proper free-motion dataset. For this purpose, we rely on the dataset provided by Karaduzovic-Hadziabdic *et al.* and described in [1] (see Section 2.2). The dataset proposed by Karaduzovic-Hadziabdic *et al.* offers a total number of 36 indoor scenes with various types of motion such as complex motion, camera motion using a handheld capturing device and occlusions. Evidently, this dataset is too small to properly train an HDR model capable of handling various types of motion, exposure ratios and scene content. Finally, we test the performance of the fine-tuned HDR rendering on free-motion images with hidden ground-truth HDR images, since the aligned and differently exposed instances of the reference LDR are not available.

Likewise, the network configuration in terms of parameter settings is identical to the previous HDR experiments. Figures 5.23 and 5.24 contain a subjective comparison of the results of the HDR experiment 3 on *free-motion* samples. The input LDR images presented in Figures 5.23 and 5.24 depict various types of motion including complex and fast motion (scenes 1, 3 and 5), large simultaneous scene and object motion (scene 4) as well as non-rigid motion (scene 3) where the moving object changes its shape between the input LDR images making it difficult to track. Furthermore, most of the aforementioned motion types occur in the saturated areas. Accordingly, the manipulation of these objects is very challenging, as the detection and manipulation of motion-related objects in such area is a strenuous task. This is especially the case for the example LDR images of scene 2 (Fig. 5.23).

In addition, the presented examples exhibit extreme differences in terms of exposure ratio. Consequently, the input images contain large saturated areas where details are lost. Altogether, the described conditions concerning the motion types and exposure differences are suitable for the requirements we discussed previously in regard to the generalization performance of the HDR model that we aim to develop.

Taking into accounting all these conditions, the results of the third HDR experiment shown in Figures 5.23 and 5.24 prove that the learned **Double-Loss FC-FlowNet**-based model trained on the full stack of available LDR images and fine-tuned on a free-motion dataset is capable of handling all types of motion as well as exposure ratios. The final HDR images resulting from the third experiment depict consequently a wider range of illumination, which in turn implies that more details from the input images can be found in the final HDR image. Furthermore, no motion-related artifacts can be seen on the final results, as opposed to the HDR results gained using Hu *et al.*'s approach, where typical blurring artifacts can be noticed, mainly due to the faulty motion correction based on the *PatchMatch* algorithm. Similar to the previous generated results, the execution time of our HDR model is smaller than the time required by the approach of Hu *et al.*.

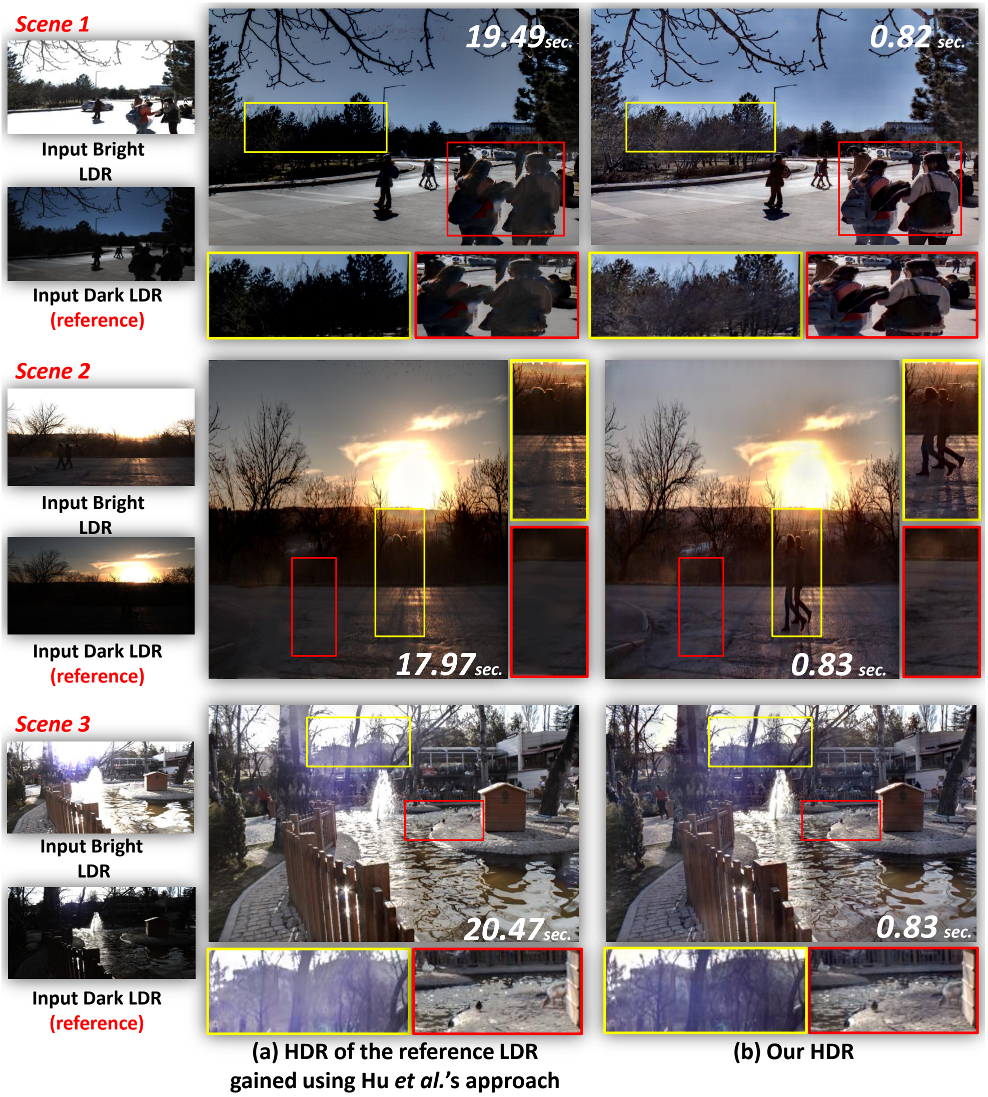


Figure 5.23: Visual Comparison between the rendered HDR images using the method of Hu *et al.* [15] (column **a**) and our HDR from the final third experiment (column **b**). In addition, we provide the corresponding execution times of both approaches. Note that the computation time of Hu *et al.*'s approach apply only to the alignment part. Despite the large difference in terms of scene content as well as exposure ration between the input LDR, our HDR model is able to effectively extend the dynamic range of the reference image while restraining all kinds of artifacts related to HDR rendering on dynamic scenes. This is more noticeable on the zoomed-in areas. Input images are courtesy of [3].

Figure 5.24: Visual Comparison between the rendered HDR images using the method of Hu *et al.* [15] (column **a**) and our HDR from the final third experiment (column **b**). In addition, we provide the corresponding execution times of both approaches. Note that the computation time of Hu *et al.*'s approach apply only to the alignment part. Once again, our HDR rendering model learned on the full stack of available HDR images is capable of successfully processing extreme cases where large motion occurs in the saturated areas. Input images of scene 4 are courtesy of [7] and images of scenes 5 and 6 are courtesy of [3].

5.4 Discussion

So far we created an end-to-end HDR rendering model which yields a HDR image from a set of 2 input LDR images, one of them being selected as the corresponding reference image. In this context, the conducted tests have confirmed the following features about our **Double-Loss FC-FlowNet**-based HDR model:

- The successful manipulation of extreme cases where the input stack of LDR images consists of only 2 images presenting large differences in regard to the depicted scene as well as the exposure ratio. Such cases are usually very challenging as the limited number of input images and the inherent large color difference restraints the effective reconstruction of all the details in the final HDR image.

- High quality of the final HDR image in terms of dynamic range. As shown in the previous experiments, learning the corresponding HDR model on an extended stack of LDR images (more than 2 images) enables the subsequent rendering of a final HDR image with a wider dynamic range, despite the limited number of input images in the application phase.

- Generalization performance of the HDR model. In addition to its flexibility concerning the exposure ratio (and the exposure times) between the input LDR images, the learned HDR model is not constrained to a specific type of motion between the input images. In fact, the results gained from the HDR experiment 3 have shown that fine-tuning the model on a free-motion dataset extends the applicability of the HDR model to various types of motion (camera and/or scene-related motion). The sole remaining constraint concerns therefore the reference image, which needs to be darker than the non-reference LDR image.

Considering these characteristics, there exist cases where the learned HDR rendering model does not effectively extend the dynamic range of the corresponding reference LDR. An example of such cases is presented in Fig. 5.25. The final HDR image corresponding to the input reference LDR (dark LDR) and generated using our **Double-Loss FC-FlowNet**-based HDR rendering model presents reconstruction artifacts in the region corresponding to the sky, as shown in the included zoom-in. In this example, the rendering model is unable to smoothly reconstruct the sky region. This observation can be explained by the fact that training dataset described earlier as well as the free-motion dataset used for the purpose of fine-tuning do not sufficiently cover such cases, which can be categorized as outdoor scenes. As a consequence, this type of scenes would benefit from extending the "*outdoor*-part of the initial training dataset and the fine-tuning dataset as well.

Figure 5.25: Example of the resulting HDR image using our **Double-Loss FC-FlowNet**-inspired HDR rendering model. In this case, the region depicting the sky details contains some artifacts. This might be explained by the fact that the initial training as well as the free-motion-set do not cover such cases sufficiently. However, the dynamic range of the remaining parts of the scene is successfully enhanced, which in turn improves the overall visual quality as shown in the output HDR image.

Furthermore, Fig. 5.26 illustrates an additional example where the expansion of the dynamic range using our HDR model is limited. As shown in the figure, the area corresponding to the face of the person in the scene does not contain all possible details available in the original non-reference image. In this context, the transfer of texture between the non-reference and the reference images is not very efficient. This might be explained by the extreme nature of the scene, in terms of exposure difference between the input LDR images as well as the fast and abrupt motion in the scene. Likewise, an expansion of the training and the free-motion datasets enables a better handling of such cases. In addition, a modified architecture might help to ensure an improved texture transfer between the input LDR images. Finally, increasing the number of input images to 3 enables an enhanced final HDR image, as the added LDR image is usually captured using balanced exposure setting and selected as the corresponding reference image. Accordingly, the architecture of the proposed CNN network needs to be adequately adapted to the case where the input stack consists of 3 images.

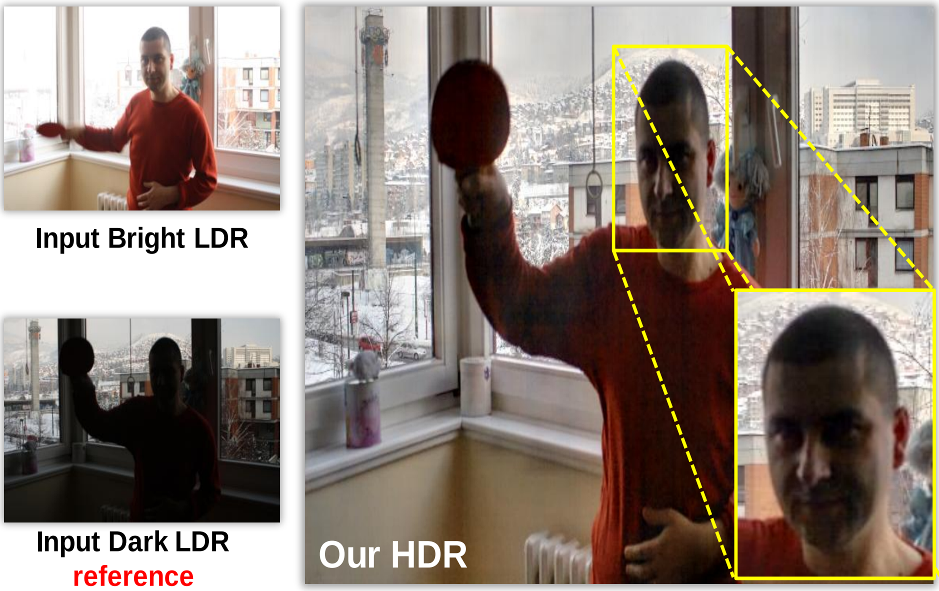


Figure 5.26: Example of the resulting HDR image using our **Double-Loss FC-FlowNet**-inspired HDR rendering model. As shown in the highlighted area, the details corresponding to the face region were not fully recovered from the non-reference image. This implies that the transfer of texture in this example was not efficient. However, the dynamic range of the remaining parts of the scene is successfully enhanced, despite the fast motion inherent to the scene. Input images courtesy of [7].

6 Conclusion

In this work, we focused on the topic of HDR rendering for dynamic scenes. As thoroughly explained in the Introduction section, our main goal is to render an HDR image based on a set of differently exposed input LDR images, where no artifacts related to the inconsistencies caused by motion can be seen. In this context, the most critical challenge is to find the perfect balance between the computational complexity and performance of the proposed approach(es). This emanates from the fact that we aim at developing a high performing HDR rendering framework on dynamic scenes for devices such as smartphone and tablet cameras. Accordingly, the inherent limitations in terms of computational resources to these devices puts a set of constraints which must be taken into consideration while developing the proposed solutions.

In this context, we start by proposing a *conventional* low-complexity de-ghosting approach based on the computation and processing of a binary motion map describing the locations of dynamic objects, in comparison to a selected reference LDR. The subsequent integration of the computed binary motion map(s) into the EF framework enables the exclusion of the dynamic objects and areas from the final HDR image. Consequently, no motion-induced artifacts or blurriness can be seen on the rendered HDR image. Aside from its ease of implementation and low-complexity, the proposed de-ghosting approach yields good quality even on extreme cases with only 2 or 3 input LDR images. Nonetheless, the analysis of the obtained results, and especially on challenging scenes, indicate that the HM algorithm is a possible weak link of the proposed processing pipeline.

Based on these observations, we propose a *Bayesian Framework* in order to improve the color mapping performance of the HM algorithm [4]. This is done by detecting possible outliers, which correspond to inaccurately estimated pixel intensities, based on the distribution of the target image. The subsequent evaluation shows that the proposed approach successfully enhances the quality of HM results. However, this improvement achieves a modest impact on the previously outlined HDR de-ghosting method. With this in mind, the examination of the task of color mapping suggests that the *one-to-one* mapping model inherent to the HM algorithm and to various other state-of-the-art color mapping approaches, results in a serious limitation of the mapping performance. Accordingly, we propose to advance upon the *one-to-one* model by considering each pixel as a inseparable component of its immediate environment during the color mapping task. For this purpose, we rely on a CNN-based framework, which through the adoption of the sliding kernel approach for performing convolution, indirectly incorporates additional information from the immediate vicinity of each pixel. The evaluation of the proposed novel technique for performing color mapping shows that we significantly improve upon HM and various other algorithms.

Furthermore, the enhancement of the color mapping model positively influences related applications such as *single-image HDRI* and *Stereo Matching*.

Considering these observations, we shift our focus back to HDRI for dynamic scenes, by adapting the previously explored CNN-based framework to the task of rendering an artifact-free HDR image from a set of LDR images with content and color differences. For this purpose, we initially review a more adequate CNN architecture to our application, namely the *FlowNet*-inspired [13] architecture. Next, we propose several modifications to the initial *FlowNet*-inspired architecture in order to enhance the quality of the HDR rendering. This results in the *Double-Loss FC-FlowNet* architecture. The ensuing assessment of the gained results implies that the proposed modifications successfully improves the HDR rendering performance. This improvement concerns in the first place the reduction of artifacts related to motion. Furthermore, the overall perceptual quality of the gained HDR is enhanced. Additional tests show as well that our CNN-based HDR rendering model can be easily extended to arbitrary motion, which meets the requirements discussed earlier concerning the desired features of the aspired solution. Finally, the low-complexity of the proposed framework enables the manipulation of target scenarios, namely extreme cases with low number of input LDR images, large exposure ratio and limited computational capabilities of the camera.

With this in mind, this work offers a novel approach to handle motion in the case of HDR rendering, namely by incorporating the CNN-based framework. In this context, several aspects of the proposed solutions can be further investigated and enhanced. These aspects are summarized in the following points:

- Explore various other CNN architectures, which build upon the proposed *Double-Loss FC-FlowNet* approach, so that further improvements to the quality of the rendered HDR images can be achieved. Accordingly, we are mostly interested in exploring alternative architectures which enable a more efficient transfer of texture and scene details between the input LDR images.

- Increase the *Diversity* aspect of the training datasets by increasing its corresponding size. A special attention should be given to scenes with challenging lighting conditions, such as outdoor or night scenes. In addition, the diversity of the training set can be enhanced by including a large number of *free-motion* scenes, which in turn positively impacts the HDR rendering generalization performance. However, alternative capturing strategies of such scenes must be developed.

- Develop novel and efficient CNN architectures and network designs for the cases where the input stack is composed of more than 2 input LDR images.

- Additional emphasis on the evaluation aspect of the rendered HDR images must be taken into consideration. The scope of novel HDR assessment approaches might include effective subjective testing, which can be in turn incorporated into the CNN framework.

Bibliography

[1] K. Karaduzovic-Hadziabdic, J. H. Telalovic, and R. Mantiuk, “Subjective and objective evaluation of multi-exposure high dynamic range image deghosting methods,” *Eurographics 2016*, vol. 35, no. 2, 2016.

[2] T. Mertens, J. Kautz, and F. Van Reeth, “Exposure fusion,” in *Pacific Graphics*, 2007, pp. 369–378.

[3] O. T. Tursun, A. O. Akyüz, A. Erdem, and E. Erdem, “An objective deghosting quality metric for HDR images,” in *Computer Graphics Forum*, vol. 35, no. 2, 2016, pp. 139–152.

[4] R. D. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. New Jersey: Prentice Hall, 2006.

[5] J. An, S. J. Ha, and N. I. Cho, “Probabilistic motion pixel detection for the reduction of ghost artifacts in high dynamic range images from multiple exposures,” in *EURASIP Journal on Image and Video Processing*, 2014, pp. 1–15.

[6] O. Gallo, N. Gelfand, W. Chen, M. Tico, and K. Pulli, “Artifact-free high dynamic range imaging,” in *IEEE International Conference on Computational Photography*, 2009, pp. 1–7.

[7] K. Karaduzovic-Hadziabdic, J. Hasic Telalovic, and R. Mantiuk, “Expert evaluation of deghosting algorithms for multi-exposure high dynamic range imaging,” in *Second International Conference and SME Workshop on HDR imaging*, 2014, pp. 1–4.

[8] P. Sen, N. Khademi, K. Maziar, Y. S. Darabi, D. B. Goldman, and E. Shechtman, “Robust patch-based HDR reconstruction of dynamic scenes,” in *SIGGRAPH Asia*, vol. 31, no. 6, 2012.

[9] J. Canny, “A computational approach to edge detection,” in *Transactions on Pattern Analysis and Machine Intelligence*, 1986, pp. 679–698.

[10] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, “High-resolution stereo datasets with subpixel-accurate ground truth,” in *German Conference on Pattern Recognition GCPR*, 2014, pp. 31–42.

[11] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001, pp. 131–140.

[12] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[13] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[14] I. Halfaoui, F. Bouzaraa, and O. Urfalioglu, "CNN-based initial background estimation," in *IEEE 23rd International Conference on Pattern Recognition*. In press, 2016.

[15] J. Hu, O. Gallo, K. Pulli, and X. Sun, "HDR deghosting: How to deal with saturation?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1163–1170.

[16] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 60–65.

[17] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *SPIE 6812, Image Processing: Algorithms and Sustem VI*, 2008.

[19] G. Johnson, "Cares and Concerns of CIE TC8-08: Spatial appearance modeling," in *Electronic Imaging Conference*, 2005, pp. 148–156.

[20] Cambridge in Colour. Dynamic range in digital photography. [Online]. Available: http://www.cambridgeincolour.com/tutorials/dynamic-range.htm

[21] ——. Dynamic range in digital photography. [Online]. Available: http://www.cambridgeincolour.com/tutorials/cameras-vs-human-eye.htm

[22] A. El Gamal and H. Eltoukhy, "Cmos image sensor," *IEEE Circuits and Devices Magazine*, vol. 21, no. 3, pp. 6–20, 2005.

[23] S. D. Freedman and F. Boussaid, "A high dynamic range CMOS image sensor with a novel pixel-level logarithmic counter memory," in *International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, 2015, pp. 14–19.

[24] S. K. Nayar and T. Mitsunaga, “High dynamic range imaging: Spatially varying pixel exposures,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 472–479.

[25] C. H. Cheng, O. C. Au, N. M. Cheung, C. H. Liu, and K. Y. Yip, “High dynamic range image capturing by spatial varying exposed color filter array with specific demosaicking algorithm,” in *2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2009, pp. 648–653.

[26] V. Brajovic and T. Kanade, “A sorting image sensor: An example of massively parallel intensity-to-time processing for low-latency computational sensors,” in *IEEE International Conference on Robotics and Automation*, vol. 2, 1996, pp. 1638–1643.

[27] V. M. Brajovic, R. I. Miyagawa, and T. Kanade, “Temporal photoreception for adaptive dynamic range image sensing and encoding,” *Neural Networks*, vol. 11, no. 7, pp. 1149–1158, 1998.

[28] A. El Gamal, “High dynamic range image sensors,” in *Tutorial at International Solid-State Circuits Conference*, vol. 290, 2002.

[29] Y. P. Orly and E. R. Fossum, “Wide intrascene dynamic range CMOS APS using dual sampling,” *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1721–1723, 1997.

[30] M. Sasaki, M. Mase, S. Kawahito, and Y. Tadokoro, “A wide-dynamic-range CMOS image sensor based on multiple short exposure-time readout with multiple-resolution column-parallel ADC,” *IEEE Sensors Journal*, vol. 7, no. 1, pp. 151–158, 2007.

[31] P. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” in *Proceedings SIGGRAPH*, 1997, pp. 369–378.

[32] T. Mitsunaga and S. K. Nayar, “Radiometric self calibration,” in *IEEE Computer Society Conference On Computer Vision and Pattern Recognition*, vol. 1, 1999, pp. 374–380.

[33] M. D. Grossberg and S. K. Nayar, “Modeling the space of camera response functions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1272–1282, 2004.

[34] M. Grossberg and S. K. Nayar, “Determining the camera response from images: What is knowable?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1455–1467, 2003.

[35] J. Takamatsu, Y. Matsushita, and K. Ikeuchi, “Estimating camera response functions using probabilistic intensity similarity,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[36] A. Yoshida, V. Blanz, K. Myszkowski, and H. P. Seidel, "Perceptual evaluation of tone mapping operators with real-world scenes," in *IS&T/SPIE Annual Symposium on Electronic Imaging*, 2005, pp. 192–203.

[37] E. A. Khan, A. O. Akyüz, and E. Reinhard, "Ghost removal in high dynamic range images," in *IEEE International Conference on Images Processing*, 2006, pp. 2005–2008.

[38] J. An, S. H. Lee, J. G. kuk, and N. L. Cho, "A multi-exposure image fusion algorithm without ghost effect," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 1565 – 1568.

[39] F. Pece and J. Kautz, "Bitmap movement detection: HDR for dynamic scenes," in *Conference on Visual Media Production*, 2010, pp. 1–8.

[40] G. Ward, "Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures," *Journal of Graphics Tools*, vol. 8, no. 2, pp. 17–30, 2003.

[41] S. Kang, M. Uyttendaeke, S. Winder, and R. Szelisk, "High dynamic range video," in *ACM SIGGRAPH*, 2003, pp. 319–325.

[42] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand HDR imaging of moving scenes with simultaneous resolution enhancement," in *Computer Graphics Forum*, vol. 30, 2011, pp. 405–414.

[43] P. Sen, N. Khademi Kalantari, M. Yaesoubi, S. Darabi, D. Goldman, and E. Shechtman, "Robust patch-based HDR reconstruction of dynamic scenes," *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012)*, vol. 31, no. 6, pp. 1–11, 2012.

[44] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *11th European Conference on Computer Vision ECCV*, 2010, pp. 29–43.

[45] O. Gallo, A. Troccoli, J. Hu, K. Pulli, and J. Kautz, "Locally non-rigid registration for mobile HDR photography," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 49–56.

[46] B. K. P. Horn and B. Schunk, "Determining optical flow," in *Artificial Intelligence*, vol. 17, 1981, pp. 185–203.

[47] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121–130.

[48] C. Liu, "Beyond pixels: Exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[49] A. Bruhn, J. Weickert, and Schnörr, "Lucas/kanade meets horn/schunk: combining local and global optical flow methods," *International Journal of Computer Vision*, vol. 67, no. 3, pp. 211–231, 2005.

[50] A. Barjatya, "Block matching algorithms for motion estimation," *IEEE Transactions Evolution Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[51] N. S. Love and C. Kamath, "An empirical study of block matching techniques for the detection of moving objects," Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, Tech. Rep. UCRL-TR-218038, 2006.

[52] F. Bouzaraa, I. Halfaoui, and O. Urfalioglu, "Ghost-free dual-exposure HDR for dynamic scenes," in *IEEE International Conference on Image Processing*, 2016, pp. 1739–1743.

[53] F. Bouzaraa, O. Urfalioglu, and G. Cordara, "Dual-exposure image registration for HDR processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 1553–1557.

[54] J. P. Rolland, V. Vo, B. Bloss, and C. K. Abbey, "Fast algorithms for histogram matching: Applications to texture synthesis," *Journal of Electronic Imaging*, vol. 9, no. 1, pp. 39–45, 2000.

[55] D. Shapira, S. Avidan, and Y. Hel-Or, "Multiple histogram matching," in *IEEE International Conference on Image Processing*, 2013, pp. 2269–2273.

[56] J. H. Justice, *Maximum Entropy and Bayesian Methods in Applied Statistics*. Cambridge: Cambridge University, 1986.

[57] F. Bouzaraa and O. Urfalioglu, "A naive bayes approach to improve histogram matching quality," in *IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

[58] ——, "CNN-based non local color mapping," in *IEEE International Symposium on Multimedia*, 2016, pp. 313–316.

[59] Y. S. Heo, K. M. Lee, and S. U. Lee, "Simultaneous color consistency and depth map estimation for radiometrically varying stereo images," *IEEE International Conference on Computer Vision*, pp. 1771–1778, 2009.

[60] H. S. Faridul, J. Stauder, J. Kervec, and A. Tremeau, "Approximate cross channel color mapping from sparse color correspondences," *IEEE International Conference on Computer Vision Workshops*, pp. 860–867, 2013.

[61] T. Osam, A. Hornung, R. Sumner, and M. Gross, "Fast and stable color balancing for images and augmented reality," *Second International IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pp. 49–56, 2012.

[62] K. Yamamoto, T. Yendo, T. Fuji, M. Tanimoto, and D. Suter, "Color correction for multiple camera system by using correspondances," *Journal of the Institute of Image Information and Television Engineers*, pp. 213–222, 2007.

[63] S. Schmidt and Z. N. Li, "High dynamic range stereo video using SIFT and simultaneous multi-exposure," in *IEEE International Conference on Signal and Image Processing Applications*, 2011, pp. 82–87.

[64] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[65] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features," *Computer Vision and Image Understanding*, pp. 346–359, 2008.

[66] D. Shapira, S. Avidan, and Y. Hel-Or, "Multiple histogram matching," *IEEE International Conference on Image Processing*, pp. 2269–2273, 2013.

[67] S. Kagarlitsky, Y. Moses, and Y. Hel-Or, "Piecewise-consistent color mappings of images acquired under various conditions," *IEEE International Conference on Computer Vision*, pp. 2311 – 2318, 2009.

[68] X. Dong, B. Bonec, Y. Zhu, and A. L. Yuille, "Region-based temporally consistent video post-processing," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 714 – 722, 2015.

[69] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," in *ACM Transactions on Graphics*, 2011, pp. 71–79.

[70] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Schechtman, "Robust patch-based HDR reconstruction of dynamic scenes," in *ACM Transactions on Graphics SIGGRAPH Asia*, 2012, pp. 203–214.

[71] C. Barnes, E. Schechtman, D. B. Goldman, and A. Finkelstein, "Patchmatch: A randomized correspondence algorithm for structural image editing," in *ACM Transactions on Graphics SIGGRAPH*, 2009.

[72] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–41, 2001.

[73] X. Xiao and L. Ma, "Color transfer in correlated color space," in *ACM International Conference on Virtual Reality Continuum and Its Applications*, 2006, pp. 305–309.

[74] ——, "Gradient-preserving color transfer," *Computer Graphics Forum*, vol. 28, pp. 1879–1886, 2009.

[75] F. Pitie, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, pp. 123–137, 2007.

[76] T. Pouli and E. Reinhard, "Progressive histogram reshaping for creative color transfer and tone reproduction," in *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering*, 2010, pp. 81–90.

[77] P. Tania and E. Reinhard, "Progressive color transfer for images of arbitrary dynamic range," *Computers and Graphics*, vol. 35, pp. 67 – 80, 2011.

[78] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015.

[79] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems 21*, 2009, pp. 769–776.

[80] A. Karbasi, A. H. Salavati, and A. Shokrollahi, "Iterative learning and denoising in convolutional neural associative memories," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 445–453.

[81] X. Junyuan, X. Linli, and C. Enhong, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 341–349.

[82] F. Guney and A. Geiger, "Displets: Resolving stereo ambiguities using object knowledge," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4165–4175.

[83] A. Dosovitskiy, J. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2015.

[84] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2015.

[85] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *IEEE International Conference on Computer Vision ICCV*, 2011.

[86] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[87] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.